Math 530 Spring 2023, Lecture diary

website: `https://www.cip.ifi.lmu.de/~grinberg/t/23s`

Note: This is a rough, unedited version of what I typed in class (but lacking the illustrations I drew on the blackboard)! See the 2022 notes for a more detailed and fleshed-out writeup of this material.

Lecture 1

0.1. Plan

This is a course on **graphs** – a rather elementary concept (actually a cluster of related concepts) that appears all over mathematics. We will discuss several kinds of graphs: simple graphs, multigraphs, simple digraphs, multidigraphs ("di" means "directed") and study their features and properties. In particular, we will see walks, cycles, paths, matchings, flows, ... on graphs.

The theory of graphs goes back to Euler in 1736. In the 19th century (?), Jacobi, Cayley, Borchardt and others picked up the subject in earnest. In the 20th century, it became mainstream. Many textbooks, lecture notes, journals on it now exist.

We will mostly follow my lecture notes from 2022 (`https://www.cip.ifi.lmu.de/~grinberg/t/22s/`). Feel free to interject with questions and ideas. If you are interested in research, this is a great place to start!

A few **administrativa**:

- The website (`https://www.cip.ifi.lmu.de/~grinberg/t/23s`) is the syllabus.
- HW1 is on the website. But I've changed problem 4.
- HWs will be due on Mondays at 23:59 (= 11:59 PM).
- We will use gradescope for HW. Please sign up there, using the code I sent out.
- **Homeworks should be typewritten, not handwritten.** You can use LaTeX or Office or Google Docs or even .txt. You can scan pictures (e.g., graphs).
- Office hours: Mon 1–3PM.

0.2. Notations

- We let $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$. In particular, $0 \in \mathbb{N}$.
- If S is a set, then the **powerset** of S means the set of all subsets of S. This powerset is denoted by $\mathcal{P}(S)$.
- Moreover, if S is a set and $k \in \mathbb{N}$, then $\mathcal{P}_k(S)$ means the set of all k-element subsets of S. For example,

$$\mathcal{P}_2(\{1,2,3\}) = \{\{1,2\}, \{1,3\}, \{2,3\}\}.$$

- For any number n and any $k \in \mathbb{N}$, we define the **binomial coefficient** $\binom{n}{k}$ to be the number
$$\frac{n(n-1)(n-2)\cdots(n-k+1)}{k!}.$$
This is the number of k-element subsets of any given n-element set. In other words, if S is an n-element set, then $|\mathcal{P}_k(S)| = \binom{n}{k}$.
If $n, k \in \mathbb{N}$ and $n \geq k$, then $\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$.
In particular, if S is an n-element set, then
$$|\mathcal{P}_2(S)| = \binom{n}{2} = \frac{n(n-1)}{2} = 1 + 2 + \cdots + (n-1).$$
Famously, the binomial coefficients satisfy **Pascal's recursion**
$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}.$$

1. Simple graphs

1.1. Definitions

The first type of graphs we will consider are the **simple graphs**, called so for the simplicity of their definition:

Definition 1.1.1. A **simple graph** is a pair (V, E), where V is a finite set, and E is a subset of $\mathcal{P}_2(V)$.

Thus, a simple graph is a pair (V, E), where V is a finite set, and E is a set of 2-element subsets of V. We will abbreviate "simple graph" as "graph" for a while, but later "graph" will have different meanings as well.

Example 1.1.2. Here is a simple graph:
$$(\{1,2,3,4\}, \{\{1,3\}, \{1,4\}, \{3,4\}\}).$$

Example 1.1.3. For any $n \in \mathbb{N}$, we can define a simple graph Cop_n to be the pair (V, E), where $V = \{1, 2, \ldots, n\}$ and
$$E = \{\{u, v\} \in \mathcal{P}_2(V) \mid \gcd(u, v) = 1\}.$$
We call this the **n-th coprimality graph**.

The purpose of simple graphs is to encode binary relations on a finite set – specifically the sort of relations that are symmetric (i.e., mutual) and irreflexive (i.e., no element relates to itself). For example, the graph Cop_n encodes the relation of being coprime on the set $\{1, 2, \ldots, n\}$, except that it "forgets" that 1 is coprime to itself.

Definition 1.1.4. Let $G = (V, E)$ be a simple graph.

(a) The set V is called the **vertex set** of G, and is denoted by $\operatorname{V}(G)$.

The elements of V are called the **vertices** (or the **nodes**) of G.

(b) The set E is called the **edge set** of G, and is denoted by $\operatorname{E}(G)$.

The elements of E are called the **edges** of G.

So every simple graph G satisfies $G = (\operatorname{V}(G), \operatorname{E}(G))$.

(c) We use the abbreviation uv for an edge $\{u, v\}$. Note that $uv = vu$.

(d) Two vertices u and v of G are said to be **adjacent** if $uv \in E$. In this case, the edge uv is said to **join** u with v (or **connect** u and v); the vertices u and v are called the **endpoints** of this edge.

(e) Let v be a vertex of G (that is, $v \in V$). Then, the **neighbors** of v (in G) are the vertices u of G that are adjacent to v (that is, satisfy $uv \in E$).

For example, the graph

$$G = (\{1, 2, 3, 4\}, \{\{1, 3\}, \{1, 4\}, \{3, 4\}\})$$

has vertex set $\operatorname{V}(G) = \{1, 2, 3, 4\}$ and edge set $\operatorname{E}(G) = \{\{1, 3\}, \{1, 4\}, \{3, 4\}\}$. Its vertices 1 and 3 are adjacent, but its vertices 1 and 2 are not. The neighbors of 1 are 3 and 4, whereas 2 has no neighbors. The endpoints of the edge 34 are 3 and 4.

Definition 1.1.5. A simple graph $G = (V, E)$ can be pictorially represented by

- drawing each vertex $v \in V$ as a point (at which we put the name of the vertex), and
- drawing each edge $uv \in E$ as a line (not necessarily straight) that connects the u-point to the v-point.

One place to draw graphs online is https://q.uiver.app/ (don't forget to remove the arrowheads).

Example 1.1.6. Consider the simple graph

$$(\{1,2,3,4,5\}, \mathcal{P}_2(\{1,2,3,4,5\})).$$

This is known as the "complete graph K_5". There are many ways to draw it. The most logical one (drawing it as a regular pentagon + regular pentagram) has a lot of crossing edges. If you are more strategic, you can reduce the crossings to just one crossing. But you cannot get rid of crossing edges completely. This is a nontrivial result in topology, one of the first results in the theory of **planar graphs**. (See [Fritsch/Fritsch].)

1.2. A first fact: The Ramsey number $R(3,3) = 6$

Proposition 1.2.1. Let G be a simple graph with $|\operatorname{V}(G)| \geq 6$. Then, at least one of the following two statements holds:

- *Statement 1:* There exist three distinct vertices a, b and c of G such that ab, bc and ca are edges of G.
- *Statement 2:* There exist three distinct vertices a, b and c of G such that none of ab, bc and ca is an edge of G.

In other words, we are claiming that if a graph G has at least 6 vertices, then it has either 3 mutually adjacent vertices, or 3 mutually non-adjacent (distinct) vertices, or both. Often, this is restated as follows: "In a group of at least six people, you can find either three mutual friends or three mutual non-friends".

Let us introduce some convenient terminology before proving this:

Definition 1.2.2. Let G be a simple graph.

(a) A set $\{a, b, c\}$ of three distinct vertices of G is said to be a **triangle** if ab, bc, ca are edges of G.

(b) A set $\{a, b, c\}$ of three distinct vertices of G is said to be an **anti-triangle** if none of ab, bc, ca is an edge of G.

So our above proposition claims that every simple graph with ≥ 6 vertices contains a triangle or an anti-triangle (or both).

Proof of the proposition. We need to show that G has a triangle or an anti-triangle.

Choose any vertex $u \in \operatorname{V}(G)$. Then, there are at least 5 vertices distinct from u (since G has at least 6 vertices). We are in one of the following two cases:

Case 1: The vertex u has at least 3 neighbors.

Case 2: The vertex u has at most 2 neighbors.

Consider Case 1. In this case, u has 3 distinct neighbors p, q, r (and possibly more). If $\{p, q, r\}$ is an anti-triangle, then we are done. If not, then at least one of pq, qr, rp is an edge, and thus forms a triangle with u. Either way, we are done in Case 1.

Now, consider Case 2. Here, u has at most 2 neighbors. Thus, u has at least 3 non-neighbors (distinct from u) (since there are at least 5 vertices distinct from u). In other words, u has 3 distinct non-neighbors p, q, r (and possibly more). If $\{p, q, r\}$ is a triangle, then we are done. If not, then at least one of pq, qr, rp is a non-edge, and thus forms an anti-triangle with u. Either way, we are done in Case 2.

So our proposition is proved. $\square$

The above proposition is the first result in a field of graph theory known as **Ramsey theory**. I shall not dwell on it, but let me state some main results. The first step beyound the above proposition is the following generalization:

Proposition 1.2.3. Let r and s be two positive integers. Let G be a simple graph with $|\operatorname{V}(G)| \geq \dbinom{r+s-2}{r-1}$. Then, at least one of the following two statements holds:

- *Statement 1:* There exist r distinct vertices of G that are mutually adjacent (i.e., any two distinct ones among them are adjacent).
- *Statement 2:* There exist s distinct vertices of G that are mutually non-adjacent (i.e., no two distinct ones among them are adjacent).

For $r = s = 3$, this becomes our previous proposition.

This generalization is actually not hard to prove by induction on $r + s$, using Pascal's recursion.

In general, the $\dbinom{r+s-2}{r-1}$ in the above proposition is not the smallest number that could stand in its place! The smallest number that could stand in its place is denoted by $R(r, s)$, and is called the (r, s)**-th Ramsey number**. We just showed that $R(3, 3) = 6$. In general, $R(r, s) \leq \dbinom{r+s-2}{r-1}$. Can we compute $R(r, s)$ for bigger r and s ?

There is no known general answer. There are some bounds that are slightly better than $\dbinom{r+s-2}{r-1}$, but all the values of $R(r, s)$ that have been really computed have been computed using a lot of case analysis and brute force. We now know that

$$R(3,4) = 9, \qquad R(3,5) = 14, \qquad R(3,6) = 18, \qquad R(3,7) = 23,$$
$$R(3,8) = 28, \qquad R(3,9) = 36, \qquad R(4,4) = 18, \qquad R(4,5) = 25,$$

and $R(1,s) = 1$ and $R(2,s) = s+1$ for all $s \geq 2$. These are all the Ramsey numbers known exactly. We also know, e.g., that $43 \leq R(5,5) \leq 48$.

Even this is far from the most general kind of Ramsey theory. See more in the 2022 notes.

1.3. Degrees

The **degree** of a vertex in a simple graph just counts how many edges contain this vertex:

Definition 1.3.1. Let $G = (V, E)$ be a simple graph. Let $v \in V$ be a vertex. Then, the **degree** of v (with respect to G) is defined to be

$$\begin{aligned} \deg v &:= (\text{the number of edges } e \in E \text{ that contain } v) \\ &= (\text{the number of neighbors of } v) \\ &= |\{u \in V \mid uv \in E\}| \\ &= |\{e \in E \mid v \in e\}| . \end{aligned}$$

(Note that these equality signs will no longer hold once we get to multigraphs.)

For example, in the graph shown on the board, we have

$$\deg 1 = 3, \qquad \deg 2 = 2, \qquad \deg 3 = 3, \qquad \deg 4 = 2, \qquad \deg 5 = 0.$$

Here are some basic properties of degrees in graphs:

Proposition 1.3.2. Let G be a simple graph with n vertices. Let v be any vertex of G. Then,

$$\deg v \in \{0, 1, \ldots, n-1\} .$$

Proposition 1.3.3 (Euler 1736)**.** Let G be a simple graph. Then, the sum of the degrees of all vertices of G equals twice the number of edges of G. In other words,

$$\sum_{v \in \mathrm{V}(G)} \deg v = 2 \cdot |\mathrm{E}(G)| .$$

Proof. Essentially, when you take the sum $\sum\limits_{v \in \mathrm{V}(G)} \deg v$, each edge is counted twice.

See the 2022 notes for a more rigorous way to say this, by "double-counting" the number of pairs (v, e) where $v \in V$ and $e \in E$ and $v \in e$. $\square$

Corollary 1.3.4 (handshake lemma)**.** Let G be a simple graph. Then, the number of vertices of G that have odd degree is even.

Proof. The preceding proposition yields that $\sum\limits_{v \in \operatorname{V}(G)} \deg v$ is even. Hence, it must have an even number of odd addends (because a sum with an odd number of odd addends would be odd). This proves the corollary. □

Here is another curious property of degrees in a simple graph:

Proposition 1.3.5. Let G be a simple graph with at least two vertices. Then, there exist two distinct vertices of G that have the same degree.

Proof. Assume the contrary. So the degrees of all n vertices are distinct, where $n = |\operatorname{V}(G)|$.

In other words, the map

$$\deg : \operatorname{V}(G) \to \{0, 1, \ldots, n-1\},$$
$$v \mapsto \deg v$$

is injective (i.e., is one-to-one). However, this map is a map between two finite sets of the same size (n). When such a map is injective, it is automatically bijective (this is one form of the pigeonhole principle). THus, our map deg is bijective. In particular, it is surjective, so that it takes both 0 and $n-1$ as values.

In other words, there are a vertex u with degree 0 and a vertex v with degree $n-1$. Are these two vertices adjacent or not? Both. Contradiction! Proof complete. □

Next time, a more interesting application.

Lecture 2

Here is an application of degrees to proving another fact about triangles:

Theorem 1.3.6 (Mantel's theorem)**.** Let G be a simple graph with n vertices and e edges. Assume that $e > n^2/4$. Then, G has a triangle (i.e., three distinct vertices that are mutually adjacent).

Proof. We will prove this by strong induction on n. Thus, we assume (as the induction hypothesis) that the theorem holds for all graphs with fewer than n vertices. We must now prove it for our graph G with its n vertices. Let $V = \operatorname{V}(G)$ and $E = \operatorname{E}(G)$, so that $G = (V, E)$.

We must prove that G has a triangle. Assume the contrary. Thus, G has no triangle.

From $e > n^2/4 \geq 0$, we see that G has at least one edge. Pick any such edge, and call it vw. Thus, $v \neq w$.

Let us now color each edge of G with one of three colors:

- The edge vw is colored black.
- Each edge that contains exactly one of v and w is colored red.
- All other edges are colored blue.

We now count the edges of each color:

- There is exactly 1 black edge, namely vw.
- There are at most $n - 2$ red edges. Indeed, any vertex other than v and w is joined to at most one of v and w by a red edge, since otherwise it would form a triangle with v and w.
- There are at most $(n-2)^2/4$ blue edges. Indeed, if there were more, then the induction hypothesis could be applied to the blue graph (i.e., the graph $(V \setminus \{v, w\}, \{\text{blue edges}\})$) would yield that the blue graph has a triangle, which would mean the same for the original graph.

In total, G has thus at most

$$1 + (n-2) + (n-2)^2/4$$

many edges. But $1 + (n-2) + (n-2)^2/4 = n^2/4$, so this is saying that $e \leq n^2/4$, which contradicts $e > n^2/4$ (our assumption). So we found the contradiction we wanted, and the induction step is complete. $\square$

Can we improve the $n^2/4$ bound in the theorem? No, since for each $n \in \mathbb{N}$, there is a simple graph with n vertices and $\lfloor n^2/4 \rfloor$ edges that has no triangle: namely,

$$(\{1, 2, \ldots, n\}, \{ij \mid i \not\equiv j \bmod 2\}).$$

So much for triangles. What about "higher" structures, e.g., several vertices mutually adjacent?

Theorem 1.3.7 (Turan's theorem)**.** Let r be a positive integer. Let G be a simple graph with n vertices and e edges. Assume that

$$e > \frac{r-1}{r} \cdot \frac{n^2}{2}.$$

Then, G has $r+1$ distinct vertices that are mutually adjacent.

This generalizes Mantel's theorem (which is obtained for $r = 2$). We will see a proof of this later.

1.4. Graph isomorphism

Two graphs can be distinct and yet "the same", meaning that they have the same vertices under different names. For instance,

$$1 - 2 - 3 \qquad \text{and} \qquad 1 - 3 - 2$$

are not the same graph, but they become the same if we relabel 2 and 3 as 3 and 2 in the former. Let us give this a name:

Definition 1.4.1. Let G and H be two simple graphs.

(a) A **graph isomorphism** (short: **isomorphism**) from G to H means a bijection $\phi : \operatorname{V}(G) \to \operatorname{V}(H)$ that "preserves edges": i.e., that has the property that for any two vertices v and w of G, we have

$$(vw \in \operatorname{E}(G)) \iff (\phi(v)\,\phi(w) \in \operatorname{E}(H)).$$

(b) We say that G and H are **isomorphic** (and write $G \cong H$) if there exists a graph isomorphism from G to H.

For example, the above two graphs

$$1 - 2 - 3 \qquad \text{and} \qquad 1 - 3 - 2$$

are isomorphic, since the map that sends $1, 2, 3$ to $1, 3, 2$ is an isomorphism. The map that sends $1, 2, 3$ to $2, 3, 1$ is also an isomorphism from the left graph to the right.

Here are some basic general properties of isomorphisms:

Proposition 1.4.2. Let G and H be two graphs. Then, the inverse of any graph isomorphism from G to H is a graph isomorphism from H to G.

Proposition 1.4.3. Let G, H and I be three graphs. If ϕ is an isomorphism from G to H, and ψ is an isomorphism from H to I, then $\psi \circ \phi$ is an isomorphism from G to I.

These propositions entail that $\cong$ is an equivalence relation.

Furthermore, graph isomorphisms preserve all "intrinsic" properties of a graph, e.g.:

Proposition 1.4.4. Let G and H be two simple graphs, and ϕ an isomorphism from G to H. Then:

(a) For every $v \in \operatorname{V}(G)$, we have $\deg_G v = \deg_H \phi(v)$. Here, $\deg_G v$ means the degree of v as a vertex of G.

(b) We have $|\operatorname{E}(H)| = |\operatorname{E}(G)|$.

Graph isomorphisms can be used to relabel the vertices of a graph. For example, if G is a graph with n vertices, then we can relabel its vertices as $1, 2, \ldots, n$ (i.e., we can find a graph isomorphic to G whose vertices are $1, 2, \ldots, n$). Let me state this precisely:

Proposition 1.4.5. Let G be a simple graph. Let S be a finite set such that $|S| = |\operatorname{V}(G)|$. Then, there exists a simple graph H that is isomorphic to G and has vertex set $\operatorname{V}(H) = S$.

1.5. Some families of graphs

We shall now discuss certain families of graphs.

1.5.1. Complete and empty graphs

Definition 1.5.1. Let V be a finite set.

(a) The **complete graph** on V means the simple graph $(V, \mathcal{P}_2(V))$. It is the simple graph with the vertex set V and with any two distinct vertices being adjacent.

If $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$, then this complete graph is called K_n.

(b) The **empty graph** on V means the simple graph $(V, \varnothing)$. It is the simple graph with the vertex set V and with no edges.

Note that a simple graph G is isomorphic to the complete graph K_n if and only if it has n vertices and every two of them are adjacent (except for a vertex and itself).

Question: Given two finite sets V and W, how many isomorphisms are there from the complete graph on V to the complete graph on W ?

Answer: If $|V| \neq |W|$, then there are none. Otherwise, $|V|!$, since they are just the bijections $V \to W$.

1.5.2. Path and cycle graphs

Definition 1.5.2. For each $n \in \mathbb{N}$, we define the n**-th path graph** P_n to be the simple graph

$$\begin{aligned} &(\{1,2,\ldots,n\}, \ \{\{i,i+1\} \ \mid \ 1 \leq i < n\}) \\ &= (\{1,2,\ldots,n\}, \ \{12, \ 23, \ 34, \ \ldots, \ (n-1)\,n\}). \end{aligned}$$

It looks as follows:

$$1 - 2 - 3 - \cdots - n.$$

It has n vertices and $n-1$ edges (unless $n = 0$, in which case it has 0 edges).

Definition 1.5.3. For each $n > 1$, we define the n**-th cycle graph** C_n to be the simple graph

$$\begin{aligned} &(\{1,2,\ldots,n\}, \ \{\{i,i+1\} \ \mid \ 1 \leq i < n\} \cup \{\{n,1\}\}) \\ &= (\{1,2,\ldots,n\}, \ \{12, \ 23, \ 34, \ \ldots, \ (n-1)\,n, \ n1\}). \end{aligned}$$

This graph has n vertices and n edges (unless $n = 2$, in which case it has 1 edge only). Actually, we will later modify the definition so that it does have 2 edges for $n = 2$.

Note that the cycle graph C_3 is the complete graph K_3. Also, $P_2 = K_2$.

Question: What are the isomorphisms from P_n to itself?

Answer: One is the identity map. The other is the reversal map

$$\begin{aligned} \{1,2,\ldots,n\} &\to \{1,2,\ldots,n\}, \\ i &\mapsto n+1-i. \end{aligned}$$

There are no others.

Question: What are the isomorphisms from C_n to itself?

Answer: For any $k \in \mathbb{Z}$, we can define a "rotation by k vertices", which is the map

$$\begin{aligned} \{1,2,\ldots,n\} &\to \{1,2,\ldots,n\}, \\ i &\mapsto (i+k \text{ reduced modulo } n). \end{aligned}$$

Thus, we get n rotations (one for each $k \in \{1, 2, \ldots, n\}$); all of them are graph isomorphisms. Furthermore, the reflections

$$\{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}, \\ i \mapsto (k - i \text{ reduced modulo } n)$$

for all $k \in \mathbb{Z}$ are isomorphisms as well, and again there are n of them. The group formed by these $2n$ isomorphisms in total (for $n > 2$) is called the n-th dihedral group.

1.5.3. Kneser graphs

Example 1.5.4. If S is a finite set, and if $k \in \mathbb{N}$, then we define the **k-th Kneser graph of** S to be the simple graph

$$K_{S,k} := (\mathcal{P}_k(S), \ \{IJ \ | \ I, J \in \mathcal{P}_k(S) \text{ with } I \cap J = \varnothing\}).$$

The vertices of $K_{S,k}$ are the k-element subsets of S, and two such subsets are adjacent if and only if they are disjoint.

The graph $K_{\{1,2,3,4,5\},\ 2}$ is called the **Petersen graph**.

1.6. Subgraphs

Definition 1.6.1. Let $G = (V, E)$ be a simple graph.

(a) A **subgraph** of G means a simple graph of the form $H = (W, F)$, where $W \subseteq V$ and $F \subseteq E$. In other words, a subgraph of G means a simple graph whose vertices are vertices of G and whose edges are edges of G.

(b) Let S be a subset of V. The **induced subgraph of** G **on the set** S denotes the subgraph

$$(S, \ E \cap \mathcal{P}_2(S))$$

of G. In other words, it denotes the subgraph of G whose vertices are the elements of S, and whose edges are those edges of G whose both endpoints lie in S.

(c) An **induced subgraph** of G means a subgraph of G that is the induced subgraph of G on S for some $S \subseteq V$.

Thus, a subgraph of a graph G is obtained by throwing away some vertices and some edges of G (in such a way that no edge is left "dangling"). Such a subgraph is an induced subgraph if and only if you don't throw away any edge without reason. Thus, induced subgraphs can be characterized as follows:

Proposition 1.6.2. Let H be a subgraph of a simple graph G. Then, H is an induced subgraph of G if and only if each edge $uv \in \operatorname{E}(G)$ whose both endpoints are vertices of H is also an edge of H.

Example 1.6.3. Let $n > 1$ be an integer.

(a) The path graph P_n is a subgraph of the cycle graph C_n. It is not an induced subgraph (for $n > 2$), since it is missing the edge $n1$ despite containing both of its endpoints.

(b) The path graph P_{n-1} is an induced subgraph of P_n. It is the induced subgraph of P_n on the set $\{1, 2, \ldots, n-1\}$. In other words, it is what remains of P_n if we remove the vertex n.

(c) Assume that $n > 3$. Is C_{n-1} a subgraph of C_n ? No, since the edge $(n-1)\,1$ exists in C_{n-1} but not in C_n.

Here is another easy fact:

Proposition 1.6.4. Let G be a simple graph, and let H be a subgraph of G. Assume that H is a complete graph. Then, H is automatically an induced subgraph of G.

We can now rewrite the notion of triangles in terms of induced subgraphs as follows:

Remark 1.6.5. Let G be a simple graph. Let u, v, w be three distinct vertices of G. Then, the following are equivalent:

1. The set $\{u, v, w\}$ is a triangle of G.
2. The induced subgraph of G on the set $\{u, v, w\}$ is isomorphic to K_3.
3. The induced subgraph of G on the set $\{u, v, w\}$ is isomorphic to C_3.

This lets you generalize questions about triangles to questions about subgraphs isomorphic to K_n or C_n or other fixed graphs, or induced subgraphs, ... – this is the beginning of a deep theory. Turan's theorem in particular is about subgraphs isomorphic to K_{r+1}.

1.7. Disjoint unions

Another way to construct new graphs from old is the disjoint union. Essentially, it takes two arbitrary graphs and puts them aside one another. You have to relabel the vertices of both graphs to ensure that the vertex sets are disjoint, but other than this it's exactly what you expect. Here is the formal definition:

Definition 1.7.1. Let $G_1, G_2, \ldots, G_k$ be simple graphs, where $G_i = (V_i, E_i)$ for each $i \in \{1, 2, \ldots, k\}$. The **disjoint union** of these k graphs $G_1, G_2, \ldots, G_k$ is the simple graph (V, E), where

$$V = \{(i, v) \mid i \in \{1, 2, \ldots, k\} \text{ and } v \in V_i\};$$
$$E = \{\{(i, v_1), (i, v_2)\} \mid i \in \{1, 2, \ldots, k\} \text{ and } \{v_1, v_2\} \in E_i\}.$$

This disjoint union is denoted by $G_1 \sqcup G_2 \sqcup \cdots \sqcup G_k$. (LaTeX for $\sqcup$ is \sqcup.)

If G and H are two graphs, then the graphs $G \sqcup H$ and $H \sqcup G$ are isomorphic, but usually not equal.

1.8. Walks and paths

Imagine a graph as a road network: Each vertex is a town, and each edge is a road that connects the respective towns. Then, you can often get from one town to another by a sequence of roads, even if they are not directly adjacent. This is made formal in the concept of a "walk":

Definition 1.8.1. Let G be a simple graph. Then:

(a) A **walk** (in G) means a finite sequence $(v_0, v_1, \ldots, v_k)$ of vertices of G (with $k \geq 0$) such that all of v_0v_1, v_1v_2, v_2v_3, $\ldots$, $v_{k-1}v_k$ are edges of G. (This latter condition is vacuously true for $k = 0$.)

(b) If $\mathbf{w} = (v_0, v_1, \ldots, v_k)$ is a walk, then:

- The **vertices** of $\mathbf{w}$ are $v_0, v_1, \ldots, v_k$.
- The **edges** of $\mathbf{w}$ are v_0v_1, v_1v_2, v_2v_3, $\ldots$, $v_{k-1}v_k$.
- The **length** of $\mathbf{w}$ is k. This is the number of edges of $\mathbf{w}$, and is 1 less than the number of vertices of $\mathbf{w}$.
- The vertex v_0 is called the **starting point** of $\mathbf{w}$, and the vertex v_k is called the **ending point** of $\mathbf{w}$.
- We say that the walk $\mathbf{w}$ **starts** (or **begins**) at v_0 and **ends** at v_k.

(c) A **path** (in G) means a walk whose vertices are distinct. In other words, a path means a walk $(v_0, v_1, \ldots, v_k)$ such that $v_0, v_1, \ldots, v_k$ are distinct.

(d) Let p and q be two vertices of G. Then, a **walk from p to q** means a walk that starts at p and ends at q. Likewise for "path from p to q".

(e) We often say "walk of G" instead of "walk in G". Likewise for paths.

Example 1.8.2. Let G be the graph on the blackboard. Then:

1. The sequence $(1,3,4,5,6,1,3,2)$ is a walk in G. Its length is 7. It is not a path. It is a walk from 1 to 2.
2. The sequence $(1,2,4,3)$ is not a walk in G, since 24 is not an edge.
3. The sequence $(1,3,2,1)$ is a walk from 1 to 1. It has length 3. It is not a path.
4. The sequence $(1,2,1)$ is a walk from 1 to 1. It has length 2. It is not a path.
5. The sequence (5) is a walk from 5 to 5. It has length 0. It is a path. More generally, every vertex v of G produces a length-0 path (v).
6. The sequence $(2,1,3,4,5,6)$ is a path from 2 to 6, and has length 5.
7. For any edge uv of G, the sequence (u,v) is a path from u to v of length 1.

Exercise 1. Prove that the edges of a path are always distinct. (Done in Spring 2017.)

1.8.1. Composing/concatenating and reversing walks

Proposition 1.8.3. Let G be a simple graph. Let u,v,w be three vertices of G. Let $\mathbf{a} = (a_0, a_1, \ldots, a_k)$ be a walk of G from u to v. Let $\mathbf{b} = (b_0, b_1, \ldots, b_\ell)$ be a walk from v to w. Then,

$$\begin{aligned} \mathbf{a} * \mathbf{b} &:= (a_0, a_1, \ldots, a_k, b_1, b_2, \ldots, b_\ell) \\ &= (a_0, a_1, \ldots, a_{k-1}, b_0, b_1, \ldots, b_\ell) \\ &= (a_0, a_1, \ldots, a_{k-1}, v, b_1, b_2, \ldots, b_\ell) \end{aligned}$$

is a walk from u to w.

Proposition 1.8.4. Let G be a simple graph. Let u and v be two vertices of G. Let $\mathbf{a} = (a_0, a_1, \ldots, a_k)$ be a walk of G from u to v. Then,

$$\operatorname{rev} \mathbf{a} := (a_k, a_{k-1}, \ldots, a_0)$$

is a walk from v to u. Moreover, if $\mathbf{a}$ is a path, then $\operatorname{rev} \mathbf{a}$ is a path.

1.8.2. Maximum lengths

Proposition 1.8.5. Let G be a simple graph with n vertices. Then, each path of G has length $\leq n-1$. In particular, G has only finitely many paths.

In contrast, G can often have infinitely many walks.

Lecture 3

2. Simple graphs (cont'd)

2.1. Walks and paths (cont'd)

To remind: A **walk** in a simple graph is a sequence of vertices where each is adjacent to the next. A **path** is a walk whose vertices are all distinct.

2.1.1. Reducing walks to paths

Proposition 2.1.1. Let G be a simple graph. Let u and v be two vertices of G. Let $\mathbf{a} = (a_0, a_1, \ldots, a_k)$ be a walk from u to v. Assume that $\mathbf{a}$ is not a path. Then, G has a walk from u to v that has length $< k$.

Proof. Since $\mathbf{a}$ is not a path, there exist $i < j$ such that $a_i = a_j$. Consider such $i < j$. Then,

$$\left(a_0, a_1, \ldots, a_i, a_{j+1}, a_{j+2}, \ldots, a_k\right)$$

is a walk from u to v that has length $k - j + i < k$.

(See 2022 notes for details.) □

Corollary 2.1.2 (When there is a walk, there is a path)**.** Let G be a simple graph. Let u and v be two vertices of G. Assume that there is a walk from u to v of length k for some $k \in \mathbb{N}$. Then, there is a path from u to v of length $\leq k$.

Proof. Apply the above proposition repeatedly. □

2.2. The equivalence relation "path-connected"

We can use the notions of walks and paths to define a certain equivalence relation on the vertex set $\operatorname{V}(G)$ of any simple graph G:

Definition 2.2.1. Let G be a simple graph. We define a binary relation $\simeq_G$ on the set $\operatorname{V}(G)$ as follows: Two vertices u and v of G satisfy $u \simeq_G v$ if and only if there exists a walk from u to v in G.

This binary relation $\simeq_G$ is called "**path-connectedness**". Two vertices u and v are said to be **path-connected** if they satisfy $u \simeq_G v$.

Proposition 2.2.2. This relation $\simeq_G$ is an equivalence relation.

Proof. We must prove that it is symmetric, reflexive and transitive:

- **Symmetry:** If $u \simeq_G v$, then $v \simeq_G u$, because we can take a walk from u to v and reverse it.
- **Reflexivity:** We always have $u \simeq_G u$, since the trivial walk (u) is a walk from u to u.
- **Transitivity:** If $u \simeq_G v$ and $v \simeq_G w$, then $u \simeq_G w$, because we can pick any walk $\mathbf{a}$ from u to v and any walk $\mathbf{b}$ from v to w and stick them together to a walk $\mathbf{a} * \mathbf{b}$ (as defined last time).

□

Proposition 2.2.3. Let G be a simple graph. Let u and v be two vertices of G. Then, $u \simeq_G v$ if and only if G has a path from u to v.

Proof. $\Longrightarrow$: A walk from u to v yields a path from u to v (according to the previous corollary).
$\Longleftarrow$: A path is a walk. □

Definition 2.2.4. Let G be a simple graph. The equivalence classes of the equivalence relation $\simeq_G$ are called the **connected components** (or, for short, **components**) of G.

Definition 2.2.5. Let G be a simple graph. We say that G is **connected** if G has exactly one component.

See the 2022 notes (Lecture 4) for examples.

Example 2.2.6. The complete graph on a nonempty set is connected.
The complete graph on an empty set is not connected: It has 0 components, not 1.

Example 2.2.7. The empty graph on a finite set V has $|V|$ many components: the singleton sets $\{v\}$ for $v \in V$. Thus, it is connected if and only if $|V| = 1$.

The following is easy to see:

Proposition 2.2.8. Let G be a simple graph. Let C be a component of G.
Then, the induced subgraph of G on the set C is connected.

Proof. Let $G[C]$ denote this induced subgraph.

We must prove that $G[C]$ is connected.

Let u and v be two vertices of $G[C]$, that is, two elements of C. Since C is a component (i.e., a $\simeq_G$-equivalence class), we then have $u \simeq_G v$. In other words, G has a walk from u to v. We claim that this walk is actually a walk of $G[C]$ (not only of G).

But this is easy: Any vertex of this walk is path-connected to u, and thus belongs to C. So this walk is really a walk of $G[C]$. Thus we obtain $u \simeq_{G[C]} v$.

So we have shown that any two vertices of $G[C]$ are path-connected in $G[C]$. Hence, the graph $G[C]$ has at most 1 component. But it also has at least 1 component (since C, being a component, is nonempty). So $G[C]$ has exactly 1 component, i.e., is connected. □

In the following proposition, we will be using the notation $G[C]$ for the induced subgraph of a graph G on a subset C of its vertex set.

Proposition 2.2.9. Let G be a simple graph. Let $C_1, C_2, \ldots, C_k$ be all components of C, listed without repetition. Then,

$$G \cong \underbrace{G[C_1] \sqcup G[C_2] \sqcup \cdots \sqcup G[C_k]}_{\text{disjoint union}}.$$

Proof. The bijection

$$G[C_1] \sqcup G[C_2] \sqcup \cdots \sqcup G[C_k] \to G,$$
$$(i, v) \mapsto v$$

is easily seen to be an isomorphism (since G has no edges that join vertices from different components). □

The above propositions show that any graph can be decomposed (up to isomorphism) as a disjoint union of connected graphs.

2.3. Closed walks and cycles

Definition 2.3.1. Let G be a simple graph.

(a) A **closed walk** of G means a walk whose starting point is its ending point. In other words, it means a walk $(w_0, w_1, \ldots, w_k)$ with $w_0 = w_k$. Sometimes, closed walks are called **circuits**, but this can also mean something else.

(b) A **cycle** of G means a closed walk $(w_0, w_1, \ldots, w_k)$ such that $k \geq 3$ and such that the vertices $w_0, w_1, \ldots, w_{k-1}$ are distinct.

Example 2.3.2. Let G be the simple graph

$$(\{1,2,3,4,5,6\}, \{12, 23, 34, 45, 56, 61, 13\}).$$

Then:

- The sequence $(1,2,3,1)$ is a closed walk of G, and actually a cycle of G. Other cycles are $(1,3,4,5,6,1)$ and $(1,6,5,4,3,2,1)$. Up to rotation and reversal, we have thus found all cycles of G.
- The sequence $(1,2,3,4,5,6,5,4,3,2,1)$ is a closed walk of G, but very much not a cycle.
- The sequence $(1,2,1)$ is a closed walk of G, but not a cycle. Same for (1).
- The walk $(1,2)$ is not a closed walk.

Authors have different opinions on whether $(1,2,3,1)$ and $(1,3,2,1)$ and $(2,3,1,2)$ count as the same cycle or not. However, this is not relevant to us, since we will not count those cycles.

We have now defined paths and cycles in an arbitrary simple graph. We have also defined path graphs P_n and cycle graphs C_n. Are these related? Yes:

Proposition 2.3.3. Let G be a simple graph.

(a) If $(p_0, p_1, \ldots, p_k)$ is a path of G, then there is a subgraph of G isomorphic to the path graph P_{k+1}, namely the subgraph

$$(\{p_0, p_1, \ldots, p_k\}, \{p_i p_{i+1} \mid 0 \leq i < k\}).$$

Conversely, any subgraph of G isomorphic to P_{k+1} gives a path of G.

(b) Now assume that $k \geq 3$. If $(c_0, c_1, \ldots, c_k)$ is a cycle of G, then there is a subgraph of G isomorphic to the cycle graph C_k, namely the subgraph

$$(\{c_0, c_1, \ldots, c_k\}, \{c_i c_{i+1} \mid 0 \leq i < k\}).$$

Conversely, any subgraph of G isomorphic to C_k gives a cycle of G.

Some graphs contain cycles; other graphs don't. For instance, the complete graph K_n contains lots of cycles (when $n \geq 3$), whereas the path graph P_n contains none. Let us try to find some criteria for when a graph can and when it cannot have cycles.

Definition 2.3.4. Let G be a simple graph. Let $\mathbf{w}$ be a walk of G. We say that $\mathbf{w}$ is **backtrack-free** if no two adjacent edges of $\mathbf{w}$ are identical.

Proposition 2.3.5. Let G be a simple graph. Let $\mathbf{w}$ be a backtrack-free walk of G. Then, $\mathbf{w}$ either is a path or contains a cycle (i.e., there exists a cycle of G whose edges are edges of $\mathbf{w}$).

Proof. Assume that $\mathbf{w}$ is not a path. We must show that $\mathbf{w}$ contains a cycle.

Write $\mathbf{w}$ as $\mathbf{w} = (w_0, w_1, \ldots, w_k)$. Since $\mathbf{w}$ is not a path, there exist $i < j$ such that $w_i = w_j$. Pick such i and j with smallest possible $j - i$.

Consider the subwalk $(w_i, w_{i+1}, \ldots, w_j)$ of $\mathbf{w}$. This is a closed walk. Moreover, the vertices $w_i, w_{i+1}, \ldots, w_{j-1}$ are distinct (since $j - i$ was minimal). To see that this $(w_i, w_{i+1}, \ldots, w_j)$ is a cycle, it thus remains to prove that $j - i \geq 3$. However, this is easy:

- We cannot have $j - i \leq 0$ (since $i < j$).
- We cannot have $j - i = 1$ (since there is no edge joining w_i to itself).
- We cannot have $j - i = 2$ (since this would mean $w_i w_{i+1} = w_{i+1} w_{i+2}$, but $\mathbf{w}$ is backtrack-free).

So we obtain a cycle. □

Corollary 2.3.6. Let G be a simple graph. Assume that G has a closed backtrack-free walk of length > 0. Then, G has a cycle.

Proposition 2.3.7. Let G be a simple graph. Let u and v be two vertices of G. Assume that there are two distinct backtrack-free walks from u to v. Then, G has a cycle.

Proof. We must prove the following:

If $\mathbf{p}$ and $\mathbf{q}$ are two distinct backtrack-free walks that start at the same vertex and end at the same vertex, then G has a cycle.

We will prove the proposition in this form.

We shall prove this by induction on the length of $\mathbf{p}$:

The *base case* (when $\mathbf{p}$ has length 0) is trivial.

In the *induction step*, we assume (as induction hypothesis) that the proposition is already proved for walks of length smaller than the length of $\mathbf{p}$. Now, consider two distinct backtrack-free walks $\mathbf{p}$ and $\mathbf{q}$ that start at the same vertex and end at the same vertex. We must show that G has a cycle.

Consider the closed walk $\mathbf{p} * \operatorname{rev} \mathbf{q}$ (where $\operatorname{rev} \mathbf{q}$ means the reversal of $\mathbf{q}$). If this walk $\mathbf{p} * \operatorname{rev} \mathbf{q}$ is backtrack-free, then we are done by the preceding corollary. If not, then the last edge of $\mathbf{p}$ is the last edge of $\mathbf{q}$ (since $\mathbf{p}$ and $\mathbf{q}$ are backtrack-free on their own), and then you can remove this last edge from both **p** and **q** and apply the induction hypothesis to the resulting shorter walks (which are still backtrack-free, still start at the same vertex, and still end at the same vertex). Qed.

(See the 2022 notes – Lecture 4, Theorem 1.2.6 – for more details.) □

2.4. The longest path trick

Proposition 2.4.1. Let G be a simple graph with at least one vertex. Let $d > 1$ be an integer. Assume that each vertex of G has degree $\geq d$. Then, G has a cycle of length $\geq d + 1$.

Proof. Let $\mathbf{p} = (v_0, v_1, \ldots, v_m)$ be a **longest** path of G. (Existence is easy.)

The vertex v_0 has degree $\geq d$, and thus has $\geq d$ neighbors.

If all its neighbors belonged to the set $\{v_1, v_2, \ldots, v_{d-1}\}$ (or $\{v_1, v_2, \ldots, v_m\}$ if $m < d - 1$), then v_0 would have $< d$ neighbors, which would contradict the previous sentence. Thus, not all neighbors of v_0 belong to the set $\{v_1, v_2, \ldots, v_{d-1}\}$. In other words, v_0 has a neighbor u that does not belong to $\{v_1, v_2, \ldots, v_{d-1}\}$. Consider this u. Note that $u \neq v_0$.

Attaching the vertex u to the front of the path $\mathbf{p}$, we obtain a walk

$$\mathbf{p}' := (u, v_0, v_1, \ldots, v_m).$$

If we had $u \notin \{v_0, v_1, \ldots, v_m\}$, then this walk $\mathbf{p}'$ would again be a path, which would contradict the fact that $\mathbf{p}$ is a **longest** path. So we must have $u \in \{v_0, v_1, \ldots, v_m\}$. In other words, $u = v_i$ for some $i \in \{0, 1, \ldots, m\}$.

Since $u \neq v_0$ and $u \notin \{v_1, v_2, \ldots, v_{d-1}\}$, this i must be $\geq d$. Therefore, the subwalk

$$(u, v_0, v_1, \ldots, v_i)$$

of $\mathbf{p}'$ has length $i + 1 \geq d + 1$. But this subwalk is a cycle. □

2.5. Bridges

One crucial question about graphs is: What happens to a graph if we remove a single edge from it? Let us introduce a notation for this:

Definition 2.5.1. Let $G = (V, E)$ be a simple graph. Let e be an edge of G. Then, $G \setminus e$ will mean the graph obtained from G by removing the edge e. In other words,

$$G \setminus e := (V,\ E \setminus \{e\}).$$

Some authors write $G - e$ for $G \setminus e$.

Theorem 2.5.2. Let G be a simple graph. Let e be an edge of G. Then:

(a) If e is an edge of some cycle of G, then the components of $G \setminus e$ are precisely the components of G.

(b) If e appears in no cycle of G (in other words, if there exists no cycle of G such that e is an edge of this cycle), then the graph $G \setminus e$ has one more component than G.

Proof. Here is a sketch (see 2022 for details).

(a) Assume that e is an edge of some cycle of G. We claim that the relation $\simeq_{G \setminus e}$ is precisely the relation $\simeq_G$. In other words, we claim that two vertices u and v of our graph G satisfy $u \simeq_{G \setminus e} v$ if and only if they satisfy $u \simeq_G v$.

The "only if" direction is obvious. For the "if" direction, you can always replace any use of e by a detour through the rest of the cycle. So the relations $\simeq_{G \setminus e}$ and $\simeq_G$ are identical. Thus, the components of $G \setminus e$ are the components of G.

(b) Assume that e appears in no cycle of G. Let u and v be the endpoints of e. We claim that:

1. The component of G that contains u and v breaks into two components of $G \setminus e$.

2. All other components of G remain components of $G \setminus e$.

Claim 2 is intuitively obvious, and easy to formalize.

For Claim 1, we first observe that u and v are not path-connected in $G \setminus e$. Indeed, if they were, then $G \setminus e$ would have a path from u to v, and we could then close this path to a cycle of G by inserting the edge e at its end. But this would contradict the assumption that e appears in no cycle of G. So we conclude that the component of G that contains u and v breaks into at least two components of $G \setminus e$. It remains to show that no more than two components are generated. In other words, we must show that every vertex w of this component is path-connected to either u or v in $G \setminus e$. To do so, we pick a vertex w of this component, and we fix a path $\mathbf{p}$ from w to u in G (such a path exists).

- If $\mathbf{p}$ does not use the edge e, then it remains a path in $G \setminus e$, so that we get $w \simeq_{G \setminus e} u$.

- If $\mathbf{p}$ does use the edge e, then e is the last edge of $\mathbf{p}$, and by removing this edge e from $\mathbf{p}$ we obtain a path from w to v in $G \setminus e$, so that $w \simeq_{G \setminus e} v$.

In either case, w is path-connected to either u or v in $G \setminus e$, thus belongs to either the component containing u or the component containing v. So the component of G that contains u and v breaks into at most two when we pass to $G \setminus e$. This completes our proof. $\square$

Here is a bit of terminology:

Definition 2.5.3. Let e be an edge of a simple graph G.

(a) We say that e is a **bridge** of G if e appears in no cycle of G.

(b) We say that e is a **cut-edge** of G if the graph $G \setminus e$ has more components than G.

Corollary 2.5.4. Let e be an edge of a simple graph G. Then, e is a bridge if and only if e is a cut-edge.

We can also define a "cut-vertex" of a simple graph G to be a vertex v such that if you remove v from G, then the resulting graph (called $G \setminus v$) has more components than G. Cut-vertices are subtler than cut-edges and also less important.

Lecture 4

2.6. Dominating sets

Now to something different:

Definition 2.6.1. Let $G = (V, E)$ be a simple graph.

A subset U of V is said to be **dominating** (for G) if every vertex $v \in V \setminus U$ has at least one neighbor in U.

Dominating subsets of V are called **dominating sets of** G.

Example 2.6.2. In the 5-cycle graph C_5, the set $\{1, 3\}$ is dominating, whereas $\{1, 2\}$ is not (since 4 has no neighbor in $\{1, 2\}$). Any set of size ≥ 3 is dominating for C_5, whereas any set of size ≤ 1 is not.

Some more examples:

- The whole vertex set V is always dominating. The empty set $\varnothing$ never is, unless $V = \varnothing$.
- In a complete graph K_n, any nonempty subset of $V = [n]$ is dominating.
- In an empty graph, only the whole vertex set V is dominating.

A useful problem is to find a dominating set of smallest possible size (for a given graph). There is no general answer, but there are some results. To state one, let me isolate a stupid case:

Definition 2.6.3. Let G be a simple graph. A vertex v of G is said to be **isolated** if it has no neighbors (i.e., if $\deg v = 0$).

An isolated vertex must belong to any dominating set. So you can ignore isolated vertices when you are looking for dominating sets (they don't help, and you just have to always keep them in). It remains to consider the case when a graph has no isolated vertices. In this case, you can show the following:

Proposition 2.6.4. Let $G = (V, E)$ be a simple graph that has no isolated vertices. Then:

(a) There exists a dominating subset of V that has size $\leq |V| / 2$.

(b) There exist two disjoint dominating subsets A and B of V such that $A \cup B = V$.

Proof. See hw#2 exercise 4. □

More surprisingly perhaps:

Theorem 2.6.5 (Brouwer's dominating set theorem)**.** Let G be a simple graph. Then, the number of dominating sets of G is odd.

Andries Brouwer gives three proofs in his 2009 note that I reference in the 2022 notes. Let me sketch a particularly neat one:

Definition 2.6.6. Let $G = (V, E)$ be a simple graph. A **detached pair** will mean a pair (A, B) of two disjoint subsets A and B of V such that there exists no edge $ab \in E$ with $a \in A$ and $b \in B$.

For instance, in the 6-cycle graph C_6, the pair $(\{1, 2\}, \{4, 5\})$ is detached.

Note that pairs are always ordered pairs in this course. Thus, if (A, B) is a detached pair, then so is (B, A), and these two pairs are distinct unless $A = B = \varnothing$. So the total number of detached pairs of a given graph G is odd.

Proof of Brouwer's dominating set theorem. Write G as $G = (V, E)$. Recall that $\mathcal{P}(V)$ means the set of all subsets of V.

Construct a new graph H with the vertex set $\mathcal{P}(V)$ as follows: Two subsets A and B of V will be adjacent vertices of H if and only if (A, B) is a detached pair.

I claim that the vertices of H having odd degree are precisely the subsets of V that are dominating:

> *Claim 1:* Let A be a subset of V. Then, the vertex A of H has odd degree if and only if A is a dominating set of G.

Proof of Claim 1. We let $N(A)$ denote the set of all vertices of G that have a neighbor in A.

The neighbors of A in H are the subsets B of V such that (A, B) is a detached pair. In other words, they are the subsets of $V \setminus (A \cup N(A))$. So there are $2^{|V \setminus (A \cup N(A))|}$ many of them. But of course, the number $2^{|V \setminus (A \cup N(A))|}$ is odd if and only if $|V \setminus (A \cup N(A))| = 0$, which means that $V \setminus (A \cup N(A)) = \varnothing$, which means precisely that A is dominating.

So we have shown that the number of neighbors of A in H is odd if and only if A is dominating (for G). This proves Claim 1. □

Claim 1 tells us that the odd-degree vertices of H are precisely the dominating sets of G. But we know from the handshaking lemma that H has an even number of odd-degree vertices. Thus, G has an even number of dominating sets.

Almost! Our definition of H has a flaw: It pretends to make $\varnothing$ adjacent to itself in H, but this falls afoul of the definition of a simple graph, which does not allow a vertex to be adjacent to itself. So we must tweak the definition of H to only allow detached pairs (A, B) with $A \neq B$. This results in the vertex $\varnothing$ changing its degree by 1, but nothing else changes (since the only detached pair (A, B) with $A = B$ is $(\varnothing, \varnothing)$). As a result, the number of odd-degree vertices changes by 1, so it becomes odd rather than even. And we're done. □

We can actually say more about the number of dominating sets. A very recent result by Heinrich and Tittmann (2017) shows the following:

Theorem 2.6.7. Let $G = (V, E)$ be a simple graph with n vertices, where $n > 0$.

Let α be the number of all detached pairs (A, B) such that both $|A|$ and $|B|$ are even and positive.

Let β be the number of all detached pairs (A, B) such that both $|A|$ and $|B|$ are odd.

Then:

(a) The numbers α and β are even.

(b) The number of dominating sets of G is $2^n - 1 + \alpha + \beta$.

This easily implies Brouwer's theorem. I reference a proof in the notes.

2.7. Hamiltonian paths and cycles

2.7.1. Basics

Now to something completely different.

We start with a simple question: Given a simple graph G, when is there a closed **walk** that contains each vertex of G ? The answer is simple: When G is connected.

The question becomes a lot more interesting if we replace "closed walk" by "path" or "cycle". The resulting objects have a name:

Definition 2.7.1. Let $G = (V, E)$ be a simple graph.

(a) A **Hamiltonian path** (short: **hamp**) in G means a walk of G that contains each vertex of G exactly once. Obviously, it is a path.

(b) A **Hamiltonian cycle** (short: **hamc**) in G means a cycle $(v_0, v_1, \ldots, v_k)$ of G such that each vertex of G appears exactly once among $v_0, v_1, \ldots, v_{k-1}$.

Some graphs have hamps; some don't. Having a hamc is even stronger than having a hamp, because if $(v_0, v_1, \ldots, v_k)$ is a hamc, then $(v_0, v_1, \ldots, v_{k-1})$ is a hamp.

The problem of finding hamps or hamcs, or even deciding their existence, is one of the famous NP-complete problems. An even harder problem is the **travelling salesman problem**, which asks for a Hamiltonian path of smallest weight. There is a lot of literature on the problem; it's an active area of research.

There are some nice sufficient criteria and necessary criteria for the existence of hamps and hamcs. Let me state a few:

Theorem 2.7.2 (Ore)**.** Let $G = (V, E)$ be a simple graph with n vertices, where $n \geq 3$.

Assume that $\deg x + \deg y \geq n$ for any two non-adjacent vertices x and y. Then, G has a hamc.

Proof sketch. Proof by gradual improvement (stepwise optimization, etc.; an idea that is used many times in graph theory and outside it):

A **listing** (of V) shall mean a list of elements of V that contains each element exactly once. It must always be an n-tuple.

The **hamness** of a listing $(v_1, v_2, \ldots, v_n)$ will mean the number of all $i \in \{1, 2, \ldots, n\}$ such that $v_i v_{i+1} \in E$. Here, we set $v_{n+1} = v_1$. In other words, the hamness of a listing tells you how often two consecutive entries of this listing are adjacent in G (where you count the last and the first entry as being adjacent too). Note that a hamc is the same as a listing of hamness n.

Now our plan is the following: Start with an arbitrary listing, and gradually improve it so that its hamness increases at every step. Eventually, its hamness will become n, at which point you will have a hamc.

For this to work, we need to show the following:

> *Claim 1:* Let $(v_1, v_2, \ldots, v_n)$ be a listing of hamness $k < n$. Then, there exists a listing of hamness $> k$.

Proof of Claim 1. Since the listing $(v_1, v_2, \ldots, v_n)$ has hamness $k < n$, there exists some $i \in \{1, 2, \ldots, n\}$ such that $v_i v_{i+1} \notin E$. Pick such an i, and observe that $\deg(v_i) + \deg(v_{i+1}) \geq n$ (by assumption). We can thus easily conclude that there exists a $j \in \{1, 2, \ldots, n\} \setminus \{i\}$ that satisfies both $v_i v_j \in E$ and $v_{i+1} v_{j+1} \in E$ (because there are $\deg(v_i)$ many j's satisfying $v_i v_j \in E$, and there are $\deg(v_{i+1})$ many j's satisfying $v_{i+1} v_{j+1} \in E$, so in total there are at least $\deg(v_i) + \deg(v_{i+1}) \geq n$ many j's that satisfy either $v_i v_j \in E$ or $v_{i+1} v_{j+1} \in E$, but $j = i$ satisfies neither condition, so that at least one j must satisfy both because $|A \cup B| + |A \cap B| = |A| + |B|$ for any finite sets A and B).

Now, replacing our listing $(v_1, v_2, \ldots, v_n)$ by

$$\left(v_j, v_{j-1}, \ldots, v_{i+1}, v_{j+1}, v_{j+2}, \ldots, v_i\right)$$

(that is, flipping the part between v_{i+1} and v_j) gives us a new listing of higher hamness than k. So we have found a listing of hamness $> k$. □

Thus, as we said above, we can gradually improve an arbitrary listing until its hamness reaches n. But at that point, it is a hamc. □

Corollary 2.7.3 (Dirac)**.** Let $G = (V, E)$ be a simple graph with n vertices, where $n \geq 3$.

Assume that $\deg x \geq \frac{n}{2}$ for each $x \in V$. Then, G has a hamc.

2.7.2. A necessary criterion

What about necessary criteria for hamcs and hamps? Here is one:

Proposition 2.7.4. Let $G = (V, E)$ be a simple graph.

For each subset S of V, we let $G \setminus S$ be the induced subgraph of G on the set $V \setminus S$. In other words, $G \setminus S$ is obtained from G when you remove the vertices in S and all edges that use these vertices.

Also, we let $\operatorname{conn} H$ denote the number of connected components of a simple graph H.

(a) If G has a hamc, then every nonempty $S \subseteq V$ satisfies $\operatorname{conn}(G \setminus S) \leq |S|$.

(b) If G has a hamp, then every $S \subseteq V$ satisfies $\operatorname{conn}(G \setminus S) \leq |S| + 1$.

Proof. **(a)** Let $S \subseteq V$ be a nonempty set. If we cut $|S|$ many vertices out of a cycle, then the cycle splits into at most $|S|$ many paths.

Therefore, if G has a hamc, then the removal of $|S|$ many vertices will break this hamc into $\leq |S|$ many paths. Thus, $G \setminus S$ has at most $|S|$ many components (since each of these $\leq |S|$ many paths remains connected). In other words, $\operatorname{conn}(G \setminus S) \leq |S|$.

(b) Analogous. □

2.7.3. Hypercubes

We move on to a concrete example of a graph that has a hamc.

Definition 2.7.5. Let $n \in \mathbb{N}$. The ***n*-hypercube** Q_n (more precisely, the ***n*-th hypercube graph**) is the simple graph with vertex set

$$\{0,1\}^n = \{(a_1, a_2, \ldots, a_n) \mid \text{each } a_i \text{ belongs to } \{0,1\}\}$$

and edge set defined as follows: Two vertices $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$ are adjacent if and only if there is **exactly** one $i \in \{1, 2, \ldots, n\}$ such that $a_i \neq b_i$. (For instance, in Q_4, the vertex $(1,0,1,1)$ is adjacent to $(1,0,0,1)$.)

The elements of $\{0,1\}^n$ are called **bitstrings** (or **binary words**), and their entries are called their **bits** (or **letters**). So two bitstrings are adjacent in Q_n if and only if they differ in exactly one bit.

We will often write a bitstring $(a_1, a_2, \ldots, a_n)$ as $a_1 a_2 \cdots a_n$. For example, $(0,1,1,0)$ is written as 0110.

Theorem 2.7.6 (Gray)**.** Let $n \geq 2$. Then, the graph Q_n has a hamc.

These hamcs are known as **Gray codes**. They are circular lists of bitstrings of length n such that two consecutive bitstrings in the list always differ in exactly one bit (and such that each bitstring appears exactly once in the list). See the WP article for applications.

Proof of the theorem. We will show something stronger:

> *Claim 1:* For each $n \geq 1$, the n-hypercube Q_n has a hamp from $00 \cdots 0$ to $100 \cdots 0$.

Once this claim is proved, the theorem will easily follow (since a hamp from $00 \cdots 0$ to $100 \cdots 0$ can be made into a hamc simply by repeating its starting point after its end).

So it suffices to prove Claim 1:

Proof of Claim 1. We induct on n:

Base case: For $n = 1$, the hamp from 0 to 1 in Q_1 is palpable.

Induction step: Fix $n \geq 2$. Assume (as the induction hypothesis) that Q_{n-1} has a hamp from $\underbrace{00 \cdots 0}_{n-1 \text{ zeroes}}$ to $1 \underbrace{00 \cdots 0}_{n-2 \text{ zeroes}}$. Let $\mathbf{p}$ be this hamp.

By attaching a 0 to the front of each bitstring in $\mathbf{p}$, we obtain a path

$$\mathbf{q} \text{ from } \underbrace{00 \cdots 0}_{n \text{ zeroes}} \text{ to } 01 \underbrace{00 \cdots 0}_{n-2 \text{ zeroes}} \text{ in } Q_n.$$

By attaching a 1 to the front of each bitstring in $\mathbf{p}$, we obtain a path

$$\mathbf{r} \text{ from } 1 \underbrace{00 \cdots 0}_{n-1 \text{ zeroes}} \text{ to } 11 \underbrace{00 \cdots 0}_{n-2 \text{ zeroes}} \text{ in } Q_n.$$

Now, I make a hamp of Q_n from $00 \cdots 0$ to $100 \cdots 0$ as follows:

- Start by walking along $\mathbf{q}$ from $\underbrace{00 \cdots 0}_{n \text{ zeroes}}$ to $01 \underbrace{00 \cdots 0}_{n-2 \text{ zeroes}} = 0$.
- Then move to the adjacent vertex $11 \underbrace{00 \cdots 0}_{n-2 \text{ zeroes}}$.
- Then walk $\mathbf{r}$ backwards from $11 \underbrace{00 \cdots 0}_{n-2 \text{ zeroes}}$ to $1 \underbrace{00 \cdots 0}_{n-1 \text{ zeroes}}$.

This is the hamp we need. So the induction step is done, and Claim 1 is proved. □

So the theorem is proved. □

See the 2022 notes (Section 1.2.5 in Lecture 6) for a generalization to Cartesian products.

2.7.4. Subset graphs

The n-hypercube Q_n can be reinterpreted in terms of subsets of $\{1, 2, \ldots, n\}$. Indeed, the bitstrings $a_1 a_2 \cdots a_n \in \{0, 1\}^n$ encode subsets of $\{1, 2, \ldots, n\}$ via the bijection

$$\{0,1\}^n \to \mathcal{P}(\{1, 2, \ldots, n\}),$$
$$a_1 a_2 \cdots a_n \mapsto \{i \in \{1, 2, \ldots, n\} \mid a_i = 1\}.$$

Under this bijection, two adjacent vertices of Q_n become two subsets of $\{1, 2, \ldots, n\}$ that differ in only one element (i.e., one of them is obtained from this other by inserting a single element). So a Gray code becomes a circular list of all subsets of $\{1, 2, \ldots, n\}$ such that two consecutive subsets in the list always differ by a single element. For instance, the Gray code

$$(000,\ 100,\ 110,\ 010,\ 011,\ 111,\ 101,\ 001,\ 000)$$

becomes the circular list

$$(\varnothing,\ \{1\},\ \{1,2\},\ \{2\},\ \{2,3\},\ \{1,2,3\},\ \{1,3\},\ \{3\},\ \varnothing).$$

So we conclude from the above theorem that we can list all the 2^n subsets of $\{1, 2, \ldots, n\}$ in a circular list such that any two consecutive subsets in the list differ by a single element.

See the 2022 notes for a somewhat different and much more difficult version of this.

Lecture 5

3. Multigraphs

3.1. Definitions

Simple graphs are just one version of graphs. Here is a more versatile but more complex one:

Definition 3.1.1. Let V be a set. Then, $\mathcal{P}_{1,2}(V)$ shall mean the set of all 1-element or 2-element subsets of V. In other words,

$$\begin{aligned}\mathcal{P}_{1,2}(V) &= \{S \subseteq V \mid |S| \in \{1,2\}\} \\ &= \{\{u,v\} \mid u,v \in V \text{ not necessarily distinct}\}.\end{aligned}$$

For instance,

$$\mathcal{P}_{1,2}(\{1,2,3\}) = \{\{1,2\}, \{1,3\}, \{2,3\}, \{1\}, \{2\}, \{3\}\}.$$

Definition 3.1.2. A **multigraph** is a triple (V,E,φ), where V and E are two finite sets, and $\varphi : E \to \mathcal{P}_{1,2}(V)$ is a map.

Example 3.1.3. The picture on the board (Example 1.1.3 in 2022 Lecture 7) is the multigraph (V,E,φ), where

$$\begin{aligned}V &= \{1,2,3,4,5\}, \\ E &= \{\alpha,\beta,\gamma,\delta,\varepsilon,\kappa,\lambda\}, \\ \varphi(\alpha) &= \{1,2\}, \\ \varphi(\beta) &= \{2,3\}, \\ \varphi(\gamma) &= \{2,3\}, \\ \varphi(\delta) &= \{4,5\}, \\ \varphi(\varepsilon) &= \{4,5\}, \\ \varphi(\kappa) &= \{4,5\}, \\ \varphi(\lambda) &= \{1\} = \{1,1\}.\end{aligned}$$

Multigraphs are a "more flexible version" of simple graphs, and many of the concepts we introduced for simple graphs have analogues for multigraphs:

Definition 3.1.4. Let $G = (V,E,\varphi)$ be a multigraph. Then:

(a) The elements of V are called the **vertices** of G.

The set V is called the **vertex set** of G, and is denoted $\operatorname{V}(G)$.

(a) The elements of E are called the **edges** of G.

The set E is called the **edge set** of G, and is denoted $\operatorname{E}(G)$.

(c) If e is an edge of G, then the elements of $\varphi(e)$ are called the **endpoints** of e.

(d) We say that an edge e contains a vertex v if $v \in \varphi(e)$ (in other words, if v is an endpoint of e).

(e) Two vertices u and v are said to be **adjacent** if G has an edge with endpoints u and v.

(f) Two edges e and f are said to be **parallel** if $\varphi(e) = \varphi(f)$. (In the example above, $\delta, \varepsilon, \kappa$ are mutually parallel.)

(g) We say that G has **no parallel edges** if no two distinct edges of G are parallel.

(h) An edge e is called a **loop** (or **self-loop**) if $\varphi(e)$ is a 1-element set. (In the above example, λ is a loop.)

(i) We say that G is **loopless** if G has no loops.

(j) The **degree** $\deg v$ (also $\deg_G v$) of a vertex v of G is defined to be the number of edges that contain v, where loops are counted twice. In other words,

$$\deg v = \deg_G v = |\{e \in E \mid v \in \varphi(e)\}| + |\{e \in E \mid \varphi(e) = \{v\}\}| .$$

(Note that, unlike the case of a simple graph, $\deg v$ is **not** the number of neighbors of v.)

(k) A **walk** in G means a list of the form

$$(v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k) \qquad (\text{where } k \geq 0),$$

where $v_0, v_1, \ldots, v_k$ are vertices of G, where $e_1, e_2, \ldots, e_k$ are edges of G, and where each $i \in \{1, 2, \ldots, k\}$ satisfies

$$\varphi(e_i) = \{v_{i-1}, v_i\}$$

(that is, the endpoints of each edge e_i are v_{i-1} and v_i).

A somewhat more intuitive notation for the above walk would be

$$\left(v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \cdots \xrightarrow{e_k} v_k\right).$$

The **vertices** of a walk $(v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k)$ are $v_0, v_1, \ldots, v_k$. The **edges** of this walk are $e_1, e_2, \ldots, e_k$. The walk is said to **start** at v_0 and **end** at v_k. Its **starting point** is v_0. Its **ending point** is v_k. Its **length** is k.

(l) A **path** means a walk whose vertices are distinct.

(m) The notions of "**path-connected**" and "**connected**" and "**component**" are defined exactly as for simple graphs. The symbol $\simeq_G$ still means "path-connected".

(n) A **closed walk** (or **circuit**) means a walk $(v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k)$ with $v_k = v_0$.

(o) A **cycle** means a closed walk $(v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k)$ such that

- the vertices $v_0, v_1, \ldots, v_{k-1}$ are distinct;
- the edges $e_1, e_2, \ldots, e_k$ are distinct;
- we have $k \geq 1$.

(Note that we are not requiring $k \geq 3$ any more, as we did for simple graphs. Thus, in the above example, $(2, \beta, 3, \gamma, 2)$ and $(1, \lambda, 1)$ are cycles, although $(2, \beta, 3, \beta, 2)$ is not. The purpose of the "$k \geq 3$" requirement for simple graphs was to disallow closed walks such as $(2, \beta, 3, \beta, 2)$ from being cycles; but now they are excluded by the "$e_1, e_2, \ldots, e_k$ are distinct" condition.)

(p) Hamiltonian paths and cycles are defined just as for simple graphs.

(q) We draw a multigraph by drawing each vertex as a point, each edge as a curve, and labelling both the vertices and the edges (just as in the above example).

So there are two main differences between simple graphs and multigraphs:

1. A multigraph can have loops, whereas a simple graph cannot.
2. In a simple graph, an edge e **is** a set of two vertices, whereas in a multigraph, an edge e **has** a set of two vertices (possibly equal ones, if e is a loop) assigned to it by the map φ. This not only allows for parallel edges, but also lets us store some information in the identities of the edges.

Nevertheless, the two notions have much in common, and so we will use the same word for them if ambiguity is not a problem:

Convention 3.1.5. The word "**graph**" means either a simple graph or a multigraph.

3.2. Conversions

There is a canonical way to turn a simple graph into a multigraph, and also a canonical way to go backwards (even though this involves losing information).

Here is the latter:

Definition 3.2.1. Let $G = (V, E, \varphi)$ be a multigraph. Then, the **underlying simple graph** G^{simp} of G means the simple graph

$$(V, \ \{\varphi(e) \ | \ e \in E \text{ is not a loop}\}).$$

In other words, it is the simple graph with vertex set V in which two vertices are adjacent if and only if they are adjacent in G. Visually speaking, it is obtained from G by removing loops and "collapsing" parallel edges into single edges.

Conversely:

Definition 3.2.2. Let $G = (V, E)$ be a simple graph. Then, the **corresponding multigraph** G^{mult} is defined to be the multigraph

$$(V, E, \iota),$$

where $\iota : E \to \mathcal{P}_{1,2}(V)$ is the map sending each $e \in E$ to e itself.

The "underlying simple graph" construction $G \mapsto G^{\text{simp}}$ loses some information (loops and parallel edges in particular), so it is irreversible. But the $G \mapsto G^{\text{mult}}$ comes as close as you can get to undoing it:

Proposition 3.2.3.

1. If G is a simple graph, then $\left(G^{\text{mult}}\right)^{\text{simp}} = G$.
2. If G is a loopless multigraph that has no parallel edges, then $\left(G^{\text{simp}}\right)^{\text{mult}} \cong G$. (This is an isomorphism of multigraphs; we will define this soon. Note that it is not an equality, because the "identities" of the edges of G are forgotten in G^{simp}.)
3. If G is a multigraph that has loops or (distinct) parallel edges, then $\left(G^{\text{simp}}\right)^{\text{mult}}$ has fewer edges than G and thus is not isomorphic to G.

We will often identify a simple graph G with the corresponding multigraph G^{mult}. In particular, when we define a notion for multigraphs, we automatically obtain the same notion for simple graphs. Sometimes this leads to a slight notational clash, when one and the same notion is defined for simple graphs and multigraphs in different ways. For instance, a cycle of a simple graph is just a list of vertices, whereas a cycle of a multigraph is a list of vertices and edges. However, the difference is not very substantial. For instance, the two meanings of "cycle" can be translated into each other:

Proposition 3.2.4. Let G be a simple graph. Then:

1. If $(v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k)$ is a cycle of the multigraph G^{mult}, then $(v_0, v_1, \ldots, v_k)$ is a cycle of the simple graph G.

2. If $(v_0, v_1, \ldots, v_k)$ is a cycle of the simple graph G, then

$$(v_0, \{v_0, v_1\}, v_1, \{v_1, v_2\}, v_2, \ldots, v_{k-1}, \{v_{k-1}, v_k\}, v_k)$$

is a cycle of the multigraph G^{mult}.

Proof. In part 1, you need to show that $k \geq 3$.

In part 2, you need to show that the edges are distinct.

Both are fairly easy, but not just a matter of definitions. Easy exercise! □

Similar facts about walks, paths, circuits are easily stated and proved.

3.3. Generalizing from simple graphs to multigraphs

Let us now look at some of the results we have seen for simple graphs and check which of them still hold (or can be generalized to hold) for multigraphs.

3.3.1. Ramsey

Recall Ramsey's $R(3,3) = 6$ theorem from Lecture 1:

Proposition 3.3.1. Let G be a simple graph with $|\mathrm{V}(G)| \geq 6$. Then, at least one of the following two statements holds:

- *Statement 1:* There exist three distinct vertices a, b and c of G such that ab, bc and ca are edges of G.

- *Statement 2:* There exist three distinct vertices a, b and c of G such that none of ab, bc and ca is an edge of G.

Is this true for multigraphs as well? Of course, we need to replace "ab is an edge of G" by "G has an edge with endpoints a and b", and likewise. If we do this, then yes, the proposition holds for multigraphs as well, since its claim does not change if we pass from the multigraph G to the simple graph G^{simp}.

3.3.2. Degrees

Back in Lecture 1, we defined degrees of vertices in a simple graph by

$$\begin{aligned}\deg v &:= (\text{the number of edges } e \in E \text{ that contain } v)\\ &= (\text{the number of neighbors of } v)\\ &= |\{u \in V \ \mid \ uv \in E\}|\\ &= |\{e \in E \ \mid \ v \in e\}|.\end{aligned}$$

These equalities **no longer hold** when G is a multigraph. Parallel edges correspond to the same neighbor, and loops are counted twice, so the number of neighbors of v is only a lower bound on $\deg v$.

Recall the following proposition:

Proposition 3.3.2. Let G be a simple graph with n vertices. Let v be any vertex of G. Then,

$$\deg v \in \{0, 1, \ldots, n-1\}.$$

This also **fails** for multigraphs, since $\deg v$ can be arbitrarily large due to loops or parallel edges.

Let us now recall Euler's old formula for the sum of the degrees:

Proposition 3.3.3 (Euler 1736)**.** Let G be a simple graph. Then, the sum of the degrees of all vertices of G equals twice the number of edges of G. In other words,

$$\sum_{v \in \operatorname{V}(G)} \deg v = 2 \cdot |\operatorname{E}(G)|.$$

This **does** generalize to multigraphs (and actually, this is one reason why we are counting loops twice when defining the degree of a vertex!). Let me state this precisely:

Proposition 3.3.4 (Euler 1736)**.** Let G be a multigraph. Then, the sum of the degrees of all vertices of G equals twice the number of edges of G. In other words,

$$\sum_{v \in \operatorname{V}(G)} \deg v = 2 \cdot |\operatorname{E}(G)|.$$

Proof. This is best explained as follows: Write G as $G = (V, E, \varphi)$. For each edge e, let us arbitrarily choose one of its endpoints and call it $\alpha(e)$. We then call the other endpoint $\beta(e)$ (of course, $\beta(e) = \alpha(e)$ if e is a loop). Then, for each vertex v, we have

$$\begin{aligned}\deg v = {} & (\text{the number of edges } e \in E \text{ with } \alpha(e) = v)\\ & + (\text{the number of edges } e \in E \text{ with } \beta(e) = v).\end{aligned}$$

Summing this equality over all $v \in V$, we obtain

$$\begin{aligned}
&\sum_{v \in V} \deg v \\
&= \underbrace{\sum_{v \in V} (\text{the number of edges } e \in E \text{ with } \alpha(e) = v)}_{=|E|} \\
&\qquad + \underbrace{\sum_{v \in V} (\text{the number of edges } e \in E \text{ with } \beta(e) = v)}_{=|E|} \\
&= |E| + |E| = 2 \cdot |E|,
\end{aligned}$$

which is precisely the claim of the proposition. □

The handshake lemma is still true for multigraphs:

Corollary 3.3.5 (handshake lemma)**.** Let G be a multigraph. Then, the number of vertices of G that have odd degree is even.

Here is another fact we saw back in Lecture 1:

Proposition 3.3.6. Let G be a simple graph with at least two vertices. Then, there exist two distinct vertices of G that have the same degree.

Is this true for multigraphs? **No**, since (e.g.) the multigraph $1 - 2 = 3$ fails it. What about Mantel's theorem?

Theorem 3.3.7 (Mantel's theorem)**.** Let G be a simple graph with n vertices and e edges. Assume that $e > n^2/4$. Then, G has a triangle (i.e., three distinct vertices that are mutually adjacent).

This again **fails for multigraphs**, since you can get the edge number arbitrarily large by spamming parallel edges or loops.

3.3.3. Graph isomorphisms

Graph isomorphisms (and isomorphy) can still be defined for multigraphs, but the definition is not the same. For simple graphs, an isomorphism is just a bijection between the vertex sets. For multigraphs, it needs to act both on vertices and on edges, so it really is a pair of two bijections:

Definition 3.3.8. Let $G = (V, E, \varphi)$ and $H = (W, F, \psi)$ be two multigraphs.

1. A **graph isomorphism** (or **isomorphism**) from G to H means a **pair** (α, β) of bijections

$$\alpha : V \to W \qquad \text{and} \qquad \beta : E \to F$$

such that if $\varphi(e) = \{v, w\}$, then $\psi(\beta(e)) = \{\alpha(v), \alpha(w)\}$.

2. We say that G and H are **isomorphic** (and we write $G \cong H$) if there exists a graph isomorphism from G to H.

Again, isomorphy of multigraphs is an equivalence relation.

3.3.4. Cycles

We have previously defined complete graphs K_n for each $n \geq 0$, as well as the path graphs P_n for $n \geq 0$, as well as the cycle graphs C_n for $n \geq 2$. We shall now define the 1-cycle graph C_1 and **redefine** the 2-cycle graph C_2 to look more like an actual cycle:

- The 2-cycle graph C_2 should now consist of two vertices 1 and 2 and two parallel edges between them.
- The 1-cycle graph C_1 shall consist of a single vertex 1 and a loop.

3.3.5. Submultigraphs

Definition 3.3.9. A **submultigraph** of a multigraph $G = (V, E, \varphi)$ means a multigraph of the form (W, F, ψ), where $W \subseteq V$ and $F \subseteq E$ and $\psi = \varphi \mid_F$.

Submultigraphs are called **subgraphs** if there is no confusion to fear.

We can also define induced submultigraphs:

Definition 3.3.10. Let $G = (V, E, \varphi)$ be a multigraph. Let S be a subset of V. The **induced submultigraph of** G **on the set** S denotes the submultigraph

$$\left(S,\ E',\ \varphi\mid_{E'}\right) \qquad \text{of } G,$$

where

$$E' := \{e \in E \mid \text{all endpoints of } e \text{ belong to } S\}.$$

It is denoted by $G[S]$.

3.3.6. Disjoint unions

Disjoint unions of multigraphs are defined similarly to the case of simple graphs.

3.3.7. Walk basics

Some basic properties of walks and paths still apply to multigraphs:

Proposition 3.3.11. Let G be a multigraph. Let u, v, w be three vertices of G. Let

$$\mathbf{a} = (a_0, e_1, a_1, e_2, a_2, \ldots, e_k, a_k) \text{ be a walk from } u \text{ to } v, \qquad \text{and}$$
$$\mathbf{b} = (b_0, f_1, b_1, f_2, b_2, \ldots, f_\ell, b_\ell) \text{ be a walk from } v \text{ to } w.$$

Then,

$$\begin{aligned}&(a_0, e_1, a_1, e_2, a_2, \ldots, e_k, a_k, f_1, b_1, f_2, b_2, \ldots, f_\ell, b_\ell)\\ &= (a_0, e_1, a_1, e_2, a_2, \ldots, e_k, b_0, f_1, b_1, f_2, b_2, \ldots, f_\ell, b_\ell)\\ &= (a_0, e_1, a_1, e_2, a_2, \ldots, e_k, v, f_1, b_1, f_2, b_2, \ldots, f_\ell, b_\ell)\end{aligned}$$

is a walk from u to w. We will denote it by $\mathbf{a} * \mathbf{b}$.

Walks can also be reversed (i.e., walked backwards). Details left to the reader. If a walk is not a path, it can be shortened:

Proposition 3.3.12. Let G be a multigraph. Let u and v be two vertices of G. Let $\mathbf{a} = (a_0, e_1, a_1, e_2, a_2, \ldots, e_k, a_k)$ be a walk from u to v. Assume that $\mathbf{a}$ is not a path. Then, G has a walk from u to v whose length is smaller than k.

Corollary 3.3.13. Let G be a multigraph. Let u and v be two vertices of G. Assume that there is a walk from u to v of length k for some $k \in \mathbb{N}$. Then, there is a path from u to v of length $\leq k$.

All the proofs are essentially the same as for simple graph.

3.3.8. Connectivity

The relation "path-connected" (denoted $\simeq_G$) is defined for multigraphs just as for simple graphs. Again, $u \simeq_G v$ if and only if there is a path from u to v.

The notions of "connected" and "component" are defined for multigraphs just as for simple graphs. Again:

- If C is a component of a multigraph G, then the induced submultigraph $G[C]$ is connected.
- If $C_1, C_2, \ldots, C_k$ are all the components of a multigraph G (listed with no repetition), then

$$G \cong G[C_1] \sqcup G[C_2] \sqcup \cdots \sqcup G[C_k].$$

3.3.9. Cycles

Furthermore, we have the following:

Proposition 3.3.14. Let G be a multigraph. Let $\mathbf{w}$ be a backtrack-free walk of G (that is, a walk such that no two adjacent edges of $\mathbf{w}$ are identical). Then, $\mathbf{w}$ either is a path or contains a cycle.

Proof. Same as for simple graphs, but easier. □

Just as for simple graphs, we get the following corollary:

Corollary 3.3.15. Let G be a multigraph. Assume that G has a closed backtrack-free walk $\mathbf{w}$ of length > 0. Then, G has a cycle.

Finally, we get:

Theorem 3.3.16. Let G be a multigraph. Let u and v be two vertices of G. Assume that there are two distinct backtrack-free walks from u to v. Then, G has a cycle.

Again, the proofs are the same as for simple graphs, but somewhat simpler. Next, recall the following:

Proposition 3.3.17. Let G be a simple graph with at least one vertex. Let $d > 1$ be an integer. Assume that each vertex of G has degree $\geq d$. Then, G has a cycle of length $\geq d + 1$.

This is **no longer** true for multigraphs, because loops and parallel edges can be used to pump up degrees without adding vertices.

3.3.10. Bridges

Definition 3.3.18. Let $G = (V, E, \varphi)$ be a multigraph. Let e be an edge of G. Then, $G \setminus e$ will mean the multigraph obtained from G by removing the edge e. In other words,

$$G \setminus e := \left(V,\ E \setminus \{e\},\ \varphi\mid_{E \setminus \{e\}}\right).$$

Some authors write $G - e$ for $G \setminus e$.

Theorem 3.3.19. Let G be a multigraph. Let e be an edge of G. Then:

(a) If e is an edge of some cycle of G, then the components of $G \setminus e$ are precisely the components of G.

(b) If e appears in no cycle of G (in other words, if there exists no cycle of G such that e is an edge of this cycle), then the graph $G \setminus e$ has one more component than G.

Proof. Same as for simple graphs. □

In particular, it follows that an edge of a multigraph is a bridge (= an edge that appears in no cycle) if and only if it is a cut-edge (= its removal increases the number of components).

3.3.11. Dominating sets

The theory of dominating sets for multigraphs is identical to the theory for simple graphs, since a dominating set of a multigraph G is the same as a dominating set of G^{simp}.

3.3.12. Hamiltonian paths and cycles

Neither Ore's nor Dirac's theorems hold for multigraphs. (Again, this is because degrees in a multigraph don't tell you much about getting around.)

The necessary criterion for hamcs and hamps actually does work for multigraphs. Here, again, the claim for multigraphs follows from the claim for simple graphs.

Lecture 6

3.4. Eulerian circuits and walks

Recall that a Hamiltonian path or cycle is a path or cycle that contains all vertices of the graph. Being a path or cycle, it has to contain each of them exactly once (except, in the case of a cycle, for its starting point).

What about a walk or closed walk that contains all **edges** exactly once? These are called "Eulerian" walks or circuits; here is the formal definition:

Definition 3.4.1. Let G be a multigraph.

1. A walk $\mathbf{w}$ of G is said to be **Eulerian** if each edge of G appears exactly once in this walk.

2. An **Eulerian circuit** of G means a circuit (i.e., a closed walk) of G that is Eulerian.

Note that Eulerian walks are usually not paths, and Eulerian circuits are usually not cycles.

I showed some examples in class (from Spring 2022 Lecture 8).

Unexpectedly, there is a general criterion for the existence of Eulerian circuits and walks, and there is a fairly efficient algorithm to find them. The criterion is called the **Euler–Hierholtzer theorem**:

Theorem 3.4.2 (Euler–Hierholzer)**.** Let G be a connected multigraph. Then:

(a) The multigraph G has an Eulerian circuit if and only if each vertex of G has even degree.

(b) The multigraph G has an Eulerian walk if and only if all but at most two vertices of G have even degree.

Why connected? If G has at least two components containing at least one edge each, then G certainly cannot have an Eulerian circuit or walk. If G has isolated vertices, then you can ignore those vertices. It thus suffices to consider connected graphs only.

It is easy to see why the "only if" parts of both parts of the Euler–Hierholzer theorem hold. Indeed, if $\mathbf{w}$ is an Eulerian walk, and v is a vertex that is neither the starting nor the ending point of $\mathbf{w}$, then $\mathbf{w}$ must enter and leave v the same number of times, so that the degree $\deg v$ must be even (since each edge appears exactly once on $\mathbf{w}$). If $\mathbf{w}$ is an Eulerian circuit, then this holds for the starting and ending point as well.

The hard parts are the "if" parts.

The proofs of the "if" parts will use some preparation.

Definition 3.4.3. Let G be a multigraph. A **trail** of G means a walk of G whose edges are distinct.

Notice: $\{\text{paths}\} \subseteq \{\text{trails}\} \subseteq \{\text{walks}\}$.
Clearly, any Eulerian walk is a trail.
So it seems like a reasonable strategy to find a Eulerian walk is to pick a longest possible trail and somehow argue that it must be Eulerian.
First of all, why is there a longest trail?

Lemma 3.4.4. Let G be a multigraph with at least one vertex. Then, G has a longest trail.

Proof. Each trail has length $\leq |\operatorname{E}(G)|$. Thus, there are only finitely many trails. But there is at least one trail (since a trivial path (v) is a trail). So there is a trail of largest length. □

Now, some notations.
We say that an edge e of a multigraph G **intersects** a walk $\mathbf{w}$ if at least one endpoint of e is a vertex of $\mathbf{w}$.

Lemma 3.4.5. Let G be a connected multigraph. Let $\mathbf{w}$ be a walk of G. Assume that $\mathbf{w}$ is not Eulerian.
Then, there exists an edge of G that is not an edge of $\mathbf{w}$ but intersects $\mathbf{w}$.

Proof. Since $\mathbf{w}$ is not Eulerian, there is an edge f that is not an edge of $\mathbf{w}$. Pick such an edge f. If f itself intersects $\mathbf{w}$, then we are done.
Otherwise, build a path from some endpoint of f to some vertex of $\mathbf{w}$ (such a path exists since G is connected). Choose the first edge of this path that is not an edge of $\mathbf{w}$. This edge must then intersect $\mathbf{w}$, so we are done. □

Lemma 3.4.6. Let G be a multigraph such that each vertex of G has even degree. Let $\mathbf{w}$ be a longest trail of G. Then, $\mathbf{w}$ is a closed walk.

Proof. Assume the contrary. Let s be the starting point of $\mathbf{w}$. Then, s is not the ending point of $\mathbf{w}$. Hence, $\mathbf{w}$ enters s one fewer time than it leaves s. As a consequence, $\mathbf{w}$ contains an odd number of edges that contain s. But in total, there is an even number of edges that contain s (since $\deg s$ is even by assumption). Hence, at least one edge that contains s is not in $\mathbf{w}$. Pick such an edge, and insert it into $\mathbf{w}$ at the beginning of $\mathbf{w}$. We get a longer trail than $\mathbf{w}$, but this contradicts the maximality of $\mathbf{w}$. □

Now we can finish the proof of the Euler–Hierholzer theorem:

Proof of Euler–Hierholzer. **(a)** $\Longrightarrow$: We have already explained this.

$\Longleftarrow$: Assume that each vertex of G has even degree.

By our first lemma, G has a longest trail. Let $\mathbf{w}$ be such a longest trail. By the last lemma, $\mathbf{w}$ is a closed walk. Assume (for contradiction) that $\mathbf{w}$ is not Eulerian. Then, by our second lemma, there exists an edge of G that is not an edge of $\mathbf{w}$ but intersects $\mathbf{w}$. Let f be this edge.

By cyclically rotating our closed walk $\mathbf{w}$, we ensure that $\mathbf{w}$ starts at an endpoint of f, and we then insert f at the beginning of $\mathbf{w}$. We thus obtain a longer trail than $\mathbf{w}$. But this contradicts the fact that $\mathbf{w}$ is a longest trail.

This contradiction shows that our assumption was false, so that $\mathbf{w}$ is Eulerian. Since $\mathbf{w}$ is closed, this yields that $\mathbf{w}$ is an Eulerian circuit. So we have proved the "$\Longleftarrow$" direction.

[Note that this proof, while being a proof by contradiction, actually contains a pretty good algorithm to find an Eulerian circuit. The way to do so is to read the proof "seriously but not literally".]

(b) $\Longrightarrow$: Already explained.

$\Longleftarrow$: Assume that all but at most two vertices of G have even degree. We must prove that G has an Eulerian walk.

If there are 0 vertices with odd degree, then we can use part **(a)** and we are done.

If there is exactly 1 vertex with odd degree, then we get a contradiction to the handshake lemma.

So it remains to consider the case when there are exactly 2 vertices with odd degree. Let u and v be these two vertices. Let G' be the multigraph obtained from G by adding an extra edge joining u with v. This graph G' has the property that all its vertices have even degree, and it is still connected (since G is connected). Thus, part **(a)** yields that G' has an Eulerian circuit. This circuit must use the edge uv exactly once. Rotate it so that it starts with this edge, and cut it at this edge. It thus becomes an Eulerian walk of G. And we are done. $\square$

4. Digraphs and multidigraphs

4.1. Definitions

We have so far seen two kinds of graphs: simple graphs and multigraphs.

For all their differences, they have a major commonality: Their edges are "two-way roads", i.e., an edge is (or has) an unordered pair of endpoints. So these kinds of graphs are good for modelling mutual (i.e., symmetric) relationships.

For non-mutual relationships, we need **directed graphs**, or, for short, **digraphs**. In such digraphs, each edge (now called an "arc") has a specified starting point (its "source") and a specified ending point (its "target"), and is drawn as an arrow (from its source to its target). Walks can only use arcs in the forward direction. Here are formal definitions:

Definition 4.1.1. A **simple digraph** is a pair (V, A), where V is a finite set, and where A is a subset of $V \times V$.

Definition 4.1.2. Let $D = (V, A)$ be a simple digraph.

1. The set V is called the **vertex set** of D, and is denoted $\operatorname{V}(D)$.

   Its elements are called the **vertices** (or **nodes**) of D.

2. The set A is called the **arc set** of D, and is denoted $\operatorname{A}(D)$.

   Its elements are called the **arcs** (or **directed edges**) of D.

   When u and v are two elements of V, we will occasionally abbreviate the pair (u, v) as uv.

3. If (u, v) is an arc of D (or, more generally, a pair in $V \times V$), then we call u the **source** of this arc, and v the **target** of this arc.

4. We draw D as follows: We represent each vertex each point, and each arc by an arrow pointing from its source to its target.

Example 4.1.3. For each $n \in \mathbb{N}$, we define the **divisibility digraph on** $\{1, 2, \ldots, n\}$ to be the simple digraph (V, A), where $V = \{1, 2, \ldots, n\}$ and

$$A = \{(i, j) \in V \times V \mid i \text{ divides } j\}.$$

Note that simple digraphs (unlike simple graphs) are allowed to have loops (i.e., arcs of the form (u, u)).

Definition 4.1.4. A **multidigraph** is a triple (V, A, ψ), where V and A are finite sets, and $\psi : A \to V \times V$ is a map.

Definition 4.1.5. Let $D = (V, A, \psi)$ be a multidigraph.

1. The set V is called the **vertex set** of D, and is denoted $\operatorname{V}(D)$.

   Its elements are called the **vertices** (or **nodes**) of D.

2. The set A is called the **arc set** of D, and is denoted $\operatorname{A}(D)$.

   Its elements are called the **arcs** (or **directed edges**) of D.

3. If a is an arc of D, and if $\psi(a) = (u, v)$, then the vertex u is called the **source** of a, and v is called the **target** of a.

4. We draw D as you would expect.

Convention 4.1.6. The word "**digraph**" means either "simple digraph" or "multidigraph", depending on the context.

4.2. Outdegrees and indegrees

Digraphs have a concept analogous to degrees for graphs. Actually they have two of these concepts:

Definition 4.2.1. Let D be a digraph with vertex set V and arc set A. (This can be a simple digraph or a multidigraph.) Let $v \in V$ be any vertex. Then:

1. The **outdegree** of v denotes the number of arcs of D whose source is v. It is denoted by $\deg^+ v$.
2. The **indegree** of v denotes the number of arcs of D whose target is v. It is denoted by $\deg^- v$.

Example 4.2.2. In the divisibility digraph on $\{1, 2, 3, 4, 5, 6\}$, we have

$$\begin{aligned} \deg^+ 1 &= 6, & \deg^- 1 &= 1, \\ \deg^+ 2 &= 3, & \deg^- 2 &= 2, \\ \deg^+ 3 &= 2, & \deg^- 3 &= 2, \\ \deg^+ 4 &= 1, & \deg^- 4 &= 3, \\ \deg^+ 5 &= 1, & \deg^- 5 &= 2, \\ \deg^+ 6 &= 1, & \deg^- 6 &= 4. \end{aligned}$$

Recall that in a graph, the sum of all degrees is twice the number of edges. For digraphs, an analogous fact holds:

Proposition 4.2.3 (diEuler)**.** Let D be a digraph with vertex set V and arc set A. Then,

$$\sum_{v \in V} \deg^+ v = \sum_{v \in V} \deg^- v = |A|.$$

Proof. Same idea as for graphs, but even easier: Each arc has exactly one source, so you can count the arcs source by source. Thus you get $\sum_{v \in V} \deg^+ v = |A|$. Similarly $\sum_{v \in V} \deg^- v = |A|$. Details in the 2022 notes (Lecture 9). □

4.3. Conversions

We now have four different types of "graph": simple graphs, multigraphs, simple digraphs, multidigraphs. There are some ways to convert between these types.

4.3.1. Multidigraphs to multigraphs

Any multidigraph D can be turned into an (undirected) graph G by "removing the arrowheads" (aka "forgetting the directions of the arcs"):

Definition 4.3.1. Let D be a multidigraph. Then, D^{und} will mean the multigraph obtained from D by replacing each arc with an edge whose endpoints are the source and the target of the arc. In other words, if $D = (V, A, \psi)$, then $D^{\text{und}} = (V, A, \varphi)$, where $\varphi : A \to \mathcal{P}_{1,2}(V)$ is defined as follows: If $\psi(a) = (u, v)$, then $\varphi(a) = \{u, v\}$.

4.3.2. Multigraphs to multidigraphs

Conversely, each multigraph G can be turned into a multidigraph G^{bidir} by "duplicating" each edge (turning it into two arcs going both ways):

Definition 4.3.2. Let $G = (V, E, \varphi)$ be a multigraph. For each edge $e \in E$, let us choose one of the endpoints of e and call it s_e; the other endpoint will then be called t_e. (If e is a loop, then $t_e = s_e$.)

We then define G^{bidir} to be the multidigraph $(V,\ E \times \{1, 2\},\ \psi)$, where $\psi : E \times \{1, 2\} \to V \times V$ is defined as follows:

$$\psi(e, 1) = (s_e, t_e) \qquad \text{and} \qquad \psi(e, 2) = (t_e, s_e)$$

for each edge $e \in E$.

We call G^{bidir} the **bidirectionalized multidigraph** of G.

Question: Is there a way to make this definition canonical?

Note that the map ψ depends on our choice of s_e's, but different choices lead to isomorphic multidigraphs G^{bidir}. (The notion of **isomorphism** for multidigraphs is defined in a pretty natural way.)

Note that the operation that sends G to G^{bidir} is injective – i.e., you can reconstruct G from G^{bidir}. In contrast, the operation that sends a multidigraph D to D^{und} is not injective, since it forgets the orientations of the arcs.

4.3.3. Simple digraphs to multidigraphs

We learned previously how to make a simple graph into a multigraph. A similar technique works for directed graphs:

Definition 4.3.3. Let $D = (V, A)$ be a simple digraph. Then, the **corresponding multidigraph** D^{mult} is defined to be the multidigraph

$$(V, A, \iota),$$

where $\iota : A \to V \times V$ is the map sending each arc $a \in A$ to a itself.

4.3.4. Multidigraphs to simple digraphs

In analogy to the $G \mapsto G^{\text{simp}}$ operation for graphs (which drops loops and collapses parallel edges), we can define a $D \mapsto D^{\text{simp}}$ operation for digraphs:

Definition 4.3.4. Let $D = (V, A, \psi)$ be a multidigraph. Then, the **underlying simple digraph** D^{simp} of D means the simple digraph

$$(V, \ \{\psi(a) \ | \ a \in A\}).$$

In other words, D^{simp} is obtained by "collapsing" parallel arcs (note that loops are not collapsed).

4.3.5. Multidigraphs as a big tent

As a consequence, every notion of "graph" we have seen so far can be transformed losslessly into a multidigraph:

- Each simple graph becomes a multigraph via $G \mapsto G^{\text{mult}}$.
- Each multigraph, in turn, becomes a multidigraph via $D \mapsto D^{\text{bidir}}$.
- Each simple digraph becomes a multidigraph via $D \mapsto D^{\text{mult}}$.

Thus, multidigraphs are the most general notion of "graphs". If a theorem holds for multidigraphs, you can then automatically conclude properties of the other types of "graphs" from it.

4.4. Walks, paths, closed walks, cycles

Let us define walks (and their various types) for digraphs. We begin with the case of simple digraphs:

Definition 4.4.1. Let D be a simple digraph. Then:

1. A **walk** (in D) means a finite sequence $(v_0, v_1, \ldots, v_k)$ of vertices of D (with $k \geq 0$) such that $v_0v_1, \ v_1v_2, \ v_2v_3, \ \ldots, \ v_{k-1}v_k$ are arcs of D.
2. If $\mathbf{w} = (v_0, v_1, \ldots, v_k)$ is a walk of D, then:

a) The **vertices** of **w** are defined to be $v_0, v_1, \ldots, v_k$.

b) The **arcs** of **w** are defined to be the pairs $v_0v_1,\ v_1v_2,\ v_2v_3,\ \ldots,\ v_{k-1}v_k$.

c) The nonnegative integer k is called the **length** of **w**.

d) The vertex v_0 is called the **starting point** of **w**, and the vertex v_k is called the **ending point** of **w**.

e) The walk **w** is said to **start** at v_0 and to **end** at v_k.

3. A **path** (in D) means a walk whose vertices are distinct.

4. A **walk from** p **to** q means a walk that starts at p and ends at q. A **path from** p **to** q means a path that starts at p and ends at q.

5. A **closed walk** means a walk whose starting point is its ending point. It is also called a **circuit**.

6. A **cycle** of D means a closed walk $(w_0, w_1, \ldots, w_k)$ such that $k \geq 1$ and such that the vertices $w_0, w_1, \ldots, w_{k-1}$ are distinct.

See the 2022 notes for examples.
We can also define all of these things for multidigraphs:

Definition 4.4.2. Let $D = (V, A,)$ be a multidigraph. Then:

1. A **walk** (in D) means a finite sequence $(v_0, a_1, v_1, a_2, v_2, \ldots, a_k, v_k)$ of vertices and arcs of D (with $k \geq 0$) such that $v_0, v_1, \ldots, v_k$ are vertices and $a_1, a_2, \ldots, a_k$ are arcs and $\psi(a_i) = (v_{i-1}, v_i)$ for each $i \in \{1, 2, \ldots, k\}$.

2. If $\mathbf{w} = (v_0, a_1, v_1, a_2, v_2, \ldots, a_k, v_k)$ is a walk of D, then:

   a) The **vertices** of **w** are defined to be $v_0, v_1, \ldots, v_k$.

   b) The **arcs** of **w** are defined to be the arcs $a_1, a_2, \ldots, a_k$.

   c) The nonnegative integer k is called the **length** of **w**.

   d) The vertex v_0 is called the **starting point** of **w**, and the vertex v_k is called the **ending point** of **w**.

   e) The walk **w** is said to **start** at v_0 and to **end** at v_k.

3. A **path** (in D) means a walk whose vertices are distinct.

4. A **walk from** p **to** q means a walk that starts at p and ends at q. A **path from** p **to** q means a path that starts at p and ends at q.

5. A **closed walk** means a walk whose starting point is its ending point. It is also called a **circuit**.

6. A **cycle** of D means a closed walk $(w_0, a_1, w_1, \ldots, a_k, w_k)$ such that $k \geq 1$ and such that the vertices $w_0, w_1, \ldots, w_{k-1}$ are distinct. (This yields that the arcs are distinct – check it!)

Lecture 7

Last time, we defined digraphs of two kinds (simple digraphs and multidigraphs), and we defined walks, paths, circuits and cycles in them.

Let's study a few of their properties:

Proposition 4.4.3. Let D be a multidigraph. Let u, v and w be three vertices of D. Let

$$\begin{aligned} \mathbf{a} &= (a_0, e_1, a_1, e_2, a_2, \ldots, e_k, a_k) && \text{be a walk from } u \text{ to } v, \quad \text{and} \\ \mathbf{b} &= (b_0, f_1, b_1, f_2, b_2, \ldots, f_\ell, b_\ell) && \text{be a walk from } v \text{ to } w. \end{aligned}$$

Then,

$$\begin{aligned} \mathbf{a} * \mathbf{b} &:= (a_0, e_1, a_1, e_2, a_2, \ldots, e_k, a_k, f_1, b_1, f_2, b_2, \ldots, f_\ell, b_\ell) \\ &= (a_0, e_1, a_1, e_2, a_2, \ldots, e_k, b_0, f_1, b_1, f_2, b_2, \ldots, f_\ell, b_\ell) \\ &= (a_0, e_1, a_1, e_2, a_2, \ldots, e_k, v, f_1, b_1, f_2, b_2, \ldots, f_\ell, b_\ell) \end{aligned}$$

is a walk from u to w.

However, unlike for undirected graphs, we can no longer reverse a walk. So the existence of a walk from u to v does not ensure the existence of a walk from v to u.

We can again reduce walks to paths:

Proposition 4.4.4. Let D be a multidigraph. Let u and v be two vertices of D. Let $\mathbf{a}$ be a walk from u to v. Let k be the length of $\mathbf{a}$. Assume that $\mathbf{a}$ is not a path. Then, there exists a walk from u to v whose length is smaller than k.

Corollary 4.4.5 (When there is a walk, there is a path)**.** Let D be a multidigraph. Let u and v be two vertices of D. Assume that there is a walk from u to v of length k for some $k \in \mathbb{N}$. Then, there is a path from u to v of length $\leq k$.

Proposition 4.4.6. Let D be a multidigraph. Let $\mathbf{w}$ be a walk of D. Then, $\mathbf{w}$ either is a path or contains a cycle (i.e., there exists a cycle of D whose arcs are arcs of $\mathbf{w}$).

All the proofs are the same as for multigraphs (occasionally easier).

4.5. Connectivity

We defined the relation "path-connected" for multigraphs in terms of paths or walks. For digraphs, this gets trickier, since the existence of a walk from u to v does not ensure the existence of a walk from v to u. So the existence of a walk from u to v does not define an equivalence relation.

There are two ways to "fix" this. One is to define **strong path-connectedness** to mean the existence of **both** walks:

Definition 4.5.1. Let D be a multidigraph. We define a binary relation $\simeq_D$ on the set $\operatorname{V}(D)$ as follows: For two vertices u and v of D, we set $u \simeq_D v$ if and only if there exists a walk from u to v and a walk from v to u.

This binary relation $\simeq_D$ is called **strong path-connectedness**.

Proposition 4.5.2. This relation $\simeq_D$ is an equivalence relation.

Proof. Straightforward using **a** $*$ **b**. □

Proposition 4.5.3. Let D be a multidigraph. Let u and v be two vertices of D. Then, $u \simeq_D v$ if and only if there exist a path from u to v and a path from v to u.

Proof. Just as for graphs. □

Definition 4.5.4. Let D be a multidigraph.

(a) The equivalence classes of the equivalence relation $\simeq_D$ are called the **strong components** of D.

(b) We say that D is **strongly connected** if D has exactly one strong component.

See Spring 2022 lecture 10 for examples.

There is also a weaker notion of connected components and connectivity:

Definition 4.5.5. Let D be a multidigraph. Consider its underlying undirected multigraph D^{und}.

(a) The components of this undirected multigraph D^{und} are called the **weak components** of D.

(b) We say that D is **weakly connected** if D has exactly one weak component (i.e., if D^{und} is connected).

Proposition 4.5.6. Any strongly connected digraph is weakly connected.

Let us look at what happens to walks, paths, circuits and cycles if we replace a graph G by its bidirectionalized digraph G^{bidir}:

Proposition 4.5.7. Let G be a multigraph. Then:

1. The walks of G are "more or less the same as" the walks of G^{bidir}. More precisely, each walk of G gives rise to a walk of G^{bidir} (with the same starting point and the same ending point), and conversely, each walk of G^{bidir} gives rise to a walk of G. If G has no loops, then this is a bijection.

2. The paths of G are "more or less the same as" the paths of G^{bidir}. This is an actual bijection.

3. The closed walks of G are "more or less the same as" the closed walks of G^{bidir}.

4. The cycles of G are not quite the same as the cycles of G^{bidir}. In fact, if e is an edge of G with two distinct endpoints u and v, then (u, e, v, e, u) is not a cycle of G, but $(u, (e,1), v, (e,2), u)$ or $(u, (e,2), v, (e,1), u)$ is a cycle of G^{bidir}. But it is still true that each cycle of G gives rise to a cycle of G^{bidir}.

4.6. Eulerian walks and circuits

Let us now define analogues of Eulerian walks and circuits for multidigraphs.

Definition 4.6.1. Let D be a multidigraph.

1. A walk of D is said to be **Eulerian** if each arc of D appears exactly once in this walk.

2. An **Eulerian circuit** of D means a circuit of D that is Eulerian.

The Euler–Hierholzer theorem gave us a nice and simple criterion for a multigraph to have an Eulerian circuit or walk. Something very similar works for multidigraphs:

Theorem 4.6.2 (diEuler, diHierholzer)**.** Let D be a weakly connected multidigraph. Then:

1. The multidigraph D has an Eulerian circuit if and only if each vertex v of D satisfies $\deg^+ v = \deg^- v$.

2. The multidigraph D has an Eulerian walk if and only if all but two vertices v of D satisfy $\deg^+ v = \deg^- v$, and the remaining two vertices v satisfy $\left|\deg^+ v - \deg^- v\right| \leq 1$.

Proof. Homework set #4. □

Incidentally, the condition "each vertex v of D satisfies $\deg^+ v = \deg^- v$" has a name:

Definition 4.6.3. A multidigraph D is said to be **balanced** if each vertex v of D satisfies $\deg^+ v = \deg^- v$.

Proposition 4.6.4. Let G be a multigraph. Then, the multidigraph G^{bidir} is balanced.

Corollary 4.6.5. Let G be a connected multigraph. Then, the multidigraph G^{bidir} has an Eulerian circuit. In other words, there is a circuit of G that contains each edge **exactly twice**, and uses it once in each direction.

Proof. Combine the last proposition with the last theorem. □

4.7. Hamiltonian cycles and paths

We can define Hamiltonian paths and cycles for simple digraphs just as we defined them for simple graphs:

Definition 4.7.1. Let $D = (V, A)$ be a simple digraph.

1. A **Hamiltonian path** of D means a walk of D that contains each vertex of D exactly once. Clearly, it is a path.
2. A **Hamiltonian cycle** of D means a cycle $(v_0, v_1, \ldots, v_k)$ of D such that each vertex of D appears exactly once among $v_0, v_1, \ldots, v_{k-1}$.

We will abbreviate Hamiltonian paths and cycles as **hamps** and **hamcs**.

What can we say about them?
Ore's theorem has the following analogue:

Theorem 4.7.2 (Meyniel)**.** Let $D = (V, A)$ be a strongly connected loopless simple digraph with n vertices. Assume that for each pair $(u, v) \in V \times V$ of two vertices u and v satisfying $u \neq v$ and $(u, v) \notin A$ and $(v, u) \notin A$, we have $\deg u + \deg v \geq 2n - 1$, where $\deg w = \deg^+ w + \deg^- w$. Then, D has a hamc.

I don't know of a simple proof, but I give references in the 2022 notes (Lecture 10). It is more complicated than Ore.

4.8. The reverse and complement digraphs

Definition 4.8.1. Let $D = (V, A)$ be a simple digraph. Then:

1. The elements of $(V \times V) \setminus A$ are called the **non-arcs** of D.

2. The **reversal** of a pair $(i, j) \in V \times V$ means the pair (j, i).

3. We define D^{rev} as the simple digraph $(V,\ A^{\text{rev}})$, where

$$A^{\text{rev}} = \{(j, i) \ \mid \ (i, j) \in A\}.$$

   Thus, D^{rev} is the digraph obtained from D by reversing each arc (i.e., swapping its source with its target). This is called the **reversal** of D.

4. We define $\overline{D}$ as the simple digraph $(V,\ (V \times V) \setminus A)$. This is called the **complement** of D. The arcs of $\overline{D}$ are the non-arcs of D.

We shall now try to count hamps in simple digraphs.

Convention 4.8.2. The symbol # means "number". For example, (# of subsets of $\{1, 2, 3\}$) $= 8$.

Proposition 4.8.3. Let D be the simple digraph (V, A), where

$$V = \{1, 2, \ldots, n\} \text{ for some } n \in \mathbb{N},$$

and

$$A = \{(i, j) \ \mid \ i < j\}.$$

Then, (# of hamps of D) $= 1$.

Proof. The only hamp of D is $(1, 2, \ldots, n)$. □

Proposition 4.8.4. Let D be a simple digraph. Then,

$$(\text{\# of hamps of } D^{\text{rev}}) = (\text{\# of hamps of } D).$$

Proof. Walking a hamp of D backwards gives a hamp of D^{rev}. And vice versa. So there is a bijection. □

Theorem 4.8.5 (Berge)**.** Let D be a simple digraph. Then,

$$\left(\text{\# of hamps of } \overline{D}\right) \equiv (\text{\# of hamps of } D) \bmod 2.$$

Proof: zoom/youtube this weekend?

4.9. Tournaments

Definition 4.9.1. A digraph D is said to be **loopless** if it has no loops.

Definition 4.9.2. A **tournament** is defined to be a loopless simple digraph D that satisfies the

- **Tournament axiom:** For any two distinct vertices u and v of D, exactly one of (u,v) and (v,u) is an arc of D.

Proposition 4.9.3. A simple digraph D is a tournament if and only if D^{rev} is $\overline{D}$ without the loops.

Theorem 4.9.4 (Easy Redei theorem)**.** A tournament always has at least one hamp.

Theorem 4.9.5 (Hard Redei theorem)**.** Let D be a tournament. Then,

$$(\# \text{ of hamps of } D) \text{ is odd.}$$

Clearly, the Hard Redei theorem implies the Easy one. The Easy one we proved on the blackboard. For the Hard one, see this weekend's lecture.

What about Hamiltonian cycles?

Not every tournament has a hamc: for example, the one constructed above (that has only 1 hamp) has no hamc. There is clearly a necessary condition:

Proposition 4.9.6. If a digraph D has a hamc, then D is strongly connected.

In general, this is only a necessary criterion, not a sufficient one. However, for tournaments, it actually is sufficient:

Theorem 4.9.7 (Camion's theorem)**.** If a tournament D is strongly connected and has at least two vertices, then D has a hamc.

Proof. See the Spring 2022 notes for more details, and the Spring 2017 notes for even more.

We pick a longest cycle $(v_1, v_2, \ldots, v_k, v_{k+1})$ of D. Assume that this is not a hamc.

We show that if w is a vertex not on this cycle, then w is either a **from-vertex** (i.e., we have an arc (w, v_i) for each i) or a **to-vertex** (i.e., we have an arc (v_i, w) for each i). Assume that our longest cycle is not a hamc. Then, there is at least one vertex that is a from-vertex or a to-vertex. Now:

1. If there exist from-vertices but not to-vertices, then we get a contradiction to strong connectedness of D, because there is no path from v_1 to our from-vertices.

2. If there exist to-vertices but not from-vertices, then we get a contradiction to strong connectedness of D, because there is no path from our to-vertices to v_1.

3. If there exist both from-vertices and to-vertices, then we argue as follows:

   - If there is some arc from a to-vertex t to a from-vertex f, then we can use this arc as a detour to make our cycle longer.
   - If no such arc exists, then there is no path from v_1 to a from-vertex (since the only arcs leading into a from-vertex are coming from from-vertices), which contradicts the strong connectedness of D.

In either case, we get a contradiction.

Except we are not quite done: We picked a longest cycle, which tacitly relies on the existence of a cycle. Why does D have a cycle?

We assumed that D is strongly connected and has at least 2 vertices; call them u and v. Hence, there is a walk from u to v and a walk from v to u. Combining these walks yields a circuit with at least one arc. By one of the propositions above, this circuit must contain a cycle. □

Note that a strongly connected tournament with at least two vertices must necessarily have at least three vertices.

4.10. A few words on finding paths

Given a digraph D. How do we efficiently find a path from a vertex u to a vertex v, or show that it does not exist?

If $u = v$, then clearly (u) works.

If $u \neq v$, then any path from u to v must have at least one arc and thus a second-to-last vertex. If w is this second-to-last vertex, then this path is a path from u to w followed by an arc from w to v. Hence, there exists a path from u to v if and only if there exists a path from u to w and an arc from w to v for some vertex w.

Better yet: There exists a path from u to v of length $\leq k$ if and only if there exists a path from u to w of length $\leq k-1$ and an arc from w to v for some vertex w.

Can you use this to obtain an efficient algorithm for finding paths (shortest paths even)? (Note that the length of a path is always $< |\operatorname{V}(D)|$).

We'll discuss the answer next time.

Lecture 8

Recall our problem from last time:

Given a multidigraph $D = (V, A, \psi)$. How do we efficiently find a path from a vertex u to a vertex v, or show that it does not exist? Even better, how do we find a shortest path (= a path of minimum length)?

The idea is to consider walks of a given length. For any two vertices $u, v \in V$ and any $k \in \mathbb{N}$, we let $W_k(u, v)$ be the set of all length-k walks from u to v. Then, it is easy to see that:

- We have
$$W_0(u,v) = \begin{cases} \{(u)\}, & \text{if } u = v; \\ \varnothing, & \text{if } u \neq v. \end{cases}$$

- For any positive integer k, we have
$$W_k(u,v) = \bigcup_{x \in V} \{\mathbf{w} * (x, a, v) \mid \mathbf{w} \in W_{k-1}(u, x) \text{ and } a \text{ is an arc } x \to v\}.$$

  In particular, a length-k walk $u \to v$ exists if and only if there is an $x \in V$ such that a length-$(k-1)$ walk $u \to x$ and an arc $x \to v$ exist.

This theoretically gives an algorithm for recursively computing all $W_k(u, v)$. (You need to recurse on k and compute $W_i(u, x)$ for all $i \leq k$ and all vertices x.) This algorithm tends to be slow because these sets $W_i(u, x)$ tend to be large. So it would be great if, instead of keeping track of all walks, we could just figure out which ones are the shortest, or, even better, just keep track of one shortest path for every i and u.

This is indeed possible: Instead of recursively computing $W_i(u, x)$ for all i and x, we can recursively choose a single walk of length i from u to x for each i and x. The recursion goes something like this:

```
def some_walk(u, v, i):
  if i == 0:
   if u == v:
    return (u)
   else:
    return None
  for x in V:
   if some_walk(u, x, i-1) is not None and arcs(x, v) is not None:
    return some_walk(u, x, i-1) * some_arc(x, v)
  return None
```

(This is Python pseudocode, not hard to implement.)

If we are looking for paths, not just walks of a given length, we just need to keep in mind that:

- A shortest walk is always a path. So if you look for a walk of length 0, then a walk of length 1, then a walk of length 2, and so on, then the first time you find such a walk you actually find a path.

- You don't have to look very far: Any path has length $\leq |V| - 1$. Thus, if you have not found a walk of length $\leq |V| - 1$, then you can stop searching; there is no walk.

Implemented well, this results in the **Bellman–Ford algorithm** for finding shortest paths. There is a variant of this algorithm that finds minimum-cost paths (aka minimum-weight paths). See the Wikipedia for more details.

5. Trees and arborescences

Trees are particularly nice graphs. Among other things, they can be characterized as

- the minimally connected graphs on a given set of vertices, or

- the maximally acyclic (= having no cycles) graphs on a given set of vertices, or

in many other ways.

Arborescences are directed analogues of trees.

We will study their theory and applications and answer some enumerative questions. See courses on TCS for many more applications (but be warned: in TCS, a tree is not quite the same as in combinatorics).

5.1. Some general properties of components and cycles

5.1.1. Backtrack-free walks revisited

Recall:

Definition 5.1.1. A walk in a multigraph G is said to be **backtrack-free** if no two adjacent edges of this walk are identical.

Proposition 5.1.2. Let G be a multigraph. Let $\mathbf{w}$ be a backtrack-free walk of G. Then, $\mathbf{w}$ either is a path or contains a cycle.

Proposition 5.1.3. Let G be a multigraph. Let u and v be two vertices of G. Assume that there are two distinct backtrack-free walks from u to v in G. Then, G has a cycle.

We proved these propositions for simple graphs at least. The proofs for multigraphs are more or less analogous.

5.1.2. Counting components

Definition 5.1.4. Let G be a multigraph. Then, $\operatorname{conn} G$ means the number of components of G.

This is also denoted by $b_0(G)$ in a homage to topologists.

So a multigraph G satisfies $\operatorname{conn} G = 1$ if and only if G is connected.

Recall:

Theorem 5.1.5. Let G be a multigraph. Let e be an edge of G. Then:

(a) If e is an edge of some cycle of G, then the components of $G \setminus e$ are precisely the components of G.

(b) If e appears is no cycle of G, then the graph $G \setminus e$ has one more component than G.

Again, we proved this for simple graphs, but the same proof applies to multigraphs mutatis mutandis.

Corollary 5.1.6. Let G be a multigraph. Let e be an edge of G. Then:

(a) If e is an edge of some cycle of G, then $\operatorname{conn}(G \setminus e) = \operatorname{conn} G$.

(b) If e appears in no cycle of G, then $\operatorname{conn}(G \setminus e) = \operatorname{conn} G + 1$.

(c) In either case, we have $\operatorname{conn}(G \setminus e) \leq \operatorname{conn} G + 1$.

Corollary 5.1.7. Let $G = (V, E, \varphi)$ be a multigraph. Then, $\operatorname{conn} G \geq |V| - |E|$.

Proof. Let's remove all edges of G and then add them back in one by one.

At first, G has no edges and V vertices, thus $|V|$ many components.

Then, we add the first edge, and obtain $\geq |V| - 1$ many components (by part **(c)** of the previous corollary).

Then, we add the second edge, and obtain $\geq |V| - 2$ many components (again by that part **(c)**).

And so on. At the end, we have added back all $|E|$ many edges, and obtained $\geq |V| - |E|$ many components. In other words, $\operatorname{conn} G \geq |V| - |E|$.

(To make it rigorous, you induct on $|E|$. See Spring 2022 Lecture 13 Corollary 1.1.7 for this.) □

Corollary 5.1.8. Let $G = (V, E, \varphi)$ be a multigraph that has no cycles. Then, $\operatorname{conn} G = |V| - |E|$.

Proof. Replay we the previous proof, but now using part **(b)** of the first corollary instead of part **(c)**, since we know that all the edges appear in no cycles. So we get an $=$ sign instead of a $\geq$ sign. □

Corollary 5.1.9. Let $G = (V, E, \varphi)$ be a multigraph that has at least one cycle. Then, $\operatorname{conn} G \geq |V| - |E| + 1$.

Proof. Again, replay the old proof, but now make sure that the last edge to be added back in is an edge of a cycle. Then, when we add that edge in, it just completes a cycle, so the number of component does not grow.

(See Spring 2022 for details.) □

Let's combine these corollaries into a theorem:

Theorem 5.1.10. Let $G = (V, E, \varphi)$ be a multigraph. Then:

(a) We have $\operatorname{conn} G \geq |V| - |E|$.

(b) We have $\operatorname{conn} G = |V| - |E|$ if and only if G has no cycles.

Remark 5.1.11. The number $\operatorname{conn} G - (|V| - |E|)$ is called the **cyclomatic number** of G. It does **not** determine the number of cycles of G, but as we just saw it is 0 if and only if G has no cycles.

5.2. Forests and trees

Definition 5.2.1. A **forest** is a multigraph that has no cycles.

(In particular, it has no parallel edges and no loops.)

Definition 5.2.2. A **tree** is a connected forest.

In particular, the empty graph with no vertices is a forest but not a tree.

Trees can be described in many equivalent ways:

Theorem 5.2.3 (tree equivalence theorem)**.** Let $G = (V, E, \varphi)$ be a multigraph. Then, the following eight statements are equivalent:

- **T1:** The multigraph G is a tree.
- **T2:** The multigraph G has no loops, and we have $V \neq \varnothing$, and for each $u, v \in V$, there is a **unique** path from u to v.
- **T3:** We have $V \neq \varnothing$, and for each $u, v \in V$, there is a **unique** backtrack-free walk from u to v.

- **T4:** The multigraph G is connected, and we have $|E| = |V| - 1$.
- **T5:** The multigraph G is connected, and $|E| < |V|$.
- **T6:** We have $V \neq \varnothing$, and the graph G is a forest, but adding any new edge to G creates a cycle.
- **T7:** The multigraph G is connected, but removing any edge from G yields a disconnected (i.e., non-connected) graph.
- **T8:** The multigraph G is a forest, and we have $|E| \geq |V| - 1$ and $V \neq \varnothing$.

Proof. (These are just sketches; see 2022 Lecture 13 for details.)

T4$\Longrightarrow$T5 is obvious. T5$\Longrightarrow$T4 follows from $\operatorname{conn} G \geq |V| - |E|$. T3$\Longrightarrow$T2 is easy. T1$\Longrightarrow$T4 follows from $\operatorname{conn} G = |V| - |E|$ when G has no cycles. T4$\Longrightarrow$T1 follows from the converse of that statement. T4$\Longrightarrow$T8 via T1. We have T7$\Longrightarrow$T1 because an edge of a cycle could be removed without disconnecting G. We have T8$\Longrightarrow$T1 because $\operatorname{conn} G = |V| - |E| \leq 1$ and thus $\operatorname{conn} G = 1$. We have T2$\Longrightarrow$T1 because a cycle is either a loop or has two different paths between two vertices. We have T1$\Longrightarrow$T3 by the second proposition today. We have T4$\Longrightarrow$T6, because adding any new edge turns $|E| = |V| - 1$ into $|E| = |V|$. Similarly, T4$\Longrightarrow$T7. Similarly, T6$\Longrightarrow$T1. □

Remark 5.2.4. Let $G = (V, E, \varphi)$ be a multigraph.

(a) If G is a forest, then $|E| \leq |V| - 1$ (unless $V = \varnothing$).

(b) If G is connected, then $|E| \geq |V| - 1$.

(c) If G is a tree, then $|E| = |V| - 1$.

So trees live in the goldilocks zone where $|E| = |V| - 1$. As a consequence, adding an edge to a tree or removing an edge from a tree breaks the treeness.

For comparison: Let $v_1, v_2, \ldots, v_k$ be a bunch of vectors in a vector space V. Then:

(a) If the vectors $v_1, v_2, \ldots, v_k$ are linearly independent, then $k \leq \dim V$.

(b) If the vectors $v_1, v_2, \ldots, v_k$ span V, then $k \geq \dim V$.

(c) If the vectors $v_1, v_2, \ldots, v_k$ form a basis of V, then $k = \dim V$.

This is not just an analogy. Consider a multigraph $G = (V, E, \varphi)$, where $V = \{1, 2, \ldots, n\}$. Model each edge $e \in E$ as a vector

$$\left(0, 0, \ldots, 0, \underbrace{1}_{\text{position } i}, 0, 0, \ldots, 0, \underbrace{-1}_{\text{position } j}, 0, 0, \ldots, 0\right) \in \mathbb{R}^n,$$

where i and j are the endpoints of e. Thus, each edge of G becomes a vector. Now:

(a) These vectors are linearly independent if and only if G is a forest.

(b) These vectors span $\{x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n \mid x_1 + x_2 + \cdots + x_n = 0\}$ if and only if G is connected.

(c) These vectors form a basis of $\{x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n \mid x_1 + x_2 + \cdots + x_n = 0\}$ if and only if G is a tree.

So you can obtain the graph-theoretical claims from the linear-algebraic ones.

As we said, trees are connected forests. Conversely, forests are made out of trees:

Proposition 5.2.5. Let G be a multigraph, and let $C_1, C_2, \ldots, C_k$ be its components. Then, G is a forest if and only if the induced subgraphs $G[C_1], G[C_2], \ldots, G[C_k]$ are trees.

Proof. Straightforward (see 2022 Lecture 13 for details). □

5.3. Leaves

Definition 5.3.1. Let T be a tree. A vertex of T is said to be a **leaf** if its degree is 1.

How to find a tree with as many leaves as possible (for a given number of vertices)? That's the star graph: For any $n \geq 3$, the simple graph

$$(\{0, 1, \ldots, n-1\}, \{0i \mid i > 0\})$$

is a tree (when considered as a multigraph), and has $n-1$ leaves $(1, 2, \ldots, n-1)$.

How to find a tree with as few leaves as possible? The path graph: For any $n \geq 2$, the n-path graph $P_n = (1 - 2 - 3 - \cdots - n)$ has only 2 leaves (1 and n).

For $n \geq 2$, this is the minimum possible number of leaves:

Theorem 5.3.2. Let T be a tree with at least 2 vertices. Then:

(a) The tree T has at least 2 leaves.

(b) Let v be a vertex of T. Then, there exist two distinct leaves p and q of T such that v lies on the path from p to q.

(I say "the path" since T is a tree and thus has only one path from p to q.)

Proof. **(a)** follows from **(b)**, so we only need to show **(b)**.

(b) Pick a longest path of T passing through v. Let p and q be its starting and ending points. I claim that p and q are leaves (and it is clear that p and q are distinct, since the path is not just (v)).

Indeed, assume the contrary. Let's say p is not a leaf (the other case is analogous). Then, there exists an edge e containing p that is not part of the path. Extending the path by this edge e, we obtain a backtrack-free walk, thus a path (since T is a tree). That path is longer than the one we started with. Contradiction!

(See the 2022 notes – Lecture 14 – for more details and for a different proof of part **(a).)** □

Lecture 9

Recall:

Definition 5.3.3. Let T be a tree. A vertex of T is said to be a **leaf** if its degree is 1.

Theorem 5.3.4. Let T be a tree with at least 2 vertices. Then:

(a) The tree T has at least 2 leaves.

(b) Let v be a vertex of T. Then, there exist two distinct leaves p and q of T such that v lies on the path from p to q.

We proved this last time. Now let us see why leaves are so important to trees:

Theorem 5.3.5 (induction principle for trees)**.** Let T be a tree with at least 2 vertices. Let v be a leaf of T. Let $T \setminus v$ be the multigraph obtained from T by removing v and all edges that contain v. Then, $T \setminus v$ is again a tree.

Proof. Since T has at least 2 vertices, $T \setminus v$ has at least 1 vertex.

Furthermore, $T \setminus v$ has no cycles (since T has none), thus is a forest.

It remains to show that $T \setminus v$ is connected.

Consider the unique edge e containing v. Removing this edge e breaks T into 2 components (since e is clearly not on any cycle). One component contains v and only v (since v has no other edges containing it). Thus, if we remove v as well, only the other component remains. Therefore, $T \setminus v$ has just that one component, i.e., is connected.

(See Spring 2022 Lecture 14 Theorem 1.2.3 for a different proof.) □

The theorem also has a converse:

Theorem 5.3.6. Let G be a multigraph. Let v be a vertex of G such that $\deg v = 1$ and such that $G \setminus v$ is a tree. Then, G is a tree.

Proof. Left to the reader. □

The above two theorems reveal a recursive structure behind trees.

5.4. Spanning trees

First, we define the notion of "spanning", which makes sense for any kind of graphs:

Definition 5.4.1. A **spanning subgraph** of a multigraph $G = (V, E, \varphi)$ means a multigraph of the form $(V, F, \varphi \mid_F)$, where F is a subset of E.

In other words, it means a submultigraph of G with the same vertex set as G.

Informally, it means a multigraph obtained from G by removing some edges but not removing any vertices.

For comparison:

- A subgraph of G can choose which vertices and which edges of G it contains.
- An induced subgraph chooses its vertices, but has to contain all the edges it can.
- A spanning subgraph chooses its edges, but has to contain all the vertices.

Spanning subgraphs are particularly useful when they are trees:

Definition 5.4.2. A **spanning tree** of a multigraph G means a spanning subgraph of G that is a tree.

Example 5.4.3. (See the blackboard or 2022 Lecture 14.)

A spanning tree of a graph G can be regarded as a minimum "backbone" of G – that is, a way to keep G connected using as few edges as possible. Of course, if G is not connected, this is not possible at all; the best you can then do is the following:

Definition 5.4.4. A **spanning forest** of a multigraph G means a spanning subgraph H of G that is a forest and satisfies $\operatorname{conn} H = \operatorname{conn} G$.

When G is a connected multigraph, a spanning forest of G is the same as a spanning tree of G.

The following theorem is crucial:

Theorem 5.4.5. Each connected multigraph G has at least one spanning tree.

First proof. Let G be a connected multigraph.

Work top-to-bottom: Start with G itself, and keep removing edges until you're left with a spanning tree.

More precisely: As long as G has a non-bridge (i.e., an edge that is part of a cycle), you can remove this edge without disconnecting G. Keep doing this (take care to never remove two non-bridges simultaneously!). Eventually, you end up with a spanning subgraph of G that has no non-bridges left, i.e., has no cycles left. But by its construction, it is connected. So it is a connected graph with no cycles, i.e., a tree. And it is a spanning subgraph of G. So it is a spanning tree of G. □

Second proof. Let G be a connected multigraph.

Work bottom-to-top: Let L be the graph with the same vertices as G but with no edges. Keep adding edges from G to L whenever needed to merge two components.

Here are the details: At the beginning, L has all the necessary vertices but no edges. Now, go over all edges of G one by one. Add each edge to L if it does not create a cycle (= if it merges two components); otherwise skip it.

The resulting graph L at the end of this procedure is a spanning tree of G. Why?

We certainly created no cycles, so the resulting graph L is a forest. And it is a spanning subgraph of G. Remains to show that it is connected.

Well: If not, then there is an edge e of G that could be added to L without creating a cycle (since it would join two components). But we have already digested this edge e and found it useless since it would have created a cycle at the time of its digestion. Contradiction, since the graph L at that time was a subgraph of the final graph L. □

Third proof. We construct a spanning tree of G by starting a rumor at some vertex r of G, and watching it spread across the edges.

In more detail: The rumor starts at vertex r. On day 0, only r knows the rumor. Each day, every vertex that knows this rumor spreads it to all its neighbors. Since G is connected, the rumor will eventually reach all vertices. Now, each vertex $v \neq r$ must have heard the rumor first from some neighbor v' (if there are ties, break them arbitrarily). Pick some edge that joins with v and v' and call it e_v; let's say that this edge e_v is "the edge through which v has learned the rumor".

I claim that the set of all these edges e_v for all vertices $v \neq r$ forms a spanning tree of G. (More precisely: If $G = (V, E, \varphi)$ and $F = \{e_v \mid v \neq r\}$, then $(V, F, \varphi\mid_F)$ is a spanning tree of G.)

Intuitively, this is because:

- We can use these F-edges (i.e., the edges e_v) to trace back the rumor from each vertex v to r. Thus, $(V, F, \varphi\mid_F)$ is connected.
- We have $|F| \leq |V| - 1$.
- Combining these, we conclude that $(V, F, \varphi\mid_F)$ is a tree (by the tree equivalence theorem).

(An alternative proof is in the 2022 notes, Lecture 14, third proof of Theorem 1.3.5.)

This tree $(V, F, \varphi\mid_F)$ has an extra feature:

> For each $k \in \mathbb{N}$, any vertex of G that has distance k from r in G also has distance k from r in $(V, F, \varphi\mid_F)$.

Note that this is only true for distances from r. Distances between two random vertices usually grow when we replace G by a spanning tree.

This tree $(V, F, \varphi \mid_F)$ is called the **breadth-first search tree ("BFS tree")** of G. □

Fourth proof (sketched). We imagine a snake that slithers along the edges of G, trying to eventually bite each vertex. It starts at some vertex r, which it immediately bites. Any time the snake enters a vertex v, it makes the following step:

- If some neighbor of v has not been bitten yet, then the snake picks such a neighbor w as well as some edge f that joins w with v; the snake then moves to w along this edge f, bites the vertex w and marks the edge f.
- If not, then the snake marks the vertex v as fully digested and backtracks (along the marked edges) to the last vertex it has visited but not yet fully digested.

Once backtracking is no longer possible (because there are no more vertices on the snake's path that are not fully digested), the snake stops, and the marked edges form a spanning tree of G.

(See examples on blackboard.)

One way to prove that they form a spanning tree is by showing the following observations:

1. After each step, the marked edges are precisely the edges along which the snake has moved so far.
2. After each step, the network of bitten vertices and marked edges is a tree.
3. After enough steps, each bitten vertex is fully digested.
4. At that point, the network of bitten vertices and marked edges is a spanning tree (since each neighbor of a fully digested vertex is bitten, and thus also fully digested by observation 3).

Details (not completely trivial) are left to the reader.

The spanning tree T obtained by this algorithm is called a **depth-first search tree ("DFS tree")** of G. It has the following extra property: If u and v are two adjacent vertices of G, then either u lies on the path from r to v in T, or v lies on the path from r to u in T. (This is called a "lineal spanning tree".) □

Spanning trees have lots of applications:

- A spanning tree of a graph can be viewed as a kind of "backbone" of that graph, which connects any two vertices in an unambiguous way. This is often used when information has to go from one vertex to another irredundantly (e.g. the "spanning tree protocol").

- Spanning trees with extra properties are often valued for those properties. For example, if c is a cost function on the edges (i.e., if each edge e of the graph G has a cost $c(e)$ attached to it), then we can look for a spanning tree of smallest total cost (i.e., the sum of the costs of its edges should be as small as possible). Nicely enough, such a minimum-cost spanning tree can be constructed by the algorithm given in the second of our above proofs: You just have to go over the edges in the order of increasing cost. (The proof that the resulting tree has minimum cost will be on homework set #6.)
- Depth-first search (the fourth proof above) can be used as a way to traverse all vertices of a graph and return to the starting point. More usefully, this method is "local": each step only requires knowledge of the neighbors of the vertex where you're at. Thus, it can be used as an algorithm to solve mazes.

Here is a theoretical application of spanning trees (one of many):

Definition 5.4.6. A vertex v of a connected multigraph G is called a **cut-vertex** if $G \setminus v$ is disconnected.

Proposition 5.4.7. Any connected multigraph with ≥ 2 vertices has at least 2 vertices that are not cut-vertices.

Proof. Pick a spanning tree of our multigraph. The spanning tree has ≥ 2 leaves. These leaves cannot be cut-vertices of the original graph. □

This proposition can be used to prove properties of connected graphs by induction on the number of vertices.

A few words on disconnected graphs:

Corollary 5.4.8. Each multigraph has a spanning forest.

Proof. Pick a spanning tree of each component, and take their union. □

5.5. Centers of graphs and trees

Definition 5.5.1. Let G be a multigraph.

For any two vertices u and v of G, we define the **distance** between u and v is the smallest length of a path from u to v. If no such path exists, then this distance is defined to be ∞.

The distance between u and v is called $d(u,v)$ or $d_G(u,v)$.

Remark 5.5.2. Distances in a multigraph satisfy the axioms of a metric space, except that they can be ∞. In particular,

$$d(u,v) + d(v,w) \geq d(u,w) \qquad \text{for all } u,v,w \in \operatorname{V}(G).$$

Also, the notion of a distance does not change if we replace "path" by "walk".

In a tree, we can replace "the smallest length of a path" by "the length of the path" in the definition of a distance (since there is only one path from u to v).

Definition 5.5.3. Let v be a vertex of a multigraph $G = (V, E, \varphi)$. The **eccentricity** of v (with respect to G) is defined to be the number

$$\operatorname{ecc}_G v := \max\{d(v,u) \mid u \in V\} \in \mathbb{N} \cup \{\infty\}.$$

Definition 5.5.4. Let $G = (V, E, \varphi)$ be a multigraph. Then, a **center** of G means a vertex of G whose eccentricity is minimum (among all vertices).

Theorem 5.5.5. Let T be a tree. Then:

1. The tree T has either 1 or 2 centers.
2. If T has two centers, then these two centers are adjacent.
3. Moreover, these centers can be constructed by the following algorithm:

   If T has more than 2 vertices, remove all leaves from T. If the resulting tree still has more than 2 vertices, remove all leaves from it. Keep doing this, until you are left with at most 2 vertices. Those remaining vertices are the centers of T.

Proof. Spring 2022 Lecture 15. □

Lecture 10

A quick reminder about trees and their centers:

- A **center** of a tree is a vertex with minimum eccentricity (i.e., distance from the farthest-away vertex). In Spring 2022 Lecture 15, you can find a proof that any tree has 1 or 2 centers only, and if there are 2, they are adjacent.
- A **centroid** of a tree is a vertex with minimum side-size (i.e., largest size of a component obtained from the tree by removing this vertex). It can be shown that any tree has 1 or 2 centroids only, and if there are 2, they are adjacent.

I am wondering why these two different concepts have such similar properties – is there a common generalization?

5.6. Arborescences

Time to return to directed graphs.

What is the best directed analogue of a tree? I.e., what digraphs play the same role as trees but for digraphs? Here are some attempts at defining such an analogue:

- We can study digraphs that are strongly connected and have no cycles. There is only such digraph up to isomorphism, namely the 1-vertex digraph with no arcs.
- We can drop the connectedness requirement. Digraphs that have no cycles are called **acyclic**, or, for short, **dags**. Unfortunately, they are not quite like trees, even if you additionally require weak connectedness.

Here is a more convincing analogue of trees for digraphs:

Definition 5.6.1. Let D be a multidigraph. Let r be a vertex of D.

1. We say that r is a **from-root** (or, short, **root**) of D if for each vertex v of D, the digraph D has a path from r to v.
2. We say that D is an **arborescence rooted from** r (short: **arb from** r) if r is a from-root of D and the undirected multigraph D^{und} has no cycles.

Of course, there are analogous notions of "to-roots" and "arbs to r", and their properties are analogous because we can just reverse all the arcs (more on that later).

Example 5.6.2. (Blackboard, or Spring 2022 Lecture 15)

We observe that an arborescence rooted from r is basically the same as a tree whose all edges have been "oriented away from r". More precisely:

Theorem 5.6.3. Let D be a multidigraph, and let r be a vertex of D. Then, the following two statements are equivalent:

- **C1:** The multidigraph D is an arb from r.
- **C2:** The undirected multigraph D^{und} is a tree, and each arc of D is "oriented away from r", which means that the source of this arc lies on the unique path between r and the target of this arc in D^{und}.

Proving this is sufficiently nontrivial that I will do this after a few preparatory results.

First, however, let's prove a bunch of equivalent criteria for arborescences, similarly to the tree equivalence theorem:

Theorem 5.6.4 (arborescence equivalence theorem)**.** Let $D = (V, A, \psi)$ be a multidigraph with a from-root r. Then, the following six statements are equivalent:

- **A1:** The multidigraph D is an arb from r. (This means that D^{und} has no cycles.)
- **A2:** We have $|A| = |V| - 1$.
- **A3:** The multidigraph D^{und} is a tree.
- **A4:** For each vertex $v \in V$, the multidigraph D has a unique walk from r to v.
- **A5:** If we remove any arc from D, then r will no longer be a from-root.
- **A6:** We have $\deg^- r = 0$, and each $v \in V \setminus \{r\}$ satisfies $\deg^- v = 1$.

Proof. We have A1$\Longrightarrow$A3 easily (connectedness from from-root) and A3$\Longleftrightarrow$A2 (by the tree equivalence theorem) and A3$\Longrightarrow$A1. Thus, A1$\Longleftrightarrow$A2$\Longleftrightarrow$A3.

We have A3$\Longrightarrow$A4 (since D has no loops, so that walks in D are backtrack-free walks in D^{und}, but the latter are unique since D^{und} has no cycles). We have A2$\Longrightarrow$A5 (since removing an arc breaks the equality $|A| = |V| - 1$). We have A4$\Longrightarrow$A6 (see blackboard). We have A6$\Longrightarrow$A2.

It remains to prove A5$\Longrightarrow$A6. Assume A5. It is easy to see that $\deg^- r = 0$ (since an arc with target r is useless for making r a from-root). It remains to show that $\deg^- v = 1$ for each $v \in V \setminus \{r\}$. So let $v \in V \setminus \{r\}$. Clearly, $\deg^- v \geq 1$.

See Spring 2022 Lecture 15 for the rest. □

To get closer to the proof of the first theorem we claimed today, let us define a few more things and prove a few more lemmas:

Proposition 5.6.5. Let $T = (V, E, \varphi)$ be a tree. Let $r \in V$ be a vertex of T. Let e be an edge of T, and let u and v be its two endpoints.

Then, the distances $d(r, u)$ and $d(r, v)$ differ by exactly 1. That is, we have $d(r, u) = d(r, v) + 1$ or $d(r, v) = d(r, u) + 1$.

Proof. Consider the path $\mathbf{p}$ from r to u and the path $\mathbf{q}$ from r to v. If $\mathbf{p}$ does not contain e, then we can attach e and v to $\mathbf{p}$ and obtain $\mathbf{q}$, so we get $d(r, v) = d(r, u) + 1$. If $\mathbf{p}$ does contain e, then $\mathbf{p}$ must end with e, so that we can remove e and u from $\mathbf{p}$ and obtain q, so we get $d(r, u) = d(r, v) + 1$. □

Definition 5.6.6. Let $T = (V, E, \varphi)$ be a tree. Let $r \in V$ be a vertex of T. Let e be an edge of T. As we just showed, the distances from r to the endpoints of e differ by 1. So one of them is smaller than the other.

1. We define the ***r*-parent** of e to be the endpoint of e whose distance from r is smaller than the other. We call it e^{-r}.

2. We define the ***r*-child** of e to be the endpoint of e whose distance from r is larger than the other. We call it e^{+r}.

Definition 5.6.7. Let $T = (V, E, \varphi)$ be a tree. Let $r \in V$ be a vertex of T. Then, we define a multidigraph $T^{r\rightarrow}$ by

$$T^{r\rightarrow} := (V, E, \psi),$$

where $\psi : E \to V \times V$ is the map that sends each edge $e \in E$ to the pair (e^{-r}, e^{+r}). Colloquially, this means that $T^{r\rightarrow}$ is the multidigraph obtained from T by turning each edge e into an arc from its r-parent e^{-r} to its r-child e^{+r} (that is, "orienting it away from r").

Now, we can rewrite the theorem whose proof we still owe as follows:

Theorem 5.6.8. Let D be a multidigraph, and let r be a vertex of D. Then, the following two statements are equivalent:

- **C1:** The multidigraph D is an arb from r.
- **C2:** The undirected multigraph D^{und} is a tree, and we have $D = \left(D^{\text{und}}\right)^{r\rightarrow}$.

We are still not quite ready to prove it. Two lemmas:

Lemma 5.6.9. Let $T = (V, E, \varphi)$ be a tree. Let $r \in V$ be a vertex of T. Then, the multidigraph $T^{r\rightarrow}$ is an arb from r.

Proof. All we need to show is that r is a from-root of $T^{r\rightarrow}$.

Let v be a vertex of T. Then, T has a path $\mathbf{p}$ from r to v. As we walk along this path, the distance from the root increases by 1 at each step. This means that this path crosses each edge in the parent-to-child direction. Thus, it is also a path in the digraph $T^{r\rightarrow}$. Hence, $T^{r\rightarrow}$ has a path from r to v. This shows that r is a from-root of $T^{r\rightarrow}$. □

Lemma 5.6.10. Let $D = (V, A, \psi)$ be an arb from r. Let $a \in A$ be an arc of D. Let s be the source of a, and let t be the target of a. Then:

1. We have $d(r, s) < d(r, t)$, where d means the distance on the tree D^{und}.
2. In the multidigraph $\left(D^{\text{und}}\right)^{r\rightarrow}$, the arc a has source s and target t.

Proof. For Claim 1, we observe that the unique walk from r to t must use the arc a (otherwise, a would be useless and could be thrown away), so the unique path from r to t in the tree D^{und} must use the edge a. Hence, s is on this path. Thus, $d(r, s) < d(r, t)$. Therefore, Claim 2 follows. □

Now we can finally prove the theorem:

Theorem 5.6.11. Let D be a multidigraph, and let r be a vertex of D. Then, the following two statements are equivalent:

- **C1:** The multidigraph D is an arb from r.
- **C2:** The undirected multigraph D^{und} is a tree, and we have $D = \left(D^{\text{und}}\right)^{r\rightarrow}$.

Proof. *Proof of C1⟹C2:* Assume C1. Then, D^{und} is a tree (by the arborescence equivalence theorem). To show that $D = \left(D^{\text{und}}\right)^{r\rightarrow}$, we must prove that after we forget the directions of the arcs of D and recover them again using the parent-to-child orientation, we actually find the original directions. But this is what the last lemma claims. Thus, C2 follows.

Proof of C2⟹C1: Assume C2. We must prove C1.

Our second-to-last lemma says that $\left(D^{\text{und}}\right)^{r\rightarrow}$ is an arb from r (since D^{und} is a tree). Since $D = \left(D^{\text{und}}\right)^{r\rightarrow}$, we conclude that D is an arb from r. Thus, C1 follows. □

So far we have been considering r as fixed. A digraph can have many from-roots (or none). However, an arborescence can only have one:

Proposition 5.6.12. Let D be an arb from r. Then, r is the **only** from-root of D.

Proof. Statement A6 in the arborescence equivalence theorem reveals that the from-root of an arb is the unique vertex having indegree 0. So it is unique. □

Definition 5.6.13. An **arborescence** means a multidigraph D that is an arb from r for some vertex r. This r is unique (as we just showed), and we shall call it the **root** of this arborescence.

Theorem 5.6.14. There are two mutually inverse maps

$$\{\text{pairs } (T,r) \text{ of a tree } T \text{ and a vertex } r \text{ of } T\} \to \{\text{arborescences}\},$$
$$(T,r) \mapsto T^{r\to}$$

and

$$\{\text{arborescences}\} \to \{\text{pairs } (T,r) \text{ of a tree } T \text{ and a vertex } r \text{ of } T\},$$
$$D \mapsto \left(D^{\text{und}}, \text{ root of } D\right).$$

So an arborescence is "the same as" a tree T equipped with a chosen vertex r.

Proof. Follows easily from the above results. □

5.7. Spanning arborescences

In analogy to spanning subgraphs of a multigraph, we can define spanning subdigraphs of a multidigraph:

Definition 5.7.1. A **spanning subdigraph** of a multidigraph $D = (V, A, \psi)$ means a multidigraph of the form $(V, B, \psi\mid_B)$ for some subset B of A.

So a spanning subdigraph D has all the vertices of D but may miss some of the arcs.

Definition 5.7.2. Let D be a multidigraph. Let r be a vertex of D. A **spanning arborescence of D rooted from r** (short: **sparb of D from r**) means a spanning subdigraph of D that is an arborescence rooted from r.

Theorem 5.7.3. Let D be a multidigraph. Let r be a from-root of D. Then, D has a spanning arborescence rooted from r.

Note that this is an actually an "if and only if": If D has a spanning arborescence rooted from r, then r must be a from-root of D, since you can use the arborescence to get from r to any vertex of D.

Proof of the theorem. We proved the analogous property of undirected graphs in four different ways. The first proof definitely generalizes to arborescences. I am pretty sure the second proof does not. I think the third does. Does the fourth? □

5.8. The BEST theorem

Recall that a multidigraph D is **balanced** if and only if each vertex v satisfies $\deg^- v = \deg^+ v$. A weakly connected balanced multidigraph has a Eulerian circuit (by a homework set #4 problem, IIRC).

Surprisingly, there is a formula for the number of these Eulerian circuits:

Theorem 5.8.1 (The BEST theorem)**.** Let $D = (V, A, \psi)$ be a balanced multidigraph such that each vertex has indegree > 0. Fix an arc a of D, and let r be its target. Let $\tau(D, r)$ be the number of spanning arborescences of D rooted from r. Let $\varepsilon(D, a)$ be the number of Eulerian circuits of D whose last arc is a. Then,

$$\varepsilon(D, a) = \tau(D, r) \cdot \prod_{u \in V} \left(\deg^- u - 1\right)!.$$

BEST = de **B**ruijn, van Aardenne-**E**hrenfest, **S**mith and **T**utte.
Next time: a proof of this theorem.
Afterwards: a way to compute $\tau(D, r)$.

Lecture 11

Recall from last time:

Theorem 5.8.2 (The BEST theorem)**.** Let $D = (V, A, \psi)$ be a balanced multidigraph such that each vertex has indegree > 0. Fix an arc a of D, and let r be its target. Let $\tau(D, r)$ be the number of spanning arborescences of D rooted from r. Let $\varepsilon(D, a)$ be the number of Eulerian circuits of D whose last arc is a. Then,

$$\varepsilon(D, a) = \tau(D, r) \cdot \prod_{u \in V} \left(\deg^- u - 1\right)!.$$

Today we will prove this theorem.

The best way to visualize this proof involves first reversing all the arcs of D. So let me introduce "reverse versions" of the existing terminology for arborescences:

Definition 5.8.3. Let D be a multigraph. Let r be a vertex of D.

1. We say that r is a **to-root** of D if for each vertex v of D, the digraph D has a path from v to r.
2. We say that D is an **arborescence rooted to** r (short: **arb to** r) if r is a to-root of D and the undirected multigraph D^{und} has no cycles.

In analogy to the equivalence theorem for arbs from r, there is an equivalence theorem for arbs to r:

Theorem 5.8.4 (arborescence equivalence theorem, opposite version)**.** Let $D = (V, A, \psi)$ be a multidigraph with a to-root r. Then, the following six statements are equivalent:

- **A'1:** The multidigraph D is an arb to r. (This means that D^{und} has no cycles.)
- **A'2:** We have $|A| = |V| - 1$.
- **A'3:** The multidigraph D^{und} is a tree.
- **A'4:** For each vertex $v \in V$, the multidigraph D has a unique walk from v to r.
- **A'5:** If we remove any arc from D, then r will no longer be a to-root.
- **A'6:** We have $\deg^+ r = 0$, and each $v \in V \setminus \{r\}$ satisfies $\deg^+ v = 1$.

You can translate between arbs from r and arbs to r by reversing the direction of each arc:

Definition 5.8.5. Let $D = (V, A, \psi)$ be a multidigraph. Then, D^{rev} shall denote the multidigraph $(V, A, \tau \circ \psi)$, where $\tau : V \times V \to V \times V$ is the map that sends each pair (s, t) to (t, s). So any arc of D with source s and target t becomes an arc with source t and target s in D^{rev}.

Now, any walk or path of D can be reversed and thus becomes a walk or path of D^{rev}. Thus, to-roots of D become from-roots of D^{rev} and vice versa. Arbs from r become arbs to r, and vice versa.

We can use this dictionary to translate the BEST theorem:

Theorem 5.8.6 (The WORST theorem)**.** Let $D = (V, A, \psi)$ be a balanced multidigraph such that each vertex has outdegree > 0. Fix an arc a of D, and let r be its source. Let $\tau(D, r)$ be the number of spanning arborescences of D rooted to r. Let $\varepsilon(D, a)$ be the number of Eulerian circuits of D whose first arc is a. Then,

$$\varepsilon(D, a) = \tau(D, r) \cdot \prod_{u \in V} \left(\deg^+ u - 1\right)!.$$

The WORST theorem is equivalent to the BEST one, but is a bit easier to prove since it is more natural to walk forward than to walk backwards.

Proof of the WORST theorem. An a-**Eulerian circuit** will mean an Eulerian circuit of D whose first arc is a.

Let $\mathbf{e}$ be an a-Eulerian circuit. Its first arc is a; thus, its first and last vertex is r.

Being a Eulerian circuit, $\mathbf{e}$ must contain each arc of D and therefore each vertex of D (since each vertex of D has outdegree > 0). For each vertex $u \neq r$, we let $e(u)$ be the **last exit** of $\mathbf{e}$ from u, that is, the last arc of $\mathbf{e}$ that has source u.

Let $\operatorname{Exit} \mathbf{e}$ denote the set of these last exits $e(u)$ for all vertices $u \neq r$. Then we claim:

> *Claim 1:* This set $\operatorname{Exit} \mathbf{e}$ (or, more precisely, the corresponding spanning subdigraph $(V, \operatorname{Exit} \mathbf{e}, \psi \mid_{\operatorname{Exit} \mathbf{e}})$) is a sparb of D to r (this is short for "spanning arborescence of D rooted to r").

Let's assume for a moment that Claim 1 is proven. Thus, given any a-Eulerian circuit $\mathbf{a}$, we have constructed a sparb of D to r, which we will just call $\operatorname{Exit} \mathbf{e}$. (For any subset B of A, we identify B with the spanning subdigraph $(V, B, \psi \mid_B)$ of D.)

Now, how many a-Eulerian circuits $\mathbf{a}$ lead to a given sparb in this way?

Claim 2: For each sparb B of D to r, there are exactly $\prod_{u \in V} \left(\deg^+ u - 1\right)!$ many a-Eulerian circuits $\mathbf{e}$ such that $\operatorname{Exit} \mathbf{e} = B$.

Once Claim 1 and 2 are both proved, we will obtain a $\prod_{u \in V} \left(\deg^+ u - 1\right)!$-to-1 correspondence between the a-Eulerian circuits and the sparbs of D to r. Thus, it will follow that the number of the former is $\prod_{u \in V} \left(\deg^+ u - 1\right)!$ times the number of the latter. Hence, the theorem will follow. So we need to prove Claim 1 and Claim 2.

Proof of Claim 1. We notice that any vertex of $\operatorname{Exit} \mathbf{e}$ (more precisely, $(V,\ \operatorname{Exit} \mathbf{e},\ \psi\mid_{\operatorname{Exit} \mathbf{e}})$) has outdegree 1, except for r, which has outdegree 0. So it remains to show that r is a to-root of $\operatorname{Exit} \mathbf{e}$.

So let $v \in V$ be any vertex. We must show that $\operatorname{Exit} \mathbf{e}$ has a walk from v to r. In other words, we must show that we can get from v to r by following only the last exit arcs. Either we can keep walking forever, or we eventually run into r. If we run into r, then we are done. So we must show that we won't keep walking forever.

However, this follows by looking at the arcs along which we walk. All of these arcs are last exit arcs of $\mathbf{e}$, and each of them comes earlier in $\mathbf{e}$ than the next (because after $\mathbf{e}$ enters a vertex, it must exit that vertex). So our walk uses arcs that come progressively later and later in $\mathbf{e}$. Hence, it cannot go on forever. Thus, Claim 1 is proved. □

Proof of Claim 2. Let B be a sparb to r. We must prove that there are exactly $\prod_{u \in V} \left(\deg^+ u - 1\right)!$ many a-Eulerian circuits $\mathbf{e}$ such that $\operatorname{Exit} \mathbf{e} = B$.

We shall refer to the arcs in B as the ***B*-arcs**. We thus are looking for a-Eulerian circuits $\mathbf{e}$ that use these B-arcs as a "last resort", i.e., only use them whenever all the other outgoing arcs have already been used.

We construct such a circuit as follows:

A turtle wants to walk through the digraph D using each arc of D at most once. It starts its walk by heading out from r along the arc a. From that point on, it proceeds in the usual way you walk on a digraph: Each time it arrives at a vertex, it chooses an arbitrary arc leading out of this vertex, observing the following two rules:

1. It never reuses an arc that it has already used.
2. It never uses a B-arc unless it has to (i.e., unless this B-arc is the only unused outgoing arc from its current position).

Clearly, the turtle will eventually get stuck at some vertex.

Let $\mathbf{w}$ be the total walk that the turtle has traced by the time it got stuck. Thus, $\mathbf{w}$ is a trail that starts with r and a.

First, we claim that $\mathbf{w}$ is a closed walk (i.e., ends at r).

[*Proof:* Assume the contrary. Thus, **w** ends at a vertex $u \neq r$. Hence, the turtle must have entered u more often that it has exited u. Thus, $\deg^- u = \deg^+ u$ since D is balanced. This is a contradiction, since it means that the turtle still has at least one arc to exit on and thus cannot be stuck.]

So **w** is a circuit. We shall next show that **w** is a Eulerian circuit.

To do so, we need one more notion: A vertex u of D will be called **exhausted** if each outgoing arc from u appears in **w**. So we must show that **all** vertices of D are exhausted.

The vertex r is definitely exhausted (since the turtle gets stuck at r).\

Let us now show that every vertex u is exhausted.

[*Proof:* Assume the contrary. Thus, there exists a vertex u that is not exhausted. Consider this u. Since B is an arb to r, there is a path **p** from u to r that uses only B-arcs. This path **p** starts at the non-exhausted vertex u but ends at the exhausted vertex r.

Let p_i be the first exhausted vertex on this path **p**. Then, $i \neq 0$, so that the preceding vertex p_{i-1} is not exhausted. Since **p** is a path, there is a B-arc from p_{i-1} to p_i.

Since p_i is exhausted, the circuit **w** contains each arc outgoing from p_i. Since $\deg^-(p_i) = \deg^+(p_i)$, this means that **w** also contains each arc incoming into p_i. In particular, **w** contains the B-arc from p_{i-1} to p_i. But this means that **w** contains each arc outgoing from p_{i-1} (since the turtle only uses B-arcs if it has exhausted all other options). Therefore, p_{i-1} is exhausted. Contradiction!]

So every vertex u is exhausted, and thus **w** is a Eulerian circuit. Since **w** starts with a, this means that **w** is an a-Eulerian circuit.

Now, let us analyze the choices that the turtle has made along its way. Every time the turtle is at a vertex $u \in V$, it has to decide which arc it takes next; this arc has to be an unused arc with source u, subject to the conditions that

1. if $u \neq r$, then the B-arc has to be used last;
2. if $u = r$, then a has to be used first.

Let us count how many options the turtle has in total. To clarify this argument, we modify the procedure somewhat: Instead of deciding ad-hoc which arc to take, the turtle should now make all its decisions before embarking on its trip. To do so, it chooses, for each vertex $u \in V$, a total order on the set of all arcs with source u, such that

1. if $u \neq r$, then the B-arc comes last in this order, and
2. if $u = r$, then a comes first in this order.

Note that this total order can be chosen in $\left(\deg^+ u - 1\right)!$ many ways. Thus, in total, there are $\prod\limits_{u \in V} \left(\deg^+ u - 1\right)!$ many ways in which the turtle can choose these orders. Once these orders have been chosen, the turtle proceeds deterministically, using these orders to decide which arc it follows.

So the turtle has $\prod_{u \in V} \left(\deg^+ u - 1\right)!$ many options, and each option leads to a different a-Eulerian circuit $\mathbf{e}$ satisfying $\operatorname{Exit} \mathbf{e} = B$ (why different? because the orders chosen by the turtle are reflected in the orders in which the arcs appear in $\mathbf{e}$). Moreover, any a-Eulerian circuit $\mathbf{e}$ satisfying $\operatorname{Exit} \mathbf{e} = B$ can be obtained from one of these options.

So we get a bijection between the turtle's options and the a-Eulerian circuits $\mathbf{e}$ satisfying $\operatorname{Exit} \mathbf{e} = B$. Thus, the number of the latter circuits equals the number of the former options, which is $\prod_{u \in V} \left(\deg^+ u - 1\right)!$. This proves Claim 2.] □

With Claims 1 and 2 proved, we are almost done. The map

$$\{a\text{-Eulerian circuits of } D\} \to \{\text{sparbs to } r\}, \qquad \mathbf{e} \mapsto \operatorname{Exit} \mathbf{e}$$

is well-defined (by Claim 1). Furthermore, Claim 2 shows that this map is a $\prod_{u \in V} \left(\deg^+ u - 1\right)!$-to-1 correspondence (recall: an m**-to-1 correspondence** means a map $f : X \to Y$ such that each $y \in Y$ has exactly m preimages under f). Hence, the multijection principle says that

$$\begin{aligned} &(\text{the number of } a\text{-Eulerian circuits of } D) \\ &= (\text{the number of sparbs of } r) \cdot \prod_{u \in V} \left(\deg^+ u - 1\right)!. \end{aligned}$$

In other words,

$$\varepsilon(D, a) = \tau(D, r) \cdot \prod_{u \in V} \left(\deg^+ u - 1\right)!.$$

The WORST theorem (and thus also the BEST theorem) is proved. □

Corollary 5.8.7. Let $D = (V, A, \psi)$ be a balanced multidigraph. For each vertex $r \in V$, let $\tau(D, r)$ be the number of spanning arborescences of D rooted to r. Then, $\tau(D, r)$ does not depend on r.

Proof. WLOG assume that $|V| > 1$ and $\deg^+ v > 0$ for all $v \in V$. Thus, we can apply the WORST theorem.

Let r and s be two vertices of D. We want to prove $\tau(D, r) = \tau(D, s)$.

Pick an arc a with source r and an arc b with source s. Then, the WORST theorem yields

$$\varepsilon(D, a) = \tau(D, r) \cdot \prod_{u \in V} \left(\deg^+ u - 1\right)! \qquad \text{and}$$
$$\varepsilon(D, b) = \tau(D, s) \cdot \prod_{u \in V} \left(\deg^+ u - 1\right)!.$$

The LHSs of these equalities are equal (since rotation of Eulerian circuits shows that $\varepsilon(D,a) = \varepsilon(D,b)$). Hence, the RHSs are equal. Cancelling $\prod_{u \in V} \left(\deg^+ u - 1\right)!$ (which is nonzero), we obtain $\tau(D,r) = \tau(D,s)$, qed. □

Next time, we will see how to compute $\tau(D,r)$: the **matrix-tree theorem** (better, **matrix-arborescence theorem**).

5.9. Spanning arborescences vs. spanning trees

As we have learned previously, an arborescence is essentially just a tree, directed "properly". Is there also such a relation between spanning arborescences and spanning trees?

In a way, yes. Let us explain this for the case of a bidirected digraph – i.e., of a digraph G^{bidir} where G is an undirected multigraph.

Proposition 5.9.1. Let $G = (V, E, \varphi)$ be a multigraph. Fix a vertex $r \in V$. Recall that the arcs of G^{bidir} are the pairs $(e,i) \in E \times \{1,2\}$. Identify each spanning tree of G with its edge set, and each spanning arborescence of G^{bidir} with its arc set.

If B is a spanning arborescence of G^{bidir} rooted to r, then we set

$$\overline{B} := \{e \mid (e,i) \in B\}.$$

Then:

1. If B is a spanning arborescence of G^{bidir} rooted to r, then $\overline{B}$ is a spanning tree of G.

2. The map

$$\left\{\text{spanning arborescences of } G^{\text{bidir}} \text{ rooted to } r\right\} \to \{\text{spanning trees of } G\},$$
$$B \mapsto \overline{B}$$

is a bijection.

Proof. Annoying belaboring of trivialities. See Spring 2022 Lecture 18. □

5.10. The matrix-tree theorem

So counting spanning trees in a multigraph is a particular case of counting spanning arborescences (rooted to a given vertex) in a multidigraph. But how do we do either? Let us start with simple examples:

Example 5.10.1. The complete graph K_1 has 1 spanning tree.
The complete graph K_2 has 1 spanning tree.
The complete graph K_3 has 3 spanning trees.
The complete graph K_4 has 16 spanning trees.
The complete graph K_5 has 125 spanning trees.

A bit more data suggests a strange conjecture: K_n seems to have n^{n-2} spanning trees.

Next time we will see why.

Lecture 12

Last time, we conjectured that for any positive integer n, the number of spanning trees of the complete graph K_n is n^{n-2}.

Today, we will prove this. But more importantly, we will prove a formula for the number of spanning trees of any graph, or, even more generally, for the number of spanning arborescences of a digraph.

First some notation:

Definition 5.10.2. We will use the **Iverson bracket notation**: If $\mathcal{A}$ is any logical statement, then

$$[\mathcal{A}] = \begin{cases} 1, & \text{if } \mathcal{A} \text{ is true;} \\ 0, & \text{if } \mathcal{A} \text{ is false.} \end{cases}$$

For example, $[K_2 \text{ is a tree}] = 1$ but $[K_3 \text{ is a tree}] = 0$.

Definition 5.10.3. Let M be a matrix. Let i and j be two integers. Then,

$M_{i,j}$ will mean the entry of M in row i and column j;

$M_{\sim i, \sim j}$ will mean the matrix M with row i and column j removed.

For example,

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}_{2,3} = f \qquad \text{and} \qquad \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}_{\sim 2, \sim 3} = \begin{pmatrix} a & b \\ g & h \end{pmatrix}.$$

Now, we shall assign a matrix to any multidigraph:

Definition 5.10.4. Let $D = (V, A, \psi)$ be a multidigraph. Assume that $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$.

For any $i, j \in V$, we let $a_{i,j}$ be the # of arcs of D that have source i and target j. (As usual, # means "number".)

The **Laplacian** of D is defined to be the $n \times n$-matrix $L \in \mathbb{Z}^{n \times n}$ whose entries are given by

$$L_{i,j} = \underbrace{\left(\deg^+ i\right)}_{\text{outdegree of } i} \cdot \underbrace{[i = j]}_{\substack{\text{This is also} \\ \text{known as } \delta_{i,j}}} - a_{i,j} \qquad \text{for all } i, j \in V.$$

In other words, it is the matrix

$$L = \begin{pmatrix} \deg^+ 1 - a_{1,1} & -a_{1,2} & \cdots & -a_{1,n} \\ -a_{2,1} & \deg^+ 2 - a_{2,2} & \cdots & -a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n,1} & -a_{n,2} & \cdots & \deg^+ n - a_{n,n} \end{pmatrix}.$$

Note that loops do not matter for L.

Proposition 5.10.5. Let $D = (V, A, \psi)$ be a multidigraph. Assume that $V = \{1, 2, \ldots, n\}$ for some positive integer n. Then, the Laplacian L of D is singular, i.e., it satisfies $\det L = 0$.

Proof. The sum of the columns (as vectors) is the zero vector, because for each $i \in V$ we have

$$\sum_{j=1}^{n} \left(\left(\deg^+ i \right) \cdot [i = j] - a_{i,j} \right) = \underbrace{\sum_{j=1}^{n} \left(\deg^+ i \right) \cdot [i = j]}_{= \deg^+ i} - \underbrace{\sum_{j=1}^{n} a_{i,j}}_{= \deg^+ i}$$
$$= \deg^+ i - \deg^+ i = 0.$$

This means that the vector $(1, 1, \ldots, 1)^T$ lies in the nullspace of L. Thus, L has a nontrivial nullspace, i.e., is singular. □

Much more interesting is the following:

Theorem 5.10.6 (Matrix-Tree Theorem)**.** Let $D = (V, A, \psi)$ be a multidigraph. Assume that $V = \{1, 2, \ldots, n\}$ for some positive integer n.

Let L be the Laplacian of D. Let r be a vertex of D. Then,

$$(\text{\# of spanning arborescences of } D \text{ rooted to } r) = \det \left(L_{\sim r, \sim r} \right).$$

Before we prove this, some remarks:

- The determinant $\det \left(L_{\sim r, \sim r} \right)$ is the (r, r)-th entry of the adjugate matrix of L.
- The $V = \{1, 2, \ldots, n\}$ assumption is a typical WLOG assumption. You can always ensure that it holds by renaming the vertices of D. You can even avoid it altogether if you are fine with $V \times V$-matrices.

Now, let us use the Matrix-Tree Theorem (short **MTT**) to count the spanning trees of K_n.

We fix a positive integer n. Let L be the Laplacian of the multidigraph K_n^{bidir} (where K_n, as we recall, is the complete graph on $\{1, 2, \ldots, n\}$). Then, each vertex of K_n^{bidir} has outdegree $n - 1$, and thus we have

$$L = \begin{pmatrix} n-1 & -1 & \cdots & -1 \\ -1 & n-1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & n-1 \end{pmatrix}_{n \times n}$$

(all diagonal entries are $n-1$, while all other entries are -1). By a proposition from last time, there is a bijection between

$$\begin{aligned} &\left\{\text{spanning arborescenes of } K_n^{\text{bidir}} \text{ rooted to } 1\right\} \\ &\text{and } \left\{\text{spanning trees of } K_n\right\}. \end{aligned}$$

Hence, by the bijection principle,

$$\begin{aligned} &(\text{\# of spanning trees of } K_n) \\ &= \left(\text{\# of spanning arborescenes of } K_n^{\text{bidir}} \text{ rooted to } 1\right) \\ &= \det\left(L_{\sim 1,\sim 1}\right) \qquad \text{(by the MTT)} \\ &= \det \begin{pmatrix} n-1 & -1 & \cdots & -1 \\ -1 & n-1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & n-1 \end{pmatrix}_{(n-1)\times(n-1)}. \end{aligned}$$

How do we compute this determinant?

$$\det\begin{pmatrix} n-1 & -1 & \cdots & -1 \\ -1 & n-1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & n-1 \end{pmatrix}_{(n-1)\times(n-1)}$$

$$= \det\begin{pmatrix} n-1 & -1 & -1 & -1 & \cdots & -1 \\ -1 & n-1 & -1 & -1 & \cdots & -1 \\ -1 & -1 & n-1 & -1 & \cdots & -1 \\ -1 & -1 & -1 & n-1 & \cdots & -1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & -1 & \cdots & n-1 \end{pmatrix}_{(n-1)\times(n-1)}$$

$$= \det\begin{pmatrix} n-1 & -1 & -1 & -1 & \cdots & -1 \\ -n & n & 0 & 0 & \cdots & 0 \\ -n & 0 & n & 0 & \cdots & 0 \\ -n & 0 & 0 & n & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -n & 0 & 0 & 0 & \cdots & n \end{pmatrix}_{(n-1)\times(n-1)}$$

$$= \det\begin{pmatrix} n & 0 & 0 & \cdots & 0 \\ 0 & n & 0 & \cdots & 0 \\ 0 & 0 & n & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & n \end{pmatrix}$$

$$\cdot \left((n-1) - \begin{pmatrix} -1 & -1 & -1 & \cdots & -1 \end{pmatrix} \begin{pmatrix} n & 0 & 0 & \cdots & 0 \\ 0 & n & 0 & \cdots & 0 \\ 0 & 0 & n & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & n \end{pmatrix}^{-1} \begin{pmatrix} -n \\ -n \\ -n \\ \vdots \\ -n \end{pmatrix} \right)$$

$$\left(\begin{array}{c}\text{by the formula for the determinant} \\ \text{in terms of the Schur complement}\end{array}\right)$$

$$= n^{n-2} \cdot \left((n-1) - \frac{n\,(n-2)}{n} \right) = n^{n-2}.$$

See the notes (Lecture 18 Spring 2022) for three other ways to compute it.
So, if we can prove the MTT, we obtain:

Theorem 5.10.7 (Cayley's formula)**.** Let n be a positive integer. Then, the # of spanning trees of K_n is n^{n-2}.

In other words, the # of simple graphs with vertex set $\{1, 2, \ldots, n\}$ that are trees is n^{n-2}.

There are several beautiful combinatorial proofs of this formula as well. I can particularly recommend the one of Joyal (e.g., Leinster's AMM paper from 2019) and also the Prüfer code proof (in most textbooks). See the 2022 notes for references.

We will however complete our algebraic proof by proving the MTT.

First, a bunch of lemmas.

Lemma 5.10.8. Let $D = (V, A, \psi)$ be a multidigraph. Let r be a vertex of D. Assume that D has no cycles and no arcs with source r. Assume furthermore that each vertex $v \in V \setminus \{r\}$ has outdegree 1. Then, D is an arborescence rooted to r.

Proof. We need to show that r is a to-root. Once this is shown, the dual arborescence equivalence theorem will give us the rest (since the outdegrees are $\deg^+ r = 0$ and $\deg^+ v = 1$).

In other words, we need to show that there is a walk from u to r for each $u \in V$.

Let $u \in V$. Start at u and keep walking. There is always an arc to follow until you arrive at r. This walk cannot go on forever without revisiting a vertex. But if it does revisit a vertex, you get a cycle, which is not allowed. So you will arrive at r eventually, qed. □

Now, here is our strategy for proving the MTT:

1. First, we will prove it in the case when each vertex $v \in V \setminus \{r\}$ has outdegree 1. In this case, removing all the arcs with source r, we have essentially two options (subcases): either D is itself an arborescence or D has a cycle.

2. Then, we will prove the MTT in the slightly more general case when each $v \in V \setminus \{r\}$ has outdegree ≤ 1. This is easy, since a vertex $v \in V \setminus \{r\}$ of outdegree 0 trivializes the theorem.

3. Finally, we will prove the MTT in the general case. This is done by strong induction on the number of arcs of D. Every time you have a vertex $v \in V \setminus \{r\}$ with outdegree > 1, you can pick such a vertex and color the outgoing arcs from it red and blue such that each color is used at least once. Then, you can consider the subdigraph D^{red} of D which has only the red arcs (and the uncolored ones), and the subdigraph D^{blue} of D which has only the blue arcs (and the uncolored ones). By the induction hypothesis, the MTT holds for D^{red} and for D^{blue}. Adding these equalities together, you get it for D as well.

Step 1 is the hard one. We first study a very special case:

Lemma 5.10.9. Let $D = (V, A, \psi)$ be a multidigraph. Let r be a vertex of D. Assume that D has no cycles and has no arcs with source r. Assume that each vertex $v \in V \setminus \{r\}$ has outdegree 1. Then:

(a) The digraph D has a unique spanning arborescence rooted to r.

(b) Assume that $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$. Let L be the Laplacian of D. Then, $\det(L_{\sim r, \sim r}) = 1$.

Proof. **(a)** The previous lemma tells us that D itself is an arborescence rooted to r. Hence, it is a sparb (= spanning arborescence rooted) to r. If you remove an arc, you lose this, so it is the only sparb.

(b) This can be derived from the existence of a toposort for D (see Exercise 5 **(a)** on homework set #6). Permuting the rows and the columns of $L_{\sim r, \sim r}$ to ensure that they appear in the same order in this matrix as they appear in this toposort, we can make this matrix upper-triangular, and therefore its determinant is $1 \cdot 1 \cdot \cdots \cdot 1 = 1$ (since the diagonal entries are 1).

Another proof: We recall how determinants are defined.

Let $r = n$ WLOG. (Otherwise, permute the vertices.)

Let S_{n-1} denote the group of permutations of the set $\{1, 2, \ldots, n-1\} = V \setminus \{r\}$.

Then, the Leibniz formula (= the definition) for the determinant yields

$$\det(L_{\sim r, \sim r}) = \det(L_{\sim n, \sim n}) = \sum_{\sigma \in S_{n-1}} \operatorname{sign} \sigma \cdot \prod_{i=1}^{n-1} L_{i, \sigma(i)}.$$

Now we claim that the product $\prod_{i=1}^{n-1} L_{i,\sigma(i)}$ is 0 for any $\sigma \in S_{n-1}$ that is distinct from id.

To do so, we assume the contrary. Thus, there exists a permutation $\sigma \neq \mathrm{id}$ such that $\prod_{i=1}^{n-1} L_{i,\sigma(i)} \neq 0$. Consider this σ.

Thus, for each $i \in \{1, 2, \ldots, n-1\}$, we have $L_{i,\sigma(i)} \neq 0$, which means that $\sigma(i)$ either equals i or has an incoming arc from i.

However, not being the identity, the permutation σ must have a nontrivial cycle

$$\left(j,\ \sigma(j),\ \sigma^2(j),\ \sigma^3(j),\ \ldots,\ \sigma^{k-1}(j),\ \sigma^k(j) = j\right) \text{ with } k > 1.$$

Since it is nontrivial, its consecutive entries are never equal. Thus, the digraph D has an arc from j to $\sigma(j)$, an arc from $\sigma(j)$ to $\sigma^2(j)$, an arc from $\sigma^2(j)$ to $\sigma^3(j)$, and so on. But this means that D has a cycle, which contradicts our assumption.

So we have shown that $\prod_{i=1}^{n-1} L_{i,\sigma(i)} = 0$ for all $\sigma \neq \mathrm{id}$.

Hence,

$$\begin{aligned}
\det\left(L_{\sim r,\sim r}\right) &= \sum_{\sigma \in S_{n-1}} \operatorname{sign} \sigma \cdot \prod_{i=1}^{n-1} L_{i,\sigma(i)} \\
&= \underbrace{\operatorname{sign} \mathrm{id}}_{=1} \cdot \prod_{i=1}^{n-1} \underbrace{L_{i,\mathrm{id}(i)}}_{\substack{=L_{i,i} \\ =\deg^+ i - a_{i,i} \\ =\deg^+ i \\ =1 \text{ (by assumption)}}} \\
&= 1.
\end{aligned}$$

□

Next, we drop the "no cycles" condition:

Lemma 5.10.10. Let $D = (V, A, \psi)$ be a multidigraph. Let r be a vertex of D. Assume that each vertex $v \in V \setminus \{r\}$ has outdegree 1. Then, the MTT holds for D and r.

Proof. First of all, we WLOG assume that D has no arcs with source r, since such arcs can be removed without modifying anything in the MTT.

Next, we WLOG assume that $r = n$.

Now, we are in one of the following two cases:

Case 1: The digraph D has a cycle.

Case 2: The digraph D has no cycles.

Consider Case 1. Here, D has a cycle $\mathbf{v} = (v_1, *, v_2, *, \ldots, *, v_m)$. This cycle $\mathbf{v}$ cannot contain r (since D has no arcs with source r). We WLOG assume that $\mathbf{v} = (1, *, 2, *, 3, *, \ldots, *, m-1, *, 1)$ (by renaming the vertices).

The first $m-1$ rows of L thus look as follows:

$$\begin{matrix}
1 & -1 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 1 & -1 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 1 & -1 & 0 & 0 & \cdots & 0 \\
-1 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 0
\end{matrix}.$$

The sum of these rows (as vectors) is the zero vector. Hence, $\det\left(L_{\sim r,\sim r}\right) = 0$ (since these rows are also rows of $L_{\sim r,\sim r}$, at least if you remove the r-th column).

Now, to prove that the MTT holds, we have to show that the # of sparbs to r is also 0. But this is clear, since r is not accessible from any of the vertices $1, 2, \ldots, m-1$.

So the MTT boils down to $0 = 0$ in this case.

Now let us consider Case 2. In this case, D has no cycles. Hence, the previous lemma tells us that the MTT boils down to $1 = 1$.

In either case, we have proved the MTT for our D and r. □

For Step 2, we extend our level of generality a bit higher:

Lemma 5.10.11. Let $D = (V, A, \psi)$ be a multidigraph. Let r be a vertex of D. Assume that each vertex $v \in V \setminus \{r\}$ has outdegree ≤ 1. Then, the MTT holds for D and r.

Proof. If each vertex $v \in V \setminus \{r\}$ has outdegree 1, then this follows from the previous lemma.

So assume not. Then, some vertex $v \in V \setminus \{r\}$ has outdegree 0. Therefore, the (essentially) v-th row of $L_{\sim r, \sim r}$ is 0. So $\det(L_{\sim r, \sim r}) = 0$.

Furthermore, D has no sparbs to r (since you cannot get from v to r).

So the MTT says that $0 = 0$, which we agree with.

□

Finally, Step 3:

Proof of the MTT in the general case. We introduce a notation:

> Let M and N be two $n \times n$-matrices that agree in all but one row. That is, there exists some $j \in \{1, 2, \ldots, n\}$ such that for each $i \neq j$, we have
> $$(\text{the } i\text{-th row of } M) = (\text{the } i\text{-th row of } N).$$
> Then, we write $M \overset{j}{\equiv} N$. Furthermore, we let $M \overset{j}{+} N$ be the $n \times n$-matrix that is obtained from M by adding the j-th row of N to the j-th row of M (while leaving all remaining rows unchanged).

For instance, if $M = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$ and $N = \begin{pmatrix} a & b & c \\ d' & e' & f' \\ g & h & i \end{pmatrix}$, then $M \overset{2}{\equiv} N$ and

$$M \overset{2}{+} N = \begin{pmatrix} a & b & c \\ d + d' & e + e' & f + f' \\ g & h & i \end{pmatrix}.$$

A well-known property of determinants (the **multilinearity of the determinant**) says that if M and N are two $n \times n$-matrices that satisfy $M \overset{j}{\equiv} N$ for some j, then

$$\det\left(M \overset{j}{+} N\right) = \det M + \det N.$$

Now, let us prove the MTT. We proceed by strong induction on the # of arcs of D.

Induction step: Let $m \in \mathbb{N}$. Assume (as the induction hypothesis, short IH) that the MTT holds for all digraphs D that have $< m$ arcs. We must now prove it for our digraph D with m arcs.

WLOG assume that $r = n$.

If each vertex $v \in V \setminus \{r\}$ has outdegree ≤ 1, then the MTT holds by the previous lemma.

Thus, we WLOG assume that some $v \in V \setminus \{r\}$ has outdegree > 1. Fix such a v, and color each arc with source v either red or blue, making sure that at least one arc is red and at least one arc is blue. All arcs whose source is not v remain uncolored.

Let D^{red} be the digraph obtained from D by removing the blue arcs. By the IH, the MTT holds for D^{red}, so we have

$$\left(\text{\# of sparbs of } D^{\text{red}} \text{ to } r\right) = \det\left(L^{\text{red}}_{\sim r, \sim r}\right),$$

where L^{red} is the Laplacian of D^{red}.

Let D^{blue} be the digraph obtained from D by removing the redcs. By the IH, the MTT holds for D^{blue}, so we have

$$\left(\text{\# of sparbs of } D^{\text{blue}} \text{ to } r\right) = \det\left(L^{\text{blue}}_{\sim r, \sim r}\right),$$

where L^{blue} is the Laplacian of D^{blue}.

Now, I claim that $L^{\text{red}}_{\sim r, \sim r} \overset{v}{\equiv} L^{\text{blue}}_{\sim r, \sim r}$ (indeed, $L^{\text{red}} \overset{v}{\equiv} L^{\text{blue}}$) and

$$L^{\text{red}}_{\sim r, \sim r} \overset{v}{+} L^{\text{blue}}_{\sim r, \sim r} = L_{\sim r, \sim r}$$

(indeed, $L^{\text{red}} \overset{v}{+} L^{\text{blue}} = L$). Hence, the multilinearity of the determinant yields

$$\det\left(L^{\text{red}}_{\sim r, \sim r} \overset{v}{+} L^{\text{blue}}_{\sim r, \sim r}\right) = \det\left(L^{\text{red}}_{\sim r, \sim r}\right) + \det\left(L^{\text{blue}}_{\sim r, \sim r}\right).$$

Thus,

$$\begin{aligned}
&\det\left(L_{\sim r, \sim r}\right) \\
&= \det\left(L^{\text{red}}_{\sim r, \sim r} \overset{v}{+} L^{\text{blue}}_{\sim r, \sim r}\right) \\
&= \det\left(L^{\text{red}}_{\sim r, \sim r}\right) + \det\left(L^{\text{blue}}_{\sim r, \sim r}\right) \\
&= \left(\text{\# of sparbs of } D^{\text{red}} \text{ to } r\right) + \left(\text{\# of sparbs of } D^{\text{blue}} \text{ to } r\right) \\
&= \left(\text{\# of sparbs of } D \text{ to } r\right).
\end{aligned}$$

This completes the induction step. Thus, the MTT is proved.

(See Spring 2022 Lecture 19 for details.) □

Next time, we will combine the MTT and the BEST theorem. In particular:

Proposition 5.10.12. Let n be a positive integer. Pick any arc a of the multidigraph K_n^{bidir}. Then, the # of Eulerian circuits of K_n^{bidir} that start with a is $n^{n-2} \cdot (n-2)!^n$.

Proof. Think about it. □

Lecture 13

Last time, we showed:

Definition 5.10.13. Let $D = (V, A, \psi)$ be a multidigraph. Assume that $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$.

For any $i, j \in V$, we let $a_{i,j}$ be the # of arcs of D that have source i and target j.

The **Laplacian** of D is defined to be the $n \times n$-matrix $L \in \mathbb{Z}^{n\times n}$ whose entries are given by

$$L_{i,j} = \underbrace{(\deg^+ i)}_{\text{outdegree of } i} \cdot \underbrace{[i = j]}_{\substack{\text{This is also} \\ \text{known as } \delta_{i,j}}} - a_{i,j} \qquad \text{for all } i, j \in V.$$

In other words, it is the matrix

$$L = \begin{pmatrix} \deg^+ 1 - a_{1,1} & -a_{1,2} & \cdots & -a_{1,n} \\ -a_{2,1} & \deg^+ 2 - a_{2,2} & \cdots & -a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n,1} & -a_{n,2} & \cdots & \deg^+ n - a_{n,n} \end{pmatrix}.$$

Theorem 5.10.14 (Matrix-Tree Theorem)**.** Let $D = (V, A, \psi)$ be a multidigraph. Assume that $V = \{1, 2, \ldots, n\}$ for some positive integer n.

Let L be the Laplacian of D. Let r be a vertex of D. Then,

$$(\text{\# of spanning arborescences of } D \text{ rooted to } r) = \det\left(L_{\sim r, \sim r}\right).$$

Meanwhile, $\det L = 0$ (if D has at least 1 vertex).

We applied this to $D = K_n^{\text{bidir}}$, and this gave us

$$\begin{aligned} &(\text{\# of spanning trees of } K_n) \\ &= (\text{\# of spanning arborescences of } K_n \text{ rooted to } 1) \\ &= n^{n-2} \end{aligned}$$

(Cayley's theorem).

I stated but did not the prove:

Proposition 5.10.15. Let n be a positive integer. Pick any arc a of the multidigraph K_n^{bidir}. Then, the # of Eulerian circuits of K_n^{bidir} whose first arc is a is $n^{n-2} \cdot (n-2)!^n$.

Proof. Let r be the source of a. The WORST theorem then yields

$$\left(\text{\# of Eulerian circuits of } K_n^{\text{bidir}} \text{ whose first arc is } a\right)$$

$$= \underbrace{\left(\text{\# of spanning arborescences of } K_n^{\text{bidir}} \text{ rooted to } r\right)}_{\substack{=(\text{\# of spanning trees of } K_n) \\ = n^{n-2}}} \cdot \prod_{u=1}^{n} \left(\underbrace{\deg^+ u}_{=n-1} - 1 \right)!$$

$$= n^{n-2} \cdot \prod_{u=1}^{n} (n-2)! = n^{n-2} \cdot (n-2)!^n.$$

□

In comparison, very little is known about the # of Eulerian circuits in the undirected K_n. See a homework exercise for one thing you can say about this number. (A135388 in the OEIS)

Remark 5.10.16. Cayley's theorem tells us how many n-vertex trees there are where the vertices are $1, 2, \ldots, n$. These are called **labelled trees**.

In contrast, **unlabelled trees** are equivalence classes of labelled trees under isomorphism.

How many unlabelled trees are there with n vertices?

The closest we can get to an answer to this question is an approximate formula (Otter 1948):

$$\approx \beta \frac{\alpha^n}{n^{5/2}} \qquad \text{with } \alpha \approx 2.955 \text{ and } \beta \approx 0.5349.$$

5.11. The undirected MTT

The Matrix-Tree Theorem (MTT) can be applied to G^{bidir} for any undirected graph G:

Theorem 5.11.1 (undirected MTT)**.** Let $G = (V, E, \varphi)$ be a multigraph. Assume that $V = \{1, 2, \ldots, n\}$ for some positive integer n.

Let L be the Laplacian of the digraph G^{bidir}. Explicitly, this is the $n \times n$-matrix $L \in \mathbb{Z}^{n \times n}$ whose entries are given by

$$L_{i,j} = (\deg i) \cdot [i = j] - a_{i,j},$$

where $a_{i,j}$ is the # of edges of G that have endpoints i and j (with loops counting twice). Then:

(a) For any vertex r of G, we have

$$(\text{\# of spanning trees of } G) = \det\left(L_{\sim r, \sim r}\right).$$

(b) Let t be an indeterminate. Expand the determinant $\det(tI_n + L)$ (where I_n is the $n \times n$ identity matrix) as a polynomial in t:

$$\det(tI_n + L) = c_n t^n + c_{n-1} t^{n-1} + \cdots + c_1 t^1 + c_0 t^0,$$

where $c_0, c_1, \ldots, c_n$ are numbers. (Note that this is the characteristic polynomial of L up to substituting $-t$ for t and multiplying by a number of -1. Some of its coefficients are $c_n = 1$ and $c_{n-1} = \operatorname{Tr} L$ and $c_0 = \det L$.) Then,

$$(\text{\# of spanning trees of } G) = \frac{1}{n} c_1.$$

(c) Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of L, listed in such a way that $\lambda_n = 0$ (we know that 0 is an eigenvalue of L, since L is singular). Then,

$$(\text{\# of spanning trees of } G) = \frac{1}{n} \lambda_1 \lambda_2 \cdots \lambda_{n-1}.$$

Proof. **(a)** Let r be a vertex of G. As we saw a lecture or two ago, there is a bijection

$$\left\{\text{spanning arborescences of } G^{\text{bidir}} \text{ rooted to } r\right\} \\ \to \{\text{spanning trees of } G\}.$$

Thus,

$$\begin{aligned} &(\text{\# of spanning trees of } G) \\ &= \left(\text{\# of spanning arborescences of } G^{\text{bidir}} \text{ rooted to } r\right) \\ &= \det(L_{\sim r, \sim r}) \qquad (\text{by the MTT}). \end{aligned}$$

Thus part **(a)**.

(b) We claim that

$$c_1 = \sum_{r=1}^{n} \det(L_{\sim r, \sim r}).$$

Note that this is a general linear-algebraic result that holds for any matrix L, not just for a Laplacian of a digraph.

Here is an outline of the proof, on the example where $n = 3$: Let $L =$

$\begin{pmatrix} a & b & c \\ a' & b' & c' \\ a'' & b'' & c'' \end{pmatrix}$. Then,

$$\begin{aligned} \det(tI_n + L) &= \det \begin{pmatrix} t+a & b & c \\ a' & t+b' & c' \\ a'' & b'' & t+c'' \end{pmatrix} \\ &= (t+a)\left(t+b'\right)\left(t+c''\right) + bc'a'' + ca'b'' \\ &\qquad - (t+a)\,c'b'' - c\left(t+b'\right)a'' - ba'\left(t+c''\right). \end{aligned}$$

The t-coefficient of this will be

$$\begin{aligned} &\left(b'c'' + ab' + ac''\right) + 0 + 0 - c'b'' - ca'' - ba' \\ &= \left(b'c'' - c'b''\right) + \left(ab' - ba'\right) + \left(ac'' - ca''\right) \\ &= \det \begin{pmatrix} b' & c' \\ b'' & c'' \end{pmatrix} + \det \begin{pmatrix} a & c \\ a'' & c'' \end{pmatrix} + \det \begin{pmatrix} a & b \\ a' & b' \end{pmatrix} \\ &= \sum_{r=1}^{n} \det\left(L_{\sim r, \sim r}\right). \end{aligned}$$

I showed a handwavy proof on the screen. More rigorous proofs can be found in references that are in the notes (2022 Lecture 19). A slicker proof can be given using exterior powers.

Either way, it follows that

$$\begin{aligned} c_1 &= \sum_{r=1}^{n} \underbrace{\det\left(L_{\sim r, \sim r}\right)}_{=(\#\text{ of spanning trees of } G)} \\ &= \sum_{r=1}^{n} (\#\text{ of spanning trees of } G) \\ &= n \cdot (\#\text{ of spanning trees of } G). \end{aligned}$$

Hence,

$$(\#\text{ of spanning trees of } G) = \frac{1}{n} c_1.$$

Part **(b)** follows.

(c) This will follow from part **(b)** if we can show that

$$c_1 = \lambda_1 \lambda_2 \cdots \lambda_{n-1}.$$

The roots of the characteristic polynomial of a matrix are its eigenvalues. That is,

$$\det(tI_n - L) = (t - \lambda_1)(t - \lambda_2) \cdots (t - \lambda_n).$$

Substituting $-t$ for t here, we obtain

$$\det(-tI_n - L) = (-t-\lambda_1)(-t-\lambda_2)\cdots(-t-\lambda_n).$$

Multiplying by $(-1)^n$, this simplifies to

$$\begin{aligned}\det(tI_n + L) &= (t+\lambda_1)(t+\lambda_2)\cdots(t+\lambda_n) \\ &= (t+\lambda_1)(t+\lambda_2)\cdots(t+\lambda_{n-1})\,t \qquad (\text{since } \lambda_n = 0).\end{aligned}$$

Hence, the t-coefficient of $\det(tI_n + L)$ is $\lambda_1\lambda_2\cdots\lambda_{n-1}$ (since this is the t-coefficient of the RHS). In other words,

$$c_1 = \lambda_1\lambda_2\cdots\lambda_{n-1},$$

qed. □

Laplacians of digraphs often have computable eigenvalues, so the restatements in parts **(b)** and **(c)** can actually come quite handy. A striking example is the n-hypercube graph Q_n, whose # of spanning trees you will compute on the HW.

Here is a simpler example, in which part **(a)** suffices:

Exercise 2. Let n and m be two positive integers. Let $K_{n,m}$ be the simple graph with $n+m$ vertices

$$1, 2, \ldots, n \qquad \text{and} \qquad -1, -2, \ldots, -m,$$

where two vertices i and j are adjacent if and only if they have opposite signs (i.e., each positive vertex is adjacent to each negative vertex, but there are no other adjacencies).

How many spanning trees does $K_{n,m}$ have?

Solution. Rename the vertices $-1, -2, \ldots, -m$ as $n+1, n+2, \ldots, n+m$. Then, the Laplacian L of the digraph $K_{n,m}^{\text{bidir}}$ can be written in block-matrix notation as follows:

$$L = \begin{pmatrix} A & B \\ C & D \end{pmatrix},$$

where

- A is a diagonal $n \times n$-matrix whose all diagonal entries are m.
- B is an $n \times m$-matrix whose all entries are -1.
- C is an $m \times n$-matrix whose all entries are -1.
- D is a diagonal $m \times m$-matrix whose all diagonal entries are n.

For example, if $n = 3$ and $m = 2$, then

$$L = \begin{pmatrix} 2 & 0 & 0 & -1 & -1 \\ 0 & 2 & 0 & -1 & -1 \\ 0 & 0 & 2 & -1 & -1 \\ -1 & -1 & -1 & 3 & 0 \\ -1 & -1 & -1 & 0 & 3 \end{pmatrix}.$$

Part **(a)** of the undirected MTT yields

$$(\text{\# of spanning trees of } K_{n,m}) = \det\left(L_{\sim r,\sim r}\right) \quad \text{for any vertex } r.$$

Let us apply this to $r = 1$. Thus we get

$$(\text{\# of spanning trees of } K_{n,m}) = \det\left(L_{\sim 1,\sim 1}\right).$$

Note that

$$L_{\sim 1,\sim 1} = \begin{pmatrix} \widetilde{A} & \widetilde{B} \\ \widetilde{C} & D \end{pmatrix},$$

where

- $\widetilde{A}$ is a diagonal $(n-1) \times (n-1)$-matrix whose all diagonal entries are m.
- $\widetilde{B}$ is an $(n-1) \times m$-matrix whose all entries are -1.
- $\widetilde{C}$ is an $m \times (n-1)$-matrix whose all entries are -1.
- D is a diagonal $m \times m$-matrix whose all diagonal entries are n.

We can thus compute $\det\left(L_{\sim 1,\sim 1}\right)$ using the Schur complement:

$$\det\left(L_{\sim 1,\sim 1}\right) = \det\begin{pmatrix} \widetilde{A} & \widetilde{B} \\ \widetilde{C} & D \end{pmatrix} = \det \widetilde{A} \cdot \det\left(D - \widetilde{C}\widetilde{A}^{-1}\widetilde{B}\right)$$
$$= m^{n-1} \cdot \det\begin{pmatrix} \text{the } m \times m\text{-matrix whose all} \\ \text{diagonal entries are } n - m^{-1}(n-1) \\ \text{and whose off-diagonal entries} \\ \text{are } -m^{-1}(n-1) \end{pmatrix}.$$

How do we compute the determinant on the LHS?

Proposition 5.11.2. Let $n \in \mathbb{N}$. Let x and a be two numbers. Then,

$$\det \underbrace{\begin{pmatrix} x & a & a & \cdots & a & a \\ a & x & a & \cdots & a & a \\ a & a & x & \cdots & a & a \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a & a & a & \cdots & x & a \\ a & a & a & \cdots & a & x \end{pmatrix}_{n\times n}}_{\substack{\text{the } n\times n\text{-matrix whose} \\ \text{diagonal entries are } x \text{ and} \\ \text{whose off-diagonal entries are } a}} = (x + (n-1)a)(x-a)^{n-1}.$$

Proof. The simplest approach is to subtract row 1 from all other rows, then factor out $x - a$ from all other rows, then add them back to row 1 with the appropriate multiple, etc. $\square$

Using this proposition, we can finish our calculation of $\det(L_{\sim 1,\sim 1})$, and obtain $m^{n-1} \cdot n^{m-1}$. Thus:

Theorem 5.11.3. Let n and m be two positive integers. Let $K_{n,m}$ be the simple graph with $n + m$ vertices

$$1, 2, \ldots, n \qquad \text{and} \qquad -1, -2, \ldots, -m,$$

where two vertices i and j are adjacent if and only if they have opposite signs (i.e., each positive vertex is adjacent to each negative vertex, but there are no other adjacencies). Then,

$$(\text{\# of spanning trees of } K_{n,m}) = m^{n-1} \cdot n^{m-1}.$$

This can also be proved combinatorially (Abu-Sbeih 1990).

On a sidenote, the above proposition can be generalized in two ways:

Proposition 5.11.4. Let $n \in \mathbb{N}$. Let $a_1, a_2, \ldots, a_n$ be n numbers, and let x be a further number. Then,

$$\det \begin{pmatrix} x & a_1 & a_2 & \cdots & a_{n-1} & a_n \\ a_1 & x & a_2 & \cdots & a_{n-1} & a_n \\ a_1 & a_2 & x & \cdots & a_{n-1} & a_n \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_1 & a_2 & a_3 & \cdots & x & a_n \\ a_1 & a_2 & a_3 & \cdots & a_n & x \end{pmatrix}_{(n+1)\times(n+1)} = \left(x + \sum_{i=1}^{n} a_i \right) \cdot \prod_{i=1}^{n} (x - a_i).$$

Proposition 5.11.5. Let $n \in \mathbb{N}$. Let $x_1, x_2, \ldots, x_n$ be n numbers, and let a be a further number. Then,

$$\det \begin{pmatrix} x_1 & a & a & \cdots & a & a \\ a & x_2 & a & \cdots & a & a \\ a & a & x_3 & \cdots & a & a \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a & a & a & \cdots & x_{n-1} & a \\ a & a & a & \cdots & a & x_n \end{pmatrix}_{n\times n} = \prod_{i=1}^{n} (x_i - a) + a \sum_{i=1}^{n} b_i,$$

where we set

$$b_i := \prod_{\substack{k \in \{1,2,\ldots,n\};\\ k \neq i}} (x_k - a) \qquad \text{for each } i \in \{1, 2, \ldots, n\}.$$

Can you merge these two generalizations, allowing both $x_1, x_2, \ldots, x_{n+1}$ on the diagonal and $a_1, a_2, \ldots, a_n$ outside of it? I don't know, and I would like to!

5.12. de Bruijn sequences

Here is a puzzle: What is special about the periodic sequence

$$\| : \ 0000\ 1111\ 0110\ 0101\ : \| \qquad ?$$

(This is an infinite sequence of 0's and 1's. The spaces between some of them are purely for readability. The || : and : || symbols are "repeat signs", signifying that everything between them is to be repeated over and over. For example, || : *abc* : || means *abcabcabc* $\cdots$. So this sequence is

$$0000\ 1111\ 0110\ 0101\ 0000\ 1111\ 0110\ 0101\ 0000\ 1111\ 0110\ 0101\ \cdots$$

)

Let us slide a length-4 window along this sequence:

$\boxed{0000}$ 1111 0110 0101 0000 1111 0110 0101 $\cdots$
0$\boxed{000\ 1}$111 0110 0101 0000 1111 0110 0101 $\cdots$
00$\boxed{00\ 11}$11 0110 0101 0000 1111 0110 0101 $\cdots$
000$\boxed{0\ 111}$1 0110 0101 0000 1111 0110 0101 $\cdots$
0000 $\boxed{1111}$ 0110 0101 0000 1111 0110 0101 $\cdots$
0000 1$\boxed{111\ 0}$110 0101 0000 1111 0110 0101 $\cdots$
0000 11$\boxed{11\ 01}$10 0101 0000 1111 0110 0101 $\cdots$
0000 111$\boxed{1\ 011}$0 0101 0000 1111 0110 0101 $\cdots$
0000 1111 $\boxed{0110}$ 0101 0000 1111 0110 0101 $\cdots$
0000 1111 0$\boxed{110\ 0}$101 0000 1111 0110 0101 $\cdots$
0000 1111 01$\boxed{10\ 01}$01 0000 1111 0110 0101 $\cdots$
0000 1111 011$\boxed{0\ 010}$1 0000 1111 0110 0101 $\cdots$
0000 1111 0110 $\boxed{0101}$ 0000 1111 0110 0101 $\cdots$
0000 1111 0110 0$\boxed{101\ 0}$000 1111 0110 0101 $\cdots$
0000 1111 0110 01$\boxed{01\ 00}$00 1111 0110 0101 $\cdots$
0000 1111 0110 010$\boxed{1\ 000}$0 1111 0110 0101 $\cdots$

We notice that every possible length-4 bitstring can be obtained by placing the window somewhere on our string. In other words, as we slide the window to the right, the bitstrings we get do not repeat until a full period (i.e., until 16 steps).

This is similar to Gray codes. In a Gray code, you run through all bitstrings of a given size in such a way that only a single bit is changed at each step. Here, on the other hand, as you slide the window along the infinite sequence, at each step, the first bit is removed and a new bit is inserted at the end.

Can we find such nice sequences for any window length, not just 4 ?

Here is an answer for window length 2:

$$|| \ : 00\ 11 : ||$$

Here is an answer for window length 3:

$$|| \ : 000\ 101\ 11 : ||$$

What about window length 5 ?

What about replacing bits by trits $(0, 1, 2)$ or, more generally, elements of any finite set K ?

Let's give these things a name:

Definition 5.12.1. Let n and k be two positive integers, and let K be a k-element set.

A **de Bruijn sequence** of order n on K means a k^n-tuple $(c_0, c_1, \ldots, c_{k^n-1})$ of elements of K such that

(A) for each n-tuple $(a_1, a_2, \ldots, a_n) \in K^n$ of elements of K, there is a **unique** $r \in \{0, 1, \ldots, k^n - 1\}$ such that

$$(a_1, a_2, \ldots, a_n) = (c_r, c_{r+1}, \ldots, c_{r+n-1}).$$

Here, the indices under the letter "c" are understood to be periodic modulo k^n; that is, we set $c_{q+k^n} = c_q$ for each $q \in \mathbb{Z}$ (so that $c_{k^n} = c_0$ and $c_{k^n+1} = c_1$ and so on).

We saw some examples above with $K = \{0, 1\}$ (so our sequences were bitstrings). Let us give an example with $K = \{0, 1, 2\}$ and $k = 3$ and $n = 2$. Then, the 9-tuple

$$(0, 0, 1, 1, 2, 2, 0, 2, 1)$$

is a de Bruijn sequence of order n on K, because if we label its entries as $c_0, c_1, \ldots, c_8$, then

$$\begin{aligned}
(0,0) &= (c_0, c_1), & (0,1) &= (c_1, c_2), & (0,2) &= (c_6, c_7), \\
(1,0) &= (c_8, c_9), & (1,1) &= (c_2, c_3), & (1,2) &= (c_3, c_4), \\
(2,0) &= (c_5, c_6), & (2,1) &= (c_7, c_8), & (2,2) &= (c_4, c_5).
\end{aligned}$$

Theorem 5.12.2 (de Bruijn, Sainte-Marie)**.** Let n and k be positive integers. Let K be a k-element set. Then, a de Bruijn sequence of order n on K exists.

How would you prove such a theorem? It sounds natural to encode it as a Hamiltonian path problem, similarly to Gray codes. Gray codes were hamcs (Hamiltonian cycles) on a graph where a change in a single bit was encoded as an edge. Here, we can consider the digraph where a "shift" (i.e., removing the first entry and inserting a new entry at the end) is encoded as an arc. So the theorem claims that this digraph has a hamc.

What is the problem with this approach? Basically nothing is known about hamcs. Just knowing that we need a hamc doesn't help us find it.

Next time, we will see how to get around this. And not only will we prove that a de Bruijn sequence exists, but we will even prove an exact formula for how many such sequences there are!

Theorem 5.12.3. Let n and k be positive integers. Let K be a k-element set. Then,

$$(\text{\# of de Bruijn sequences of order } n \text{ on } K) = k!^{k^{n-1}}.$$

We will prove this, too!

Lecture 14

Last time, we said:

Definition 5.12.4. Let n and k be two positive integers, and let K be a k-element set.

A **de Bruijn sequence** of order n on K means a k^n-tuple $(c_0, c_1, \ldots, c_{k^n-1})$ of elements of K such that

(A) for each n-tuple $(a_1, a_2, \ldots, a_n) \in K^n$ of elements of K, there is a **unique** $r \in \{0, 1, \ldots, k^n - 1\}$ such that

$$(a_1, a_2, \ldots, a_n) = (c_r, c_{r+1}, \ldots, c_{r+n-1}).$$

Here, the indices under the letter "c" are understood to be periodic modulo k^n; that is, we set $c_{q+k^n} = c_q$ for each $q \in \mathbb{Z}$ (so that $c_{k^n} = c_0$ and $c_{k^n+1} = c_1$ and so on).

Theorem 5.12.5 (de Bruijn, Sainte-Marie)**.** Let n and k be positive integers. Let K be a k-element set. Then, a de Bruijn sequence of order n on K exists.

Theorem 5.12.6. Let n and k be positive integers. Let K be a k-element set. Then,

$$(\text{\# of de Bruijn sequences of order } n \text{ on } K) = k!^{k^{n-1}}.$$

Today, we shall prove these two theorems.

Proof of the first theorem. Let us reinterpret de Bruijn cycles as Eulerian circuits of a certain digraph. Let D be the multidigraph $\left(K^{n-1}, K^n, \psi\right)$, where $\psi : K^n \to K^{n-1} \times K^{n-1}$ is the map given by

$$\psi(a_1, a_2, \ldots, a_n) = ((a_1, a_2, \ldots, a_{n-1}), (a_2, a_3, \ldots, a_n)).$$

Thus, the vertices of D are the $(n-1)$-tuples (not the n-tuples!) of elements of K, and there is an arc from each $(n-1)$-tuple i to each $(n-1)$-tuple j if j can be obtained from i by dropping the first entry and inserting a new entry at the end. (NB: If $n = 1$, then D has only one vertex but n arcs. Anyway, the $n = 1$ case is trivial.)

Let us make some observations about D:

1. The multidigraph D is strongly connected.

[*Proof:* If $i = (i_1, i_2, \ldots, i_{n-1})$ and $j = (j_1, j_2, \ldots, j_{n-1})$ are any two vertices of D, then there is a walk from i to j going as follows:

$$\begin{aligned} i &= (i_1, i_2, \ldots, i_{n-1}) \\ &\to (i_2, i_3, \ldots, i_{n-1}, j_1) \\ &\to (i_3, i_4, \ldots, i_{n-1}, j_1, j_2) \\ &\to \cdots \\ &\to (j_1, j_2, \ldots, j_{n-1}) = j. \end{aligned}$$

Note that this walk has length $n-1$, and is the unique walk from i to j that has length $n-1$. This will come useful further below.]

2. Thus, the multidigraph D is weakly connected.

3. The multidigraph D is balanced, and in fact each vertex of D has outdegree k and indegree k.

   [*Proof:* Just remember what the arcs of D are.]

4. The digraph D has a Eulerian circuit.

   [*Proof:* The digraph D is balanced and weakly connected, so diEuler–diHierholtzer yields the claim.

   Alternatively, we can derive this from the BEST theorem: Pick an arbitrary arc a of D, and let r be its source. Then, r is a from-root of D (since D is strongly connected), and thus D has a sparb from r. In other words, using the notations of the BEST theorem, we have $\tau(D, r) \neq 0$. Moreover, each vertex of D has outdegree $k > 0$. Thus, the BEST theorem yields

$$\varepsilon(D, a) = \underbrace{\tau(D, r)}_{\neq 0} \cdot \underbrace{\prod_{u \in V} \left(\deg^- u - 1\right)!}_{\neq 0} \neq 0.$$

   In other words, D has an Eulerian circuit whose last arc is a.]

So we know that D has an Eulerian circuit $\mathbf{c}$. This circuit leads to a de Bruijn sequence as follows:

Let $p_0, p_1, \ldots, p_{k^n-1}$ be the arcs of $\mathbf{c}$ (from first to last). Extend the subscripts periodically modulo k^n (that is, set $p_{q+k^n} = p_q$ for each $q \in \mathbb{N}$). Thus, we obtain an infinite periodic walk with arcs $p_0, p_1, p_2, \ldots$, which repeats itself every k^n steps.

Then, for each $i \in \mathbb{N}$, the last $n-1$ entries of p_i are the first $n-1$ entries of p_{i+1} (since the target of p_i is the source of p_{i+1}). Hence, for each $i \in \mathbb{N}$ and each $s \in \{1, 2, \ldots, n\}$, we have

$$\begin{aligned}
&(\text{the } s\text{-th entry of } p_i) \\
&= (\text{the } (s-1)\text{-st entry of } p_{i+1}) \\
&= (\text{the } (s-2)\text{-nd entry of } p_{i+2}) \qquad (\text{by the same reasoning}) \\
&= \cdots \\
&= (\text{the 1-st entry of } p_{i+s-1}).
\end{aligned}$$

Consequently, for each $i \in \mathbb{N}$, the n entries of p_i (from first to last) are precisely the first entries of the n-tuples $p_i, p_{i+1}, \ldots, p_{i+n-1}$.

In other words, for each $i \in \mathbb{N}$, the n-tuple formed of the first entries of the n-tuples $p_i, p_{i+1}, \ldots, p_{i+n-1}$ is exactly p_i.

Hence, as i ranges from 0 to $k^n - 1$, this n-tuple takes each possible value in K^n exactly once (since p_i takes each possible value in K^n exactly once). This means that the sequence of the first entries of $p_0, p_1, \ldots, p_{k^n-1}$ is a de Bruijn sequence of order n on K. Hence, we have shown that such a sequence exists. □

Thus the first of our theorems is proved.

De Bruijn sequences have many variants and extensions. I give a few references in Spring 2022 Lecture 20. There are also recent results on "universal cycles".

How do we prove the second theorem? I.e., how do we count the de Bruijn cycles?

We could piggyback on our above proof, but we would need a way to compute $\tau(D, r)$ for our digraph D. Since our D is not of the form G^{bidir}, we cannot use the undirected MTT. However, fortunately, D is balanced, and there is a generalization of the undirected MTT for balanced digraphs:

Theorem 5.12.7 (balanced MTT)**.** Let $D = (V, A, \psi)$ be a balanced multidigraph. Assume that $V = \{1, 2, \ldots, n\}$ for some positive integer n.

Let L be the Laplacian of D. Then:

(a) For any vertex r of D, we have

$$(\#\text{ of sparbs of } D \text{ to } r) = \det(L_{\sim r, \sim r}).$$

Moreover, this number does not depend on r.

(b) Let t be an indeterminate. Expand the determinant $\det(tI_n + L)$ (here, I_n denotes the $n \times n$ identity matrix) as a polynomial in t:

$$\det(tI_n + L) = c_n t^n + c_{n-1} t^{n-1} + \cdots + c_1 t^1 + c_0 t^0,$$

where $c_0, c_1, \ldots, c_n$ are numbers. (Note that this is the characteristic polynomial of L, up to substituting $-t$ for t and possibly multiplying by a power of -1.) Then, for any vertex r of D, we have

$$(\text{\# of sparbs of } D \text{ to } r) = \frac{1}{n} c_1.$$

(c) Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of L, listed in such a way that $\lambda_n = 0$. Then, for any vertex r of D, we have

$$(\text{\# of sparbs of } D \text{ to } r) = \frac{1}{n} \cdot \lambda_1 \lambda_2 \cdots \lambda_{n-1}.$$

(d) Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of L, listed in such a way that $\lambda_n = 0$. If all vertices of D have outdegree > 0, then

$$\begin{aligned}&(\text{\# of Eulerian circuits of } D)\\ &= |A| \cdot \frac{1}{n} \cdot \lambda_1 \lambda_2 \cdots \lambda_{n-1} \cdot \prod_{u \in V} \left(\deg^+ u - 1\right)!.\end{aligned}$$

Proof. **(a)** The equality comes from the MTT. The independence from r was a corollary of the BEST theorem.

(b) follows from **(a)** just as in the undirected case.

(c) follows from **(b)** just as in the undirected case.

(d) Assume that all vertices of D have outdegree > 0. Then,

$$\begin{aligned}&(\text{\# of Eulerian circuits of } D)\\ &= \sum_{a \in A} (\text{\# of Eulerian circuits of } D \text{ whose first arc is } a).\end{aligned}$$

However, if $a \in A$ is any arc, and if r is the source of A, then the WORST theorem yields

$$\begin{aligned}&(\text{\# of Eulerian circuits of } D \text{ whose first arc is } a)\\ &= (\text{\# of sparbs of } D \text{ to } r) \cdot \prod_{u \in V} \left(\deg^+ u - 1\right)!\\ &= \frac{1}{n} \cdot \lambda_1 \lambda_2 \cdots \lambda_{n-1} \cdot \prod_{u \in V} \left(\deg^+ u - 1\right)!.\end{aligned}$$

Hence,

$$\begin{aligned}
&(\text{\# of Eulerian circuits of } D) \\
&= \sum_{a\in A} \underbrace{(\text{\# of Eulerian circuits of } D \text{ whose first arc is } a)}_{=\frac{1}{n}\cdot\lambda_1\lambda_2\cdots\lambda_{n-1}\cdot\prod_{u\in V}\left(\deg^+ u-1\right)!} \\
&= \sum_{a\in A} \frac{1}{n}\cdot\lambda_1\lambda_2\cdots\lambda_{n-1}\cdot\prod_{u\in V}\left(\deg^+ u-1\right)! \\
&= |A|\cdot\frac{1}{n}\cdot\lambda_1\lambda_2\cdots\lambda_{n-1}\cdot\prod_{u\in V}\left(\deg^+ u-1\right)!.
\end{aligned}$$

□

Now, let us come back to the proof of the second theorem about de Bruijn circuits – namely, compute their number.

Recall the digraph D that we constructed in our proof of the first theorem. We constructed a de Bruijn sequence of order n on K from an Eulerian circuit of D. This actually works both ways. Thus, we get a bijection

$$\begin{aligned}
\{\text{Eulerian circuits of } D\} &\to \{\text{de Bruijn sequences of order } n \text{ on } K\}, \\
\mathbf{c} &\mapsto (\text{the sequence of first entries of the arcs of } \mathbf{c}).
\end{aligned}$$

Hence,

$$\begin{aligned}
&(\text{\# of de Bruijn sequences of order } n \text{ on } K) \\
&= (\text{\# of Eulerian circuits of } D) \\
&= |K^n|\cdot\frac{1}{k^{n-1}}\cdot\lambda_1\lambda_2\cdots\lambda_{k^{n-1}-1}\cdot\prod_{u\in K^{n-1}}\left(\deg^+ u-1\right)!
\end{aligned}$$

(by part **(d)** of the balanced MTT), where $\lambda_1, \lambda_2, \ldots, \lambda_{k^{n-1}}$ are the eigenvalues of the Laplacian L of D, indexed such that $\lambda_{k^{n-1}} = 0$. (Note that k^{n-1} is the number of vertices of D; this was called n in the balanced MTT).

Some parts of the above equality simplify easily:

$$|K^n|\cdot\frac{1}{k^{n-1}} = k^n\cdot\frac{1}{k^{n-1}} = k$$

and

$$\prod_{u\in K^{n-1}}\left(\underbrace{\deg^+ u}_{=k}-1\right)! = \prod_{u\in K^{n-1}}(k-1)! = (k-1)!^{k^{n-1}}.$$

It remains to find $\lambda_1\lambda_2\cdots\lambda_{k^{n-1}-1}$. What are the eigenvalues of L ?

The Laplacian L of our digraph D is a $k^{n-1}\times k^{n-1}$-matrix whose rows and columns are indexed by $(n-1)$-tuples in K^{n-1}. Strictly speaking, we should relabel these tuples as $1, 2, \ldots, k^{n-1}$, in order to make D a "proper matrix" with a well-defined order on rows and columns. But let's not do this, and keep it a "matrix in the wider sense of this word".

Let C be the adjacency matrix of the digraph D; this is the $k^{n-1} \times k^{n-1}$-matrix (again with rows and columns indexed by the $(n-1)$-tuples in K^{n-1}) whose (i, j)-th entry is the # of arcs with source i and target j. Note that the loops of D are precisely the arcs of the form $(x, x, \ldots, x)$ for $x \in K$; thus, D has exactly k loops. Hence, the trace of C is k.

Recall the definition of the Laplacian L. We can restate it as follows:

$$L = \Delta - C,$$

where Δ is the diagonal matrix whose diagonal entries are the outdegrees of the vertices of D. Since each vertex of D has outdegree k, the latter diagonal matrix Δ is just $k \cdot I$, where I is the identity matrix of size k^{n-1}. So our above equation rewrites as

$$L = k \cdot I - C.$$

Hence, if $\gamma_1, \gamma_2, \ldots, \gamma_{k^{n-1}}$ are the eigenvalues of C, then $k - \gamma_1, k - \gamma_2, \ldots, k - \gamma_{k^{n-1}}$ are the eigenvalues of L. Computing the former will therefore help us find the latter.

Furthermore, let J be the $k^{n-1} \times k^{n-1}$-matrix whose all entries are 1. It is easy to see that the eigenvalues of J are

$$\underbrace{0, 0, \ldots, 0}_{k^{n-1}-1 \text{ many zeroes}}, k^{n-1}$$

(easy linear algebra).

Now, we observe that

$$C^{n-1} = J.$$

[*Proof:* Let i and j be two vertices of D. Then, the (i, j)-th entry of C^{n-1} is the # of length-$(n-1)$ walks from i to j (by homework set #4 exercise 4 **(a)**), and thus equals 1 as we have seen above. In other words, the (i, j)-th entry of C^{n-1} equals the (i, j)-th entry of J. Thus, $C^{n-1} = J$.]

How does this help us compute the eigenvalues of C ?

Well, let $\gamma_1, \gamma_2, \ldots, \gamma_{k^{n-1}}$ be the eigenvalues of C. Then, for any $\ell \in \mathbb{N}$, the eigenvalues of C^ℓ are $\gamma_1^\ell, \gamma_2^\ell, \ldots, \gamma_{k^{n-1}}^\ell$ (this can be shown, e.g., using triangularization or Jordan normal form). Applying this to $\ell = n - 1$, we conclude that the eigenvalues of J are $\gamma_1^{n-1}, \gamma_2^{n-1}, \ldots, \gamma_{k^{n-1}}^{n-1}$ (since $C^{n-1} = J$). But we already know that these eigenvalues are $\underbrace{0, 0, \ldots, 0}_{k^{n-1}-1 \text{ many zeroes}}, k^{n-1}$.

Hence,

$$\left(\gamma_1^{n-1}, \gamma_2^{n-1}, \ldots, \gamma_{k^{n-1}}^{n-1}\right) = \left(\underbrace{0, 0, \ldots, 0}_{k^{n-1}-1 \text{ many zeroes}}, k^{n-1}\right) \text{ up to rearrangement.}$$

By rearranging the γ's, we thus obtain

$$\left(\gamma_1^{n-1}, \gamma_2^{n-1}, \ldots, \gamma_{k^{n-1}}^{n-1}\right) = \left(\underbrace{0, 0, \ldots, 0}_{k^{n-1}-1 \text{ many zeroes}}, k^{n-1} \right).$$

Hence, $\gamma_1 = \gamma_2 = \cdots = \gamma_{k^{n-1}-1} = 0$. (We cannot get $\gamma_{k^{n-1}} = k$ yet, since there might be several $(n-1)$-st roots of k.)

However, the sum of all eigenvalues of a matrix is its trace. Thus, $\gamma_1 + \gamma_2 + \cdots + \gamma_{k^{n-1}} = \operatorname{Tr} C = k$ (as we proved above). Hence, $\gamma_1 = \gamma_2 = \cdots = \gamma_{k^{n-1}-1} = 0$ entails $\gamma_{k^{n-1}} = k$.

Thus, the eigenvalues of L are $k, k, \ldots, k, 0$ (since $k - 0 = k$ and $k - k = 0$). Hence, in particular, $\lambda_1 = \lambda_2 = \cdots = \lambda_{k^{n-1}-1} = k$ and $\lambda_{k^{n-1}} = 0$. Therefore, $\lambda_1 \lambda_2 \cdots \lambda_{k^{n-1}-1} = k^{k^{n-1}-1}$.

Now, our above equality becomes

$$\begin{aligned}
&(\text{\# of de Bruijn sequences of order } n \text{ on } K) \\
&= \underbrace{|K^n| \cdot \frac{1}{k^{n-1}}}_{=k} \cdot \underbrace{\lambda_1 \lambda_2 \cdots \lambda_{k^{n-1}-1}}_{=k^{k^{n-1}-1}} \cdot \underbrace{\prod_{u \in K^{n-1}} \left(\deg^+ u - 1\right)!}_{=(k-1)!^{k^{n-1}}} \\
&= \underbrace{k \cdot k^{k^{n-1}-1}}_{=k^{k^{n-1}}} \cdot (k-1)!^{k^{n-1}} = k^{k^{n-1}} \cdot (k-1)!^{k^{n-1}} \\
&= \left(k \cdot (k-1)!\right)^{k^{n-1}} = k!^{k^{n-1}}.
\end{aligned}$$

So the second theorem is proved.

(There is a combinatorial proof in a paper by Bidkhori and Kishore 2011.)

5.13. On the left nullspace of the Laplacian

Much more can be said about the Laplacian of a digraph. The study of matrices associated to a graph or digraph is known as **spectral graph theory**. The original form of the MTT was found by Gustav Kirchhoff in his study of electric networks. Instead of counting spanning trees, he worked with "weighted counts", i.e., sums of weights. We will discuss this in the next section.

Another use of Laplacians is a "canonical" way to draw a graph, called "spectral layout".

Let me mention one more result about Laplacians of digraphs. Recall that the Laplacian L of a digraph D always satisfies $Le = 0$, where $e = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$. Thus, the vector e belongs to the right nullspace (= right kernel) of L. It is not hard to see that if D has a to-root and we are working over a characteristic-0 field, then e spans this nullspace, i.e., there are no vectors in that nullspace other than scalar multiples of e. (This is actually an "if and only if".)

What about the left nullspace? Can we find an explicit nonzero vector f such that $fL = 0$? Yes, we can:

Theorem 5.13.1 (harmonic vector theorem for Laplacians)**.** Let $D = (V, A, \psi)$ be a multidigraph, where $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$. For each $r \in V$, let $\tau(D, r)$ be the # of sparbs of D to r. Let f be the row vector

$$(\tau(D,1), \ \tau(D,2), \ \ldots, \ \tau(D,n)).$$

Then, $fL = 0$.

Proof. Homework set #7 exercise 1 **(b)**. □

Among the applications of this theorem is a way to compute the steady state of a Markov chain.

5.14. A weighted Matrix-Tree Theorem

We have so far been **counting** arborescences. A natural generalization of counting is **weighted counting** – i.e., you assign a certain number (a "weight") to each arborescence (or whatever else you're counting), and then you **sum** these numbers. In particular, if all the weights are 1, then you just get the number of the objects you're looking at.

If you pick the weights to be completely random, you usually get a result that doesn't simplify. Good results can often be obtained if the weights follow certain patterns. In the case of sparbs of a given digraph D, we can find a nice formula in the case when the weight of an arborescence is the product of the weights of the arcs of this arborescence.

Next time, we will see this.

Lecture 15

Last time, I promised to generalize the MTT (= Matrix-Tree Theorem) from just an equality between numbers to a "weighted equality".

Specifically, I will assign a weight to each edge (or arc), and then let the weight of a spanning tree (or sparb) be the product of the weights of its edges (or arcs).

Definition 5.14.1. Let $D = (V, A, \psi)$ be multidigraph.

Let $\mathbb{K}$ be a commutative ring. (Feel free to assume that $\mathbb{K}$ is a polynomial ring over $\mathbb{Z}$, or the ring $\mathbb{R}$, or even $\mathbb{Z}$ itself.)

Assume that an element $w_a \in \mathbb{K}$ is assigned to each arc $a \in A$. We call this w_a the **weight** of a. (If $\mathbb{K} = \mathbb{R}$, then these weights are real numbers.)

- **(a)** For any two vertices $i, j \in V$, we let $a^w_{i,j}$ be the sum of the weights of all arcs of D that have source i and target j.
- **(b)** For any vertex $i \in V$, we define the **weighted outdegree** $\deg^{+w} i$ of i to be the sum
$$\sum_{\substack{a \in A;\\ \text{the source of } a \text{ is } i}} w_a.$$
- **(c)** If B is a subdigraph of D, then the **weight** $w(B)$ of B is defined to be the product $\prod_{a \text{ is an arc of } B} w_a$. This is the product of the weights of all arcs of B.
- **(d)** Assume that $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$. The **weighted Laplacian** L^w of D (with respect to the weights w_a) is defined to be the $n \times n$-matrix $L^w \in \mathbb{K}^{n \times n}$ whose entries are given by
$$L^w_{i,j} = \left(\deg^{+w} i\right) \cdot [i = j] - a^w_{i,j} \qquad \text{for all } i, j \in V.$$

If we set all weights w_a equal to 1, then these notions specialize back to the original notions of $a_{i,j}$, outdegree and Laplacian.

We can now generalize the MTT as follows:

Theorem 5.14.2 (weighted MTT)**.** Let $D = (V, A, \psi)$ be a multidigraph.

Let $\mathbb{K}$ be a commutative ring. Assume that a weight $w_a \in \mathbb{K}$ is assigned to each arc $a \in A$.

Assume that $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$. Let L^w be the weighted Laplacian of D.

Let r be a vertex of D. Then,
$$\sum_{\substack{B \text{ is a sparb}\\ \text{of } D \text{ to } r}} w(B) = \det\left(L^w_{\sim r, \sim r}\right).$$

Example 5.14.3. Let D be as on the blackboard (a digraph with 3 vertices $1, 2, 3$ and 4 arcs $\alpha, \beta, \gamma, \delta$ with $\psi(\alpha) = (1, 2)$ and $\psi(\beta) = (2, 3)$ and $\psi(\gamma) = \psi(\delta) = (1, 3)$). Then, the weighted Laplacian L^w is

$$L^w = \begin{pmatrix} w_\alpha + w_\gamma + w_\delta & -w_\alpha & -w_\gamma - w_\delta \\ 0 & w_\beta & -w_\beta \\ 0 & 0 & 0 \end{pmatrix}.$$

Also, for $r = 3$, we have

$$\sum_{\substack{B \text{ is a sparb} \\ \text{of } D \text{ to } r}} w(B) = w_\alpha w_\beta + w_\beta w_\gamma + w_\beta w_\delta.$$

The weighted MTT theorem says that this equals

$$\begin{aligned} \det\left(L^w_{\sim 3, \sim 3}\right) &= \det \begin{pmatrix} w_\alpha + w_\gamma + w_\delta & -w_\alpha \\ 0 & w_\beta \end{pmatrix} \backslash \\ &= (w_\alpha + w_\gamma + w_\delta) w_\beta - (-w_\alpha) 0. \end{aligned}$$

So it says that

$$\begin{aligned} & w_\alpha w_\beta + w_\beta w_\gamma + w_\beta w_\delta \\ &= (w_\alpha + w_\gamma + w_\delta) w_\beta - (-w_\alpha) 0. \end{aligned}$$

This is a polynomial identity in $w_\alpha, w_\beta, w_\gamma, w_\delta$.

As we already said, the weighted MTT becomes the original MTT if we set all w_a equal to 1.

However, we shall now go backwards: We will derive the weighted MTT from the original MTT.

First, we recall a standard result in algebra, known as the **principle of permanence of polynomial identities** or as the **polynomial identity trick** (or under several other names). Here is one incarnation of this principle:

Theorem 5.14.4 (principle of permanence of polynomial identities)**.** Let $P(x_1, x_2, \ldots, x_m)$ and $Q(x_1, x_2, \ldots, x_m)$ be two polynomials with integer coefficients in several indeterminates $x_1, x_2, \ldots, x_m$. Assume that the equality

$$P(k_1, k_2, \ldots, k_m) = Q(k_1, k_2, \ldots, k_m)$$

holds for every m-tuple $(k_1, k_2, \ldots, k_m) \in \mathbb{N}^m$ of nonnegative integers. Then, $P(x_1, x_2, \ldots, x_m)$ and $Q(x_1, x_2, \ldots, x_m)$ are identical as polynomials (so that, in particular, the equality

$$P(k_1, k_2, \ldots, k_m) = Q(k_1, k_2, \ldots, k_m)$$

holds not only for every $(k_1, k_2, \ldots, k_m) \in \mathbb{N}^m$ but also for every $(k_1, k_2, \ldots, k_m) \in \mathbb{Z}^m$ and for every $(k_1, k_2, \ldots, k_m) \in \mathbb{C}^m$ and more generally for every $(k_1, k_2, \ldots, k_m) \in \mathbb{K}^m$ for any commutative ring $\mathbb{K}$).

In other words, this is saying that if you want to prove that two polynomials (with integer coefficients) are equal, it suffices to prove that they are equal on all nonnegative integer inputs. For example, if you can prove the equality

$$(x+y)^4 + (x-y)^4 = 2x^4 + 12x^2y^2 + 2y^4$$

for all $x, y \in \mathbb{N}$, then you automatically can conclude (by the above theorem) that it holds for all $x, y \in \mathbb{K}$ for any commutative ring $\mathbb{K}$.

The theorem is often used in combinatorics to prove equalities between binomial coefficients. For instance, the Chu–Vandermonde identity

$$\binom{x+y}{n} = \sum_{k=0}^{n} \binom{x}{k} \binom{y}{n-k}$$

(where $n \in \mathbb{N}$) is not hard to verify for all $x, y \in \mathbb{N}$ (do it if you haven't seen it before!), but both of its sides are polynomials in x and y with rational coefficients (for example, $\binom{x+y}{n} = \dfrac{(x+y)(x+y-1)\cdots(x+y-n+1)}{n!}$), so you can apply the above theorem to it after multiplying by the common denominator (which is $n!$), and conclude that the identity holds for any $x, y \in \mathbb{R}$. Many other identities are proved in this way.

Proofs of the theorem can be found in various places, in particular in most good texts on abstract algebra (I give a couple references in Spring 2022 Lecture 21).

Now, let us apply the theorem to prove the weighted MTT.

Proof of the weighted MTT.. Both sides of the claim are polynomials (with integer coefficients) in the weights w_a. Thus, by the principle of permanence, we can prove it by showing that it holds when all arc weights w_a are nonnegative integers.

So let us WLOG assume that w_a are nonnegative integers.

Let us now replace each arc a of D by w_a many copies of a (having the same source as a and the same target as a). The result is a new digraph D'.

This digraph D' has the same vertices as D, but each arc a of D has turned into w_a arcs of D'. Thus, the weighted outdegree $\deg^{+w} i$ of a vertex i of D equals its usual (i.e., non-weighted) outdegree $\deg^+ i$ in D'. Moreover, each subdigraph B of D gives rise to $w(B)$ many subdigraphs of D' (because each arc a of B can be replaced by any of its w_a many copies in D'). Hence,

$$\sum_{\substack{B \text{ is a sparb} \\ \text{of } D \text{ to } r}} w(B) = (\# \text{ of sparbs of } D' \text{ rooted to } r).$$

Moreover, the weighted Laplacian L^w of D is the usual (i.e., non-weighted) Laplacian of D'. Altogether, this shows that applying the original MTT to D' yields the weighted MTT for D. And this completes the proof.

[*Remark:* Alternatively, you can adapt our proof of the original MTT to the weighted case.] □

The weighted MTT has some applications that wouldn't be obvious from the original MTT. Here is one:

Exercise 3. Let $n \geq 2$ be an integers, and let $d_1, d_2, \ldots, d_n$ be n positive integers.

An n-**tree** shall mean a simple graph with vertex set $\{1, 2, \ldots, n\}$ that is a tree.

We know from Cayley's theorem that there are n^{n-2} many n-trees.

How many of these n-trees have the property that

$$\deg i = d_i \qquad \text{for each vertex } i \text{ ?}$$

Solution. The n-trees are just the spanning trees of the complete graph K_n.

To incorporate the $\deg i = d_i$ condition into our count, we use a generating function. So let us **not** fix the numbers $d_1, d_2, \ldots, d_n$, but rather consider the polynomial

$$P(x_1, x_2, \ldots, x_n) = \sum_{T \text{ is an } n\text{-tree}} x_1^{\deg 1} x_2^{\deg 2} \cdots x_n^{\deg n},$$

where $\deg i$ means the degree of i in T. Then, the $x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}$-coefficient of this polynomial $P(x_1, x_2, \ldots, x_n)$ is the # of n-trees T satisfying

$$\deg i = d_i \qquad \text{for each vertex } i.$$

Thus, to solve our problem, we need to compute the $x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}$-coefficient of this polynomial $P(x_1, x_2, \ldots, x_n)$.

Let us assign to each edge ij of K_n the weight $w_{ij} := x_i x_j$. Then, the definition of $P(x_1, x_2, \ldots, x_n)$ rewrites as follows:

$$P(x_1, x_2, \ldots, x_n) = \sum_{T \text{ is an } n\text{-tree}} w(T),$$

where $w(T)$ denotes the product of the weights of all edges of T. (Indeed, for any subgraph T of K_n, the weight $w(T)$ equals $x_1^{\deg 1} x_2^{\deg 2} \cdots x_n^{\deg n}$.)

We have assigned weights to the edges of the graph K_n; let us now assign the same weights to the arcs of the digraph K_n^{bidir} by setting

$$w_{(ij,1)} = w_{(ij,2)} = w_{ij} = x_i x_j.$$

As we already have seen a few times, we can replace spanning trees of K_n by sparbs of K_n^{bidir} rooted to 1, since the former are in bijection to the latter. Thus,

$$\begin{aligned}&(\text{\# of spanning trees of } K_n)\\&= \left(\text{\# of sparbs of } K_n^{\text{bidir}} \text{ rooted to } 1\right).\end{aligned}$$

Moreover, since this bijection preserves weights, we also have

$$\sum_{\substack{T \text{ is a spanning}\\ \text{tree of } K_n}} w(T) = \sum_{\substack{B \text{ is a sparb}\\ \text{of } K_n^{\text{bidir}} \text{ to } 1}} w(B).$$

In other words,

$$\sum_{T \text{ is an } n\text{-tree}} w(T) = \sum_{\substack{B \text{ is a sparb}\\ \text{of } K_n^{\text{bidir}} \text{ to } 1}} w(B).$$

We shall now compute the RHS using the weighted MTT. To do so, we need the weighted Laplacian L^w of K_n^{bidir}. Its entries are

$$\begin{aligned}L^w_{i,j} &= \left(\deg^{+w} i\right) \cdot [i = j] - a^w_{i,j}\\&= \begin{cases} -x_i x_j, & \text{if } i \neq j;\\ x_i x_1 + x_i x_2 + \cdots + x_i x_{i-1} + x_i x_{i+1} + \cdots + x_i x_n, & \text{if } i = j\end{cases}\\&= \begin{cases} -x_i x_j, & \text{if } i \neq j;\\ x_i \left(x_1 + x_2 + \cdots + x_{i-1} + x_{i+1} + \cdots + x_n\right) & \text{if } i = j\end{cases}\\&= x_i \left(x_1 + x_2 + \cdots + x_n\right) \cdot [i = j] - x_i x_j.\end{aligned}$$

We can find its minor $\det\left(L^w_{\sim 1, \sim 1}\right)$ without too much trouble (e.g., using row transformations similar to the ones we used in the proof of Cayley's theorem). The result is

$$\det\left(L^w_{\sim 1, \sim 1}\right) = x_1 x_2 \cdots x_n \left(x_1 + x_2 + \cdots + x_n\right)^{n-2}.$$

Summarizing, we see that

$$\begin{aligned}P\left(x_1, x_2, \ldots, x_n\right) &= \sum_{T \text{ is an } n\text{-tree}} w(T)\\&= \sum_{\substack{B \text{ is a sparb}\\ \text{of } K_n^{\text{bidir}} \text{ to } 1}} w(B)\\&= \det\left(L^w_{\sim 1, \sim 1}\right) \qquad \text{(by the weighted MTT)}\\&= x_1 x_2 \cdots x_n \left(x_1 + x_2 + \cdots + x_n\right)^{n-2}.\end{aligned}$$

Thus,

$$\begin{aligned}
&\left(\text{the } x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}\text{-coefficient of } P\left(x_1, x_2, \ldots, x_n\right)\right) \\
&= \left(\text{the } x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}\text{-coefficient of } x_1 x_2 \cdots x_n \left(x_1 + x_2 + \cdots + x_n\right)^{n-2}\right) \\
&= \left(\text{the } x_1^{d_1-1} x_2^{d_2-1} \cdots x_n^{d_n-1}\text{-coefficient of } \left(x_1 + x_2 + \cdots + x_n\right)^{n-2}\right).
\end{aligned}$$

Now we are looking for the latter coefficient. More generally, what are the coefficients of a power $(x_1 + x_2 + \cdots + x_n)^m$?

These are the **multinomial coefficients** (named in analogy to the binomial coefficients, which are the $n = 2$ case). They are defined as follows: If $p_1, p_2, \ldots, p_n, q$ are nonnegative integers with $q = p_1 + p_2 + \cdots + p_n$, then the **multinomial coefficient** $\binom{q}{p_1, p_2, \ldots, p_n}$ is defined to be $\dfrac{q!}{p_1! p_2! \cdots p_n!}$. The **multinomial formula/theorem** says that

$$(x_1 + x_2 + \cdots + x_n)^m = \sum_{\substack{i_1, i_2, \ldots, i_n \in \mathbb{N};\\ i_1 + i_2 + \cdots + i_n = m}} \binom{m}{i_1, i_2, \ldots, i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}.$$

(Compare: How many anagrams does the word "anagram" have? These are the words with 3 a's, 1 g, 1 m, 1 n and 1 r, so they correspond to the terms $a^3 g^1 m^1 n^1 r^1$ in $(a + g + m + n + r)^7$, and thus their number is $\binom{7}{3, 1, 1, 1, 1} = \dfrac{7!}{3! \cdot 1! \cdot 1! \cdot 1! \cdot 1!} = 840$. Of course, most of these anagrams are meaningless.)

Now back to our problem:

$$\begin{aligned}
&\left(\text{the } x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}\text{-coefficient of } P\left(x_1, x_2, \ldots, x_n\right)\right) \\
&= \left(\text{the } x_1^{d_1-1} x_2^{d_2-1} \cdots x_n^{d_n-1}\text{-coefficient of } \left(x_1 + x_2 + \cdots + x_n\right)^{n-2}\right) \\
&= \binom{n-2}{d_1 - 1,\ d_2 - 1,\ \ldots,\ d_n - 1}.
\end{aligned}$$

But as we recall, the $x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}$-coefficient of $P(x_1, x_2, \ldots, x_n)$ is the # of n-trees T satisfying

$$\deg i = d_i \qquad \text{for each vertex } i.$$

Thus, we have proved:

Theorem 5.14.5 (refined Cayley's formula)**.** Let $n \geq 2$ be an integers. Let $d_1, d_2, \ldots, d_n$ be n positive integers. Then, the # of n-trees T satisfying

$$\deg i = d_i \qquad \text{for each vertex } i$$

is the multinomial coefficient

$$\binom{n-2}{d_1 - 1,\ d_2 - 1,\ \ldots,\ d_n - 1}.$$

Note that this theorem is not very hard to prove by induction on n. (Think about it!)

The harmonic vector theorem for Laplacians also has a weighted version (see Theorem 1.1.7 in Spring 2022 Lecture 21).

6. Colorings

Now to something completely different: Let's color the vertices of a graph!

6.1. Definitions

Our colors will be positive integers. Coloring the vertices thus means assigning a positive integer to each vertex (called the color of this vertex). Here are the details:

Definition 6.1.1. Let $G = (V, E, \varphi)$ be a multigraph. Let $k \in \mathbb{N}$.

(a) A **k-coloring** of G means a map $f : V \to \{1, 2, \ldots, k\}$. Given such a k-coloring f, we refer to the numbers $1, 2, \ldots, k$ as the **colors**, and we refer to each value $f(v)$ as the **color** of the vertex v in the k-coloring f.

(b) A k-coloring f of G is said to be **proper** if no two adjacent vertices of G have the same color. (In other words, it is proper if and only if there exists no edge of G whose endpoints u and v satisfy $f(u) = f(v)$).

See Spring 2022 Lecture 21 for examples.

As we see, some graphs have proper 3-colorings, but others don't. Clearly, having 4 mutually adjacent vertices makes a proper 3-coloring impossible (by the pigeonhole principle), but this is not an "if and only if". The question of determining whether a graph has a proper 3-coloring is NP-complete.

6.2. 2-colorings

The situation for proper 2-colorings is much nicer:

Theorem 6.2.1 (2-coloring equivalence theorem)**.** Let $G = (V, E, \varphi)$ be a multigraph. Then, the following three statements are equivalent:

- **B1:** The graph G has a proper 2-coloring.
- **B2:** The graph G has no cycles of odd length.
- **B3:** The graph G has no circuits of odd length.

Let us prove this theorem. First, we recall/introduce some notation:

- A walk is said to be **odd-length** if it has odd length.
- A walk **w** is said to **contain** a walk **v** if each edge of **v** is an edge of **w**.
- A **circuit** means a closed walk.

Now, we prove an auxiliary result:

Proposition 6.2.2. Let G be a multigraph. Let u and v be two vertices of G. Let **w** be an odd-length walk from u to v. Then, **w** contains either an odd-length **path** from u to v or an odd-length **cycle** (or both).

Proof. Suppose that **w** is not a path. Then, writing **w** as $\mathbf{w} = (w_0, *, w_1, *, w_2, \ldots, *, w_k)$, there must be $i < j$ satisfying $w_i = w_j$. Pick such a pair (i, j) with $j - i$ minimum. If $j - i$ is odd, then $\left(w_i, *, w_{i+1}, *, \ldots, *, w_j\right)$ is an odd-length cycle contained in **w** (indeed, it is backtrack-free since $j - i \neq 2$), and this immediately proves the claim. So let us assume WLOG that $j - i$ is even. Thus, cutting the cycle $\left(w_i, *, w_{i+1}, *, \ldots, *, w_j\right)$ out of **w**, we obtain another odd-length walk from u to v, which has smaller length than **w** but is contained in **w**. So we can apply the induction hypothesis if we induct on the length of **w**. □

Lecture 16

Last time, we stated but did not prove the following:

Theorem 6.2.3 (2-coloring equivalence theorem)**.** Let $G = (V, E, \varphi)$ be a multigraph. Then, the following three statements are equivalent:

- **B1:** The graph G has a proper 2-coloring.
- **B2:** The graph G has no cycles of odd length.
- **B3:** The graph G has no circuits of odd length.

But we proved the following:

Proposition 6.2.4. Let G be a multigraph. Let u and v be two vertices of G. Let $\mathbf{w}$ be an odd-length walk from u to v. Then, $\mathbf{w}$ contains either an odd-length **path** from u to v or an odd-length **cycle** (or both).

Let us prove one more simple lemma:

Lemma 6.2.5. Let G be a multigraph with a proper 2-coloring f. Let u and v be two vertices of G. Let $\mathbf{w}$ be a walk from u to v that has length k. Then,

$$f(u) - f(v) \equiv k \mod 2.$$

Proof. Let $w_0, w_1, \ldots, w_k$ be the vertices of $\mathbf{w}$, from first to last. Since f is a proper 2-coloring, we have

$$f(w_0) \neq f(w_1) \neq \cdots \neq f(w_k).$$

But since f is a 2-coloring, there are only 2 possible values for each $f(w_i)$: the colors 1 and 2. So two distinct colors must be 1 and 2 in some order. Hence, if $f(w_0) = 1$, then $f(w_1) = 2$ and $f(w_2) = 1$ and $f(w_3) = 2$ and so on (alternating between 1 and 2). Likewise, if $f(w_0) = 2$, then $f(w_1) = 1$ and $f(w_2) = 2$ and $f(w_3) = 1$ and so on. In either case, we see (by induction on i) that

$$f(w_i) \equiv f(w_0) + i \mod 2 \qquad \text{for each } i \in \{0, 1, \ldots, k\}.$$

Applying this to $i = k$, we find $f(w_k) \equiv f(w_0) + k \mod 2$. In other words, $f(v) \equiv f(u) + k \mod 2$ (since $w_0 = u$ and $w_k = v$). In other words, $f(u) - f(v) \equiv -k \equiv k \mod 2$. This proves the lemma. □

Now we are ready to prove the theorem:

Proof of the theorem. We shall prove the implications B1 $\Longrightarrow$ B2 $\Longrightarrow$ B3 $\Longrightarrow$ B1.

Proof of B1 $\Longrightarrow$ B2: Assume that B1 holds. Thus, G has a proper 2-coloring f.

Now, assume (for contradiction) that G has a cycle $\mathbf{w}$ of odd length. Let u be the starting and ending point of $\mathbf{w}$. Then, $\mathbf{w}$ is a walk from u to u of length k, where k is odd. So the previous lemma yields $f(u) - f(u) \equiv k \mod 2$. In other words, $0 \equiv k \mod 2$. But k is odd, so this cannot be.

So we got a contradiction, and conclude that G has no cycle of odd length. This proves B1 $\Longrightarrow$ B2.

Proof of B2 $\Longrightarrow$ B3: Assume that B2 holds, i.e., that G has no cycles of odd length.

We must prove that B3 holds, i.e., that G has no circuits of odd length. Assume the contrary. Thus, G has a circuit $\mathbf{w}$ of odd length. Let u be the starting and ending point of this circuit $\mathbf{w}$. Then, $\mathbf{w}$ is an odd-length walk from u to u. Hence, by today's first (or last time's last) proposition, we conclude that $\mathbf{w}$ contains either an odd-length **path** from u to u or an odd-length **cycle** (or both). Since G has no odd-length cycles, we conclude that $\mathbf{w}$ contains an odd-length path from u to u. But the only path from u to u is (u), which has even length. Contradiction. Thus, B2 $\Longrightarrow$ B3 holds.

Proof of B3 $\Longrightarrow$ B1: Assume that B3 holds, i.e., that G has no circuits of odd length. We must show that G has a proper 2-coloring.

Algorithmically, it is clear how to construct such a proper 2-coloring: We start by choosing some vertex and assigning a color to it arbitrarily, and then spreading the colors recursively to its neighbors (using the fact that if u and v are adjacent, then the color of v must be distinct from the color of u, which determines it uniquely because there are only two possible colors). This has to be done once for each component of G. But does this actually produce a proper 2-coloring?

This can be proved rigorously if we rigorously define how nondeterministic algorithms work. For us, it is easier to rephrase this algorithmic construction as a more direct definition:

We WLOG assume that G is connected (otherwise, let $C_1, C_2, \ldots, C_k$ be the components of G, and color the graphs $G[C_1]$, $G[C_2]$, $\ldots$, $G[C_k]$ separately). Thus, any two vertices u and v of G have a (finite) distance $d(u,v)$, which is the smallest length of a path from u to v.

Let $G = (V, E, \varphi)$. Fix any vertex r of G. Define a map $f : V \to \{1,2\}$ by

$$f(v) = \begin{cases} 1, & \text{if } d(v,r) \text{ is even;} \\ 2, & \text{if } d(v,r) \text{ is odd} \end{cases} \qquad \text{for each } v \in V.$$

Now I claim that f is a proper 2-coloring. To prove this, we pick two adjacent vertices u and v.

Now, consider the walk from r to r obtained by

- first going from r to u in $d(r,u) = d(u,r)$ steps;

- then going from u to v in 1 step;
- then going from v to r in $d(v,r)$ steps.

This walk is a circuit, so its length is even (since G has no odd-length circuits). But its length is $d(u,r) + 1 + d(v,r)$. So we conclude that $d(u,r) + 1 + d(v,r)$ is even. In other words, $d(u,r) + d(v,r)$ is odd. In other words, the numbers $d(u,r)$ and $d(v,r)$ have different parities. Therefore, $f(u) \neq f(v)$.

So we have shown that $f(u) \neq f(v)$ whenever u and v are two adjacent vertices of G. In other words, the 2-coloring f is proper. This proves B3 $\Longrightarrow$ B1, and completes the proof of the theorem.

[See Lecture 22 in the 2022 notes for an alternative proof.] □

Remark 6.2.6. A graph G that satisfies the three equivalent statements B1, B2 and B3 is sometimes called a "bipartite graph". For us, a bipartite graph will be a graph G **equipped with** a proper 2-coloring, which is not the same as a graph G that **has** a proper 2-coloring. The same graph G can have many proper 2-colorings.

How many?

Proposition 6.2.7. Let G be a multigraph that has a proper 2-coloring. Then, G has exactly $2^{\operatorname{conn} G}$ many proper 2-colorings.

Proof. Each component of G has exactly 2 proper 2-colorings (as we can freely choose the color of some arbitrarily preselected vertex, but then our algorithm decides the colors of all the other vertices). □

6.3. The Brooks theorems

Here is a sufficient criterion for the existence of a proper coloring:

Theorem 6.3.1 (Little Brooks theorem)**.** Let $G = (V, E, \varphi)$ be a loopless multigraph with at least one vertex. Let

$$\alpha := \max \{\deg v \mid v \in V\}.$$

Then, G has a proper $(\alpha + 1)$-coloring.

Proof. Let $v_1, v_2, \ldots, v_n$ be the vertices of V, listed in some order (with no repetitions). We construct a proper $(\alpha + 1)$-coloring $f : V \to \{1, 2, \ldots, \alpha + 1\}$ of G as follows:

- First, we choose $f(v_1)$ arbitrarily.
- Then, we choose $f(v_2)$ to be distinct from the colors of all already-colored neighbors of v_2.

- Then, we choose $f(v_3)$ to be distinct from the colors of all already-colored neighbors of v_3.
- ...
- And so on, until all n values $f(v_1), f(v_2), \ldots, f(v_n)$ have been chosen.

I claim that we never run out of colors in this process. Indeed, at every step, we have $\alpha + 1$ colors to choose from, but only $\leq \alpha$ many already-colored neighbors, so there is at least one color that we can choose without clashing with a neighbor. So the algorithm works, and we get a proper $(\alpha + 1)$-coloring, so the theorem is proved. □

This is an example of a **greedy algorithm**: an algorithm which doesn't "think ahead" but only cares about the current step. Greedy algorithm don't work for all problems; for hard problems they will get you stuck. But the above is a case where it works.

As the name suggests, the Little Brooks theorem can be improved. You might wonder how, seeing that the $\alpha + 1$ is optimal in at least two cases:

- If $n \geq 2$, then the cycle graph C_n has maximum degree $\alpha = \max\{\deg v \mid v \in V\} = 2$. Thus, the theorem shows that it has a proper 3-colorings. When n is odd, it has no proper 2-coloring, so the $\alpha + 1$ cannot be improved in this case.
- If $n \geq 1$, then the complete graph K_n has maximum degree $\alpha = \max\{\deg v \mid v \in V\} = n - 1$, and thus (by the theorem) has a proper n-coloring. Again, this cannot be improved.

It turns out that these two examples are the only cases where $\alpha + 1$ cannot be improved, at least for a connected loopless multigraph! In all other cases, we can improve the $\alpha + 1$ to α:

Theorem 6.3.2 (Brooks theorem)**.** Let $G = (V, E, \varphi)$ be a connected loopless multigraph that is neither a complete graph nor an odd-length cycle. Let

$$\alpha := \max\{\deg v \mid v \in V\}.$$

Then, G has a proper α-coloring.

Proof. This is significantly harder to prove than the Little Brooks theorem. See the 2022 notes for a reference. □

6.4. The chromatic polynomial

Surprisingly, the number of proper k-colorings of a given graph G turns out to be a polynomial in k (with integer coefficients). More precisely:

Theorem 6.4.1 (Whitney's chromatic polynomial theorem)**.** Let $G = (V, E, \varphi)$ be a multigraph. Let χ_G be the polynomial in the single indeterminate x with coefficients in $\mathbb{Z}$ defined as follows:

$$\chi_G = \sum_{F \subseteq E} (-1)^{|F|} x^{\operatorname{conn}(V, F, \varphi|_F)} = \sum_{\substack{H \text{ is a spanning} \\ \text{subgraph of } G}} (-1)^{|\operatorname{E}(H)|} x^{\operatorname{conn} H}.$$

Then, for any $k \in \mathbb{N}$, we have

$$(\# \text{ of proper } k\text{-colorings of } G) = \chi_G(k).$$

This is completely useless for finding proper k-colorings, except for some small families of graphs. Yet the polynomial χ_G (called the **chromatic polynomial** of G) has many interesting properties and appears in various places. Let me sketch a proof of the theorem.

Again, we will use Iverson brackets: $[\mathcal{A}]$ is the truth value of $\mathcal{A}$. We will use the following fact:

Lemma 6.4.2 (cancellation lemma)**.** Let P be a finite set. Then,

$$\sum_{A \subseteq P} (-1)^{|A|} = [P = \varnothing].$$

Proof. If $P = \varnothing$, then the LHS is $(-1)^{|\varnothing|} = 1$, as is the RHS.

Now consider the case when $P \neq \varnothing$. Pick an arbitrary element $p \in P$. Now, the subsets A of P come in two forms: the ones that contain p, and the ones that don't. I claim that the corresponding addends in the sum $\sum_{A \subseteq P} (-1)^{|A|}$ cancel out: If A is a subset of P that contains p, then $A \setminus \{p\}$ is a subset that doesn't, and conversely, if A is a subset of P that doesn't contain p, then $A \cup \{p\}$ is a subset that does. Thus, we obtain a bijection

$$\{\text{subsets of } P \text{ that contain } p\} \to \{\text{subsets of } P \text{ that don't contain } p\},$$
$$A \mapsto A \setminus \{p\}$$

(with inverse map $A \mapsto A \cup \{p\}$). This bijection shows that

$$\sum_{\substack{A \subseteq P \\ \text{doesn't contain } p}} (-1)^{|A|} = \sum_{\substack{A \subseteq P \\ \text{contains } p}} \underbrace{(-1)^{|A \setminus \{p\}|}}_{\substack{=(-1)^{|A|-1} \\ =-(-1)^{|A|}}} = - \sum_{\substack{A \subseteq P \\ \text{contains } p}} (-1)^{|A|}.$$

Hence,

$$\sum_{\substack{A \subseteq P \\ \text{doesn't contain } p}} (-1)^{|A|} + \sum_{\substack{A \subseteq P \\ \text{contains } p}} (-1)^{|A|} = 0.$$

In other words,

$$\sum_{A \subseteq P} (-1)^{|A|} = 0.$$

But $[P = \varnothing]$ is also 0, so we are done with the lemma. □

Now, a notation:

Definition 6.4.3. Let $G = (V, E, \varphi)$ be a multigraph. Let $k \in \mathbb{N}$. Let $f : V \to \{1, 2, \ldots, k\}$ be a k-coloring. We then define a subset E_f of E by

$$E_f := \{e \in E \mid \text{the two endpoints of } e \text{ have the same color in } f\}.$$

The elements of E_f are called the ***f*-monochromatic** edges of G.

Proposition 6.4.4. Let $G = (V, E, \varphi)$ be a multigraph. Let $k \in \mathbb{N}$. Let $f : V \to \{1, 2, \ldots, k\}$ be a k-coloring. Then, the k-coloring f is proper if and only if $E_f = \varnothing$.

Proof. Obvious. □

Lemma 6.4.5. Let $G = (V, E, \varphi)$ be a multigraph. Let B be a subset of E. Let $k \in \mathbb{N}$. Then, the number of all k-colorings $f : V \to \{1, 2, \ldots, k\}$ satisfying $B \subseteq E_f$ is $k^{\operatorname{conn}(V, B, \varphi|_B)}$.

Proof. This is a sketch; see the 2022 notes (Lecture 22) for details.

Let $H = (V, B, \varphi \mid_B)$ be the spanning subgraph of G with edge set B. Then, a k-coloring $f : V \to \{1, 2, \ldots, k\}$ satisfying $B \subseteq E_f$ is the same as a k-coloring of H where any two adjacent vertices have equal colors. Clearly, such a k-coloring must be constant on each component of H. Since H has $\operatorname{conn} H$ many components, the number of such k-colorings is therefore $k^{\operatorname{conn} H} = k^{\operatorname{conn}(V, B, \varphi|_B)}$. □

Proof of Whitney's theorem. Fix $k \in \mathbb{N}$. We must prove that

$$(\text{\# of proper } k\text{-colorings of } G) = \chi_G(k).$$

Well: The definition of χ_G yields

$$\chi_G = \sum_{F \subseteq E} (-1)^{|F|} x^{\operatorname{conn}(V,F,\varphi|_F)} = \sum_{B \subseteq E} (-1)^{|B|} x^{\operatorname{conn}(V,B,\varphi|_B)}.$$

Hence,

$$\begin{aligned}
\chi_G(k) &= \sum_{B \subseteq E} (-1)^{|B|} \underbrace{k^{\operatorname{conn}(V,B,\varphi|_B)}}_{\substack{=(\text{\# of } k\text{-colorings } f \text{ of } G \text{ such that } B \subseteq E_f) \\ \text{(by the lemma)}}} \\
&= \sum_{B \subseteq E} (-1)^{|B|} \underbrace{(\text{\# of } k\text{-colorings } f \text{ of } G \text{ such that } B \subseteq E_f)}_{\substack{= \sum_{f:V \to \{1,2,\ldots,k\};\ B \subseteq E_f} 1 \\ \text{(because a sum of a bunch of 1's} \\ \text{is just the number of these 1's)}}} \\
&= \sum_{B \subseteq E} (-1)^{|B|} \sum_{\substack{f:V \to \{1,2,\ldots,k\}; \\ B \subseteq E_f}} 1 = \sum_{B \subseteq E} \sum_{\substack{f:V \to \{1,2,\ldots,k\}; \\ B \subseteq E_f}} (-1)^{|B|} \\
&= \sum_{f:V \to \{1,2,\ldots,k\}} \sum_{\substack{B \subseteq E; \\ B \subseteq E_f}} (-1)^{|B|} = \sum_{f:V \to \{1,2,\ldots,k\}} \underbrace{\sum_{B \subseteq E_f} (-1)^{|B|}}_{\substack{=[E_f = \varnothing] \\ \text{(by the cancellation lemma)}}} \\
&\qquad \left(\text{since any subset of } E_f \text{ is automatically a subset of } E\right) \\
&= \sum_{f:V \to \{1,2,\ldots,k\}} [E_f = \varnothing] \\
&= (\text{\# of } k\text{-colorings } f \text{ of } G \text{ such that } E_f = \varnothing) \\
&= (\text{\# of proper } k\text{-colorings } f \text{ of } G),
\end{aligned}$$

qed. □

Definition 6.4.6. The polynomial χ_G in the above theorem is known as the **chromatic polynomial** of G.

Here are the chromatic polynomials of some graphs:

Proposition 6.4.7. Let $n \geq 1$ be an integer. Then:

(a) For the path graph P_n with n vertices, we have

$$\chi_{P_n} = x(x-1)^{n-1}.$$

(b) More generally: For any tree T with n vertices, we have

$$\chi_T = x\,(x-1)^{n-1}.$$

(c) For the complete graph K_n with n vertices, we have

$$\chi_{K_n} = x\,(x-1)\,(x-2)\cdots(x-n+1).$$

(d) For the empty graph E_n with n vertices, we have

$$\chi_{E_n} = x^n.$$

(e) Assume that $n \geq 2$. For the cycle graph C_n with n vertices, we have

$$\chi_{C_n} = (x-1)^n + (-1)^n\,(x-1).$$

Proof. See 2022 notes or figure it out. □

6.5. Vizing's theorem

So far we have been coloring the vertices of a graph. What about coloring the edges instead?

Definition 6.5.1. Let $G = (V, E, \varphi)$ be a multigraph. Let $k \in \mathbb{N}$.

A k**-edge-coloring** of G means a map $f : E \to \{1, 2, \ldots, k\}$.

Such a k-edge-coloring f is said to be **proper** if no two distinct edges that have a common endpoint have the same color.

The most prominent fact about edge-colorings is the following theorem:

Theorem 6.5.2 (Vizing's theorem)**.** Let G be a simple graph with at least one vertex. Let

$$\alpha := \max\{\deg v \mid v \in V\}.$$

Then, G has a proper $(\alpha+1)$-edge-coloring.

Proof. See a reference in the notes. □

Note that G really needs to be a simple graph here.

Lecture 17

7. Independent sets

7.1. Definition and lower bound

Next, we define one of the most fundamental notions in graph theory:

Definition 7.1.1. An **independent set** of a multigraph G means a subset S of $\operatorname{V}(G)$ such that no two elements of S are adjacent.

In other words, an independent set of G means an induced subgraph of G that has no edges. Note that I didn't say "no two distinct elements"; I said "no two elements".

Thus, for example, what we called an "anti-triangle" in Lecture 1 is an independent set of size 3.

Remark 7.1.2. Independent sets are closely related to proper colorings.

Indeed, let G be a graph, and $k \in \mathbb{N}$. Let $f : V \to \{1, 2, \ldots, k\}$ be some k-coloring of G. For each $i \in \{1, 2, \ldots, k\}$, we set

$$V_i := \{\text{vertices having color } i\} = f^{-1}(i).$$

Then, the k-coloring f is proper if and only if the k sets $V_1, V_2, \ldots, V_k$ are independent sets of G.

One classical computational problem is to find a maximum-size independent set of a given graph. This problem is NP-hard, so don't expect a good algorithm. However, there are some lower bounds for this maximum size. Here is one:

Theorem 7.1.3. Let $G = (V, E, \varphi)$ be a loopless multigraph. Then, G has an independent set of size

$$\geq \sum_{v \in V} \frac{1}{1 + \deg v}.$$

We will give two proofs of this theorem, both illustrating important ideas.

First proof. Assume the contrary. Thus, each independent set S of G has size

$$|S| < \sum_{v \in V} \frac{1}{1 + \deg v}.$$

A V**-listing** shall mean a list of all vertices of V, with each vertex appearing exactly once. (So there are $|V|!$ many V-listings.)

If σ is a V-listing, then we define a subset J_σ of V by

$$J_\sigma := \{v \in V \mid v \text{ occurs } \textbf{before} \text{ all neighbors of } v \text{ in } \sigma\}.$$

This J_σ is an independent set of G (because if u and v are two adjacent vertices in J_σ, then u appears before v, but v appears before u, which is a contradiction). Thus, by our assumption,

$$|J_\sigma| < \sum_{v \in V} \frac{1}{1 + \deg v}.$$

This inequality holds for **every** V-listing σ. Thus, summing it over all σ, we obtain

$$\sum_{\sigma \text{ is a } V\text{-listing}} |J_\sigma| < \sum_{\sigma \text{ is a } V\text{-listing}} \sum_{v \in V} \frac{1}{1 + \deg v}$$
$$= (\# \text{ of all } V\text{-listings}) \cdot \sum_{v \in V} \frac{1}{1 + \deg v}.$$

On the other hand, I claim the following:

Claim 1: For each $v \in V$, we have

$$(\# \text{ of all } V\text{-listings } \sigma \text{ satisfying } v \in J_\sigma) \geq \frac{(\# \text{ of all } V\text{-listings})}{1 + \deg v}.$$

[*Proof of Claim 1:* Fix a vertex $v \in V$. Define $\deg' v$ to be the # of all neighbors of v. Clearly, $\deg' v \leq \deg v$.

We call a V-listing σ **good** if the vertex v occurs in it before all its neighbors (i.e., it $v \in J_\sigma$). Thus, we must prove that

$$(\# \text{ of all good } V\text{-listings } \sigma) \geq \frac{(\# \text{ of all } V\text{-listings})}{1 + \deg v}.$$

We shall actually show that

$$(\# \text{ of all good } V\text{-listings } \sigma) = \frac{(\# \text{ of all } V\text{-listings})}{1 + \deg' v}$$

(which suffices since $\deg' v \leq \deg v$). In other words, we shall prove that a (uniformly) random V-listing is good with probability $\dfrac{1}{1 + \deg' v}$.

Why is this the case? You can argue probabilistically, e.g., as follows: We consider only the set $M := \{v\} \cup N(v)$, where $N(v)$ is the set of all neighbors of v. If we restrict a V-listing to this set, then all orderings of this set will be equally likely. Among all the $|M|!$ many orderings of this set, the good ones (i.e., the ones that lead to good V-listings) are the ones that begin with v, and there are $(|M|-1)!$ many of those (since we only need to choose the order of the remaining $|M|-1$ many elements). Hence, the probability for a V-listing to be good is $\dfrac{(|M|-1)!}{|M|!} = \dfrac{1}{|M|} = \dfrac{1}{|\{v\} \cup N(v)|} = \dfrac{1}{1+\deg' v}$.

Alternatively, you can argue bijectively as follows: We define a map

$$\Gamma : \{\text{all } V\text{-listings}\} \to \{\text{all good } V\text{-listings}\}$$

as follows: Whenever τ is a V-listing, we let $\Gamma(\tau)$ be the V-listing obtained from τ by swapping v with the first neighbor of v that occurs in τ (or, if τ is already good, then we just do nothing, i.e., we set $\Gamma(\tau) = \tau$). This map Γ is a $(1+\deg' v)$-to-1 correspondence – i.e., for each good V-listing σ, there are exactly $1+\deg' v$ many V-listings τ that satisfy $\Gamma(\tau) = \sigma$ (indeed, one of these τ's is σ itself, and the others are obtained by swapping v with a neighbor of v). Hence, by the multijection principle, we conclude that

$$(\text{\# of all } V\text{-listings}) = (1+\deg' v) \cdot (\text{\# of all good } V\text{-listings}).$$

Hence,

$$(\text{\# of all good } V\text{-listings}) = \frac{(\text{\# of all } V\text{-listings})}{1+\deg' v}.$$

As we said, this completes the proof of Claim 1.]

Now, recall that

$$\sum_{\sigma \text{ is a } V\text{-listing}} |J_\sigma| < (\text{\# of all } V\text{-listings}) \cdot \sum_{v \in V} \frac{1}{1+\deg v}.$$

Hence,

$$\begin{aligned}
&(\text{\# of all } V\text{-listings}) \cdot \sum_{v\in V} \frac{1}{1+\deg v} \\
&> \sum_{\sigma \text{ is a } V\text{-listing}} \underbrace{|J_\sigma|}_{\substack{=\sum\limits_{v\in V}[v\in J_\sigma] \\ \text{(since the Iverson bracket } [v\in J_\sigma] \text{ is 1} \\ \text{for each } v\in J_\sigma\text{, and is 0 for all the other } v\text{'s)}}} \\
&= \sum_{\sigma \text{ is a } V\text{-listing}} \sum_{v\in V} [v \in J_\sigma] \\
&= \sum_{v\in V} \underbrace{\sum_{\sigma \text{ is a } V\text{-listing}} [v \in J_\sigma]}_{=(\text{\# of } V\text{-listings } \sigma \text{ such that } v\in J_\sigma)} \\
&= \sum_{v\in V} \underbrace{(\text{\# of } V\text{-listings } \sigma \text{ such that } v \in J_\sigma)}_{\geq \frac{(\text{\# of all } V\text{-listings})}{1+\deg v}} \\
&\geq \sum_{v\in V} \frac{(\text{\# of all } V\text{-listings})}{1+\deg v} = (\text{\# of all } V\text{-listings}) \cdot \sum_{v\in V} \frac{1}{1+\deg v}.
\end{aligned}$$

But this is absurd, since no $x \in \mathbb{R}$ satisfies $x > x$. Contradiction, and we're done. □

Remark 7.1.4. This proof is an example of a **probabilistic proof**. In fact, we have been manipulating sums, but we could easily replace these sums by averages. Claim 1 would then say that for a given vertex v, the **probability** that a (uniformly random) V-listing σ satisfies $v \in J_\sigma$ is $\geq \frac{1}{1+\deg v}$. Thus, the expectation of $|J_\sigma|$ is $\geq \sum_{v\in V} \frac{1}{1+\deg v}$. Therefore, at least one V-listing σ actually satisfies $|J_\sigma| \geq \sum_{v\in V} \frac{1}{1+\deg v}$. The proof does not tell you how to find σ; it just guarantees that such a σ exists and in fact this is true "on average". This does not mean that at least half the σ's satisfy $|J_\sigma| \geq \sum_{v\in V} \frac{1}{1+\deg v}$ (since the median is not the mean).

However, there is a second proof that actually lets you construct a good J in polynomial time. Moreover, this second proof is motivated by the first. Indeed, we try to gradually narrow down our space of choices (= the V-listings σ) in a strategically reasonable way (each time making sure that we are narrowing it down as little as we can). One such way is to begin by choosing $\sigma(1)$ to be a vertex $v \in V$ with minimum $\deg v$. Thus we get a recursive algorithm for computing σ:

Second proof. We proceed by strong induction on $|V|$. Thus, we fix $p \in \mathbb{N}$, and we assume that the theorem is proved for all loopless multigraphs G with $< p$ vertices. We shall now prove it for a loopless multigraph $G = (V, E, \varphi)$ with p vertices.

If $|V| = 0$, then this is clear. So we WLOG assume that $|V| \neq 0$. We also WLOG assume that G is a simple graph (otherwise, replace G by G^{simp}, and the bound only gets better).

Since $|V| \neq 0$, there exists a vertex $u \in V$ with $\deg_G u$ minimum (note: we write $\deg_G u$ for the degree of u in G, since we will soon have $\deg_{G'} u$ for another graph G'). Pick such a u. Thus,

$$\deg_G v \geq \deg_G u \qquad \text{for each } v \in V.$$

Let $U := \{u\} \cup N(u)$, where $N(u)$ is the set of all neighbors of u. Then, $U \subseteq V$ and $|U| = 1 + \deg_G u$ (an honest equality, since G is a simple graph).

Let G' be the induced subgraph of G on the set $V \setminus U$. This graph G' has fewer vertices than G, and thus (by the induction hypothesis) contains an independent set T of size

$$|T| \geq \sum_{v \in V \setminus U} \frac{1}{1 + \deg_{G'} v}.$$

Consider this T, and set $S := \{u\} \cup T$. Then, S is an independent set of G. Now I claim that $|S| \geq \sum\limits_{v \in V} \dfrac{1}{1 + \deg_G v}$. Indeed,

$$\begin{aligned}
\sum_{v \in V} \frac{1}{1 + \deg_G v} &= \sum_{v \in U} \underbrace{\frac{1}{1 + \deg_G v}}_{\substack{\leq \frac{1}{1 + \deg_G u} \\ \text{(since } \deg_G v \geq \deg_G u)}} + \sum_{v \in V \setminus U} \underbrace{\frac{1}{1 + \deg_G v}}_{\substack{\leq \frac{1}{1 + \deg_{G'} v} \\ \text{(since } \deg_G v \geq \deg_{G'} v)}} \\
&\leq \underbrace{\sum_{v \in U} \frac{1}{1 + \deg_G u}}_{\substack{= |U| \cdot \frac{1}{1 + \deg_G u} \\ =1 \\ \text{(since } |U| = 1 + \deg_G u)}} + \underbrace{\sum_{v \in V \setminus U} \frac{1}{1 + \deg_{G'} v}}_{\leq |T|} \\
&\leq 1 + |T| = |S|.
\end{aligned}$$

Thus we have found an independent set of G having size $\geq \sum\limits_{v \in V} \dfrac{1}{1 + \deg_G v}$ (namely, S). Thus, our theorem holds for G, and we are done. □

Remark 7.1.5. This second proof (unlike the first) gives a very efficient algorithm for finding an independent set of size $\geq \sum\limits_{v \in V} \dfrac{1}{1 + \deg_G v}$. The way we obtained it from the first proof is sometimes called the **method of conditional probabilities**.

7.2. A weaker lower bound

Let us now weaken the theorem a bit:

Corollary 7.2.1. Let G be a loopless multigraph with n vertices and m edges. Then, G has an independent set of size

$$\geq \frac{n^2}{n+2m}.$$

To prove this, we will need the following inequality:

Lemma 7.2.2. Let $a_1, a_2, \ldots, a_n$ be n positive reals. Then,

$$\frac{1}{a_1} + \frac{1}{a_2} + \cdots + \frac{1}{a_n} \geq \frac{n^2}{a_1 + a_2 + \cdots + a_n}.$$

Proof. Pick any of the following approaches:

- Apply Jensen's inequality to the convex function $x \mapsto \frac{1}{x}$ (on $\mathbb{R}^+$).
- Apply Cauchy–Schwarz to get

$$(a_1 + a_2 + \cdots + a_n)\left(\frac{1}{a_1} + \frac{1}{a_2} + \cdots + \frac{1}{a_n}\right)$$
$$\geq \left(\sqrt{a_1 \frac{1}{a_1}} + \sqrt{a_2 \frac{1}{a_2}} + \cdots + \sqrt{a_n \frac{1}{a_n}}\right)^2 = n^2.$$

- Apply the AM-HM inequality.
- Apply the AM-GM inequality twice.
- There is a direct proof using the inequality $\frac{u}{v} + \frac{v}{u} \geq 2$ for any positive

reals u and v. Indeed,

$$\begin{aligned}
&(a_1 + a_2 + \cdots + a_n)\left(\frac{1}{a_1} + \frac{1}{a_2} + \cdots + \frac{1}{a_n}\right) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} a_i \frac{1}{a_j} = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{a_i}{a_j} \\
&= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \underbrace{\left(\frac{a_i}{a_j} + \frac{a_j}{a_i}\right)}_{\geq 2} \qquad \text{(by symmetrization)} \\
&\geq \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} 2 = \sum_{i=1}^{n} \sum_{j=1}^{n} 1 = n^2.
\end{aligned}$$

□

Proof of the corollary. Write the multigraph G as $G = (V, E, \varphi)$. Then, $|V| = n$ and $|E| = m$. WLOG assume that $V = \{1, 2, \ldots, n\}$. Hence,

$$\sum_{v=1}^{n} \deg v = \sum_{v \in V} \deg v = 2 \cdot |E| = 2m.$$

However, the theorem yields that G has an independent set of size

$$\begin{aligned}
&\geq \sum_{v \in V} \frac{1}{1 + \deg v} = \sum_{v=1}^{n} \frac{1}{1 + \deg v} \\
&\geq \frac{n^2}{\sum_{v=1}^{n} (1 + \deg v)} \qquad \text{(by the lemma, applied to } a_v = 1 + \deg v\text{)} \\
&= \frac{n^2}{n + \sum_{v=1}^{n} \deg v} = \frac{n^2}{n + 2m},
\end{aligned}$$

qed. □

7.3. A proof of Turan's theorem

Recall Turan's theorem, which we can now easily prove:

> **Theorem 7.3.1** (Turan's theorem)**.** Let r be a positive integer. Let G be a simple graph with n vertices and e edges. Assume that
>
> $$e > \frac{r-1}{r} \cdot \frac{n^2}{2}.$$
>
> Then, there exist $r + 1$ distinct vertices of G that are mutually adjacent (i.e., any two of them are adjacent if they are distinct).

Proof. Write the simple graph G as $G = (V, E)$. Thus, $|V| = n$ and $|E| = e$ and $E \subseteq \mathcal{P}_2(V)$.

Let $E' := \mathcal{P}_2(V) \setminus E$. Thus, the set E' consists of all "non-edges" of G – that is, of all 2-element subsets of V that are not edges of G. Therefore, $|E'| = \binom{n}{2} - e$ (since $|\mathcal{P}_2(V)| = \binom{n}{2}$).

Now, let G' be the simple graph (V, E'). This graph G' is called the **complementary graph** of G; it has n vertices and $|E'| = \binom{n}{2} - e$ edges. Hence, the last corollary yields that it has an independent set of size

$$\geq \frac{n^2}{n + 2 \cdot \left(\binom{n}{2} - e\right)} = \frac{n^2}{n + n(n-1) - 2e} = \frac{n^2}{n^2 - 2e}.$$

Let S be this independent set. Then, $|S| \geq \frac{n^2}{n^2 - 2e} > r$ (this follows by high-school algebra from $e > \frac{r-1}{r} \cdot \frac{n^2}{2}$). So S has at least $r + 1$ many elements.

Now, the elements of S are vertices of G, and every two of them are adjacent in G (since they are non-adjacent in G' (because S is an independent set of G')). Thus, we have found $r + 1$ mutually adjacent vertices of G, qed. □

This is only one of many proofs of Turan's theorem. See Aigner/Ziegler *Proofs from the Book* for a few more proofs.

8. Matchings

8.1. Introduction

Independent sets of a graph are sets of vertices that "have no edges in common" (i.e., no two belong to the same edge).

In a sense, **matchings** are the dual notion: they are sets of edges that "have no vertices in common" (i.e., no two contain the same vertex). Here is the formal definition:

Definition 8.1.1. Let $G = (V, E, \varphi)$ be a loopless multigraph.

(a) A **matching** of G means a subset M of E such that no two distinct edges in M have a common endpoint.

(b) If M is a matching of G, then an M**-edge** means an edge in M.

(c) If M is a matching of G, and if $v \in V$ is any vertex, then we say that v is **matched** in M if v is an endpoint of an M-edge. In this case, this latter M-edge is necessarily unique, and will be called the **M-edge of** v. The other endpoint of this M-edge (i.e., the endpoint $\neq v$) is called the **M-partner** of v.

(d) A matching M of G is said to be **perfect** if each vertex of G is matched in M.

(e) Let A be a subset of V. A matching M of G is said to be **A-complete** if each vertex in A is matched in M.

Hence, a matching M of a multigraph $G = (V, E, \varphi)$ is perfect if and only if it is V-complete.

One algorithmic problem is to find a maximum-size matching of a given multigraph $G = (V, E, \varphi)$. This is far from trivial, since the greedy algorithm (adding new edges until this is no longer possible) does not work (in general, it gets stuck without getting a maximum-size matching). Nevertheless, there is a good algorithm of running time $O\left(|E| \cdot |V|^2\right)$, called the Edmonds blossom algorithm. (It is typically described in courses on combinatorial optimization, or advanced graph theory courses.)

We will consider a particular case of the matching problem: that of **bipartite matching** (i.e., finding matchings in bipartite graphs).

8.2. Bipartite graphs

Definition 8.2.1. A **bipartite graph** is a triple (G, X, Y), where

- $G = (V, E, \varphi)$ is a multigraph, and
- X and Y are two disjoint subsets of V such that $X \cup Y = V$ and such that each edge of G has one endpoint in X and one endpoint in Y.

Example 8.2.2. The 6-cycle graph C_6 can be made into a bipartite graph in two ways: Both triples

$$(C_6, \ \{1,3,5\}, \ \{2,4,6\}) \qquad \text{and}$$
$$(C_6, \ \{2,4,6\}, \ \{1,3,5\})$$

are bipartite graphs.

We tend to draw a bipartite graph (G, X, Y) in such a way that the vertices in X are aligned on a single vertical line, and the vertices in Y are also aligned on a single vertical line. The X-line should be left of the Y-line.

(See Spring 2022 Lecture 24 for examples.)

This example suggests the following terminology:

Definition 8.2.3. Let (G, X, Y) be a bipartite graph. We shall refer to the vertices in X as the **left vertices**, and to the vertices in Y as the **right vertices**. The edges of G will be called the **edges** of this bipartite graph (G, X, Y).

Thus, each edge of a bipartite graph joins a left vertex with a right vertex. Bipartite graphs are "the same as" multigraphs with a proper 2-coloring:

Proposition 8.2.4. Let $G = (V, E, \varphi)$ be a multigraph.

1. If (G, X, Y) is a bipartite graph, then the map

$$f : V \to \{1, 2\},$$
$$v \mapsto \begin{cases} 1, & \text{if } v \in X; \\ 2, & \text{if } v \in Y \end{cases}$$

is a proper 2-coloring of G.

2. Conversely, if $f : V \to \{1, 2\}$ is a proper 2-coloring of G, then

$$(G, \ \{\text{vertices of color } 1\}, \ \{\text{vertices of color } 2\})$$

is a bipartite graph.

3. These two operations are mutually inverse.

Proof. Follows from the definitions. □

Proposition 8.2.5. Let (G, X, Y) be a bipartite graph. Then, the graph G has no circuits of odd length. In particular, G has no loops or triangles.

Proof. Follows from the 2-coloring equivalence theorem. □

Definition 8.2.6. Let $G = (V, E, \varphi)$ be any multigraph. Let U be a subset of V. Then,

$$N(U) := \{v \in V \mid v \text{ has a neighbor in } U\}.$$

This is called the **neighbor set** of U.

For bipartite graphs, the neighbor set has a nice property:

Proposition 8.2.7. Let (G, X, Y) be a bipartite graph. Let $A \subseteq X$. Then,

$$N(A) \subseteq Y.$$

Proof. Follows from the definitions. □

The following is easy:

Proposition 8.2.8. Let (G, X, Y) be a bipartite graph. Assume that G has an X-complete matching. Then, each subset A of X satisfies $|N(A)| \geq |A|$.

But more interestingly, the converse of this proposition is also true:

Theorem 8.2.9 (Hall's marriage theorem, short: HMT)**.** Let (G, X, Y) be a bipartite graph. Assume that each subset A of X satisfies $|N(A)| \geq |A|$. Then, G has an X-complete matching.

We will discuss this in more detail next time.

Lecture 18

Recall:

Definition 8.2.10. Let $G = (V, E, \varphi)$ be a loopless multigraph.

(a) A **matching** of G means a subset M of E such that no two distinct edges in M have a common endpoint.

(b) If M is a matching of G, then an M**-edge** means an edge in M.

(c) If M is a matching of G, and if $v \in V$ is any vertex, then we say that v is **matched** in M if v is an endpoint of an M-edge. In this case, this latter M-edge is necessarily unique, and will be called the M**-edge of** v. The other endpoint of this M-edge (i.e., the endpoint $\neq v$) is called the M**-partner** of v.

(d) A matching M of G is said to be **perfect** if each vertex of G is matched in M.

(e) Let A be a subset of V. A matching M of G is said to be A**-complete** if each vertex in A is matched in M.

Definition 8.2.11. A **bipartite graph** is a triple (G, X, Y), where

- $G = (V, E, \varphi)$ is a multigraph, and
- X and Y are two disjoint subsets of V such that $X \cup Y = V$ and such that each edge of G has one endpoint in X and one endpoint in Y.

Definition 8.2.12. Let (G, X, Y) be a bipartite graph. We shall refer to the vertices in X as the **left vertices**, and to the vertices in Y as the **right vertices**. The edges of G will be called the **edges** of this bipartite graph (G, X, Y).

Definition 8.2.13. Let $G = (V, E, \varphi)$ be any multigraph. Let U be a subset of V. Then,

$$N(U) := \{v \in V \mid v \text{ has a neighbor in } U\}.$$

This is called the **neighbor set** of U.

For bipartite graphs, the neighbor set has a nice property:

Proposition 8.2.14. Let (G, X, Y) be a bipartite graph. Let $A \subseteq X$. Then,

$$N(A) \subseteq Y.$$

8.3. Hall's marriage theorem

Let us begin with some very elementary and simple facts.

Proposition 8.3.1. Let (G, X, Y) be a bipartite graph. Let M be a matching of G. Then:

1. The M-partner of a vertex $x \in X$ belongs to Y (if it exists).
   The M-partner of a vertex $y \in Y$ belongs to X (if it exists).
2. We have $|M| \leq |X|$ and $|M| \leq |Y|$.
3. If M is X-complete, then $|X| \leq |Y|$.
4. If M is perfect, then $|X| = |Y|$.

Proof. Easy. See Spring 2022 Lecture 24 proof of Proposition 1.3.1. □

Proposition 8.3.2. Let (G, X, Y) be a bipartite graph. Let A be a subset of X. Assume that G has an X-complete matching. Then, $|N(A)| \geq |A|$.

Proof. Let V be the vertex set of G. Let M be an X-complete matching of G (we assumed that such an M exists). The map

$$\begin{aligned} \mathbf{p} : X &\to V, \\ x &\mapsto (\text{the } M\text{-partner of } x) \end{aligned}$$

is injective (since two distinct edges in M cannot have a common endpoint). Therefore, $|\mathbf{p}(A)| = |A|$. But $\mathbf{p}(A) \subseteq N(A)$ (since an M-partner of some $x \in A$ will always lie in $N(A)$), so that $|\mathbf{p}(A)| \leq |N(A)|$. Thus, $|N(A)| \geq |\mathbf{p}(A)| = |A|$. □

So this proposition shows us a necessary condition for the existence of an X-complete matching in a bipartite graph (G, X, Y): Namely, the condition says that every $A \subseteq X$ satisfies $|N(A)| \geq |A|$.

Hall's marriage theorem (short: **HMT**) claims that this condition is also sufficient:

Theorem 8.3.3 (Hall's marriage theorem, aka HMT)**.** Let (G, X, Y) be a bipartite graph. Assume that each subset A of X satisfies $|N(A)| \geq |A|$. (This assumption is called the **Hall condition**.)

Then, G has an X-complete matching.

This theorem was originally found by Philip Hall in 1935 to solve a group theory problem and simultaneously by Wilhelm Maak for use in analysis. Nowadays, most of its uses are in combinatorics.

I will not prove it today, but next time I will prove it using the theory of **network flows**; there are many other proofs, including some very elementary ones. Network flows also give an efficient algorithm for finding a maximum matching in a bipartite graph.

But before that, let me talk about variants and applications of the HMT.

8.4. König and Hall–König

Hall's marriage theorem is famous for its many forms and versions, which are "secretly" equivalent to it. We will start with one that is known as **Kőnig's theorem** (discovered by Dénes Kőnig and Jenő Egerváry in 1931). This relies on the notion of a **vertex cover**:

Definition 8.4.1. Let $G = (V, E, \varphi)$ be a multigraph. A **vertex cover** of G means a subset C of V such that each edge of G contains at least one vertex in C.

Remark 8.4.2. Each vertex cover of a multigraph G is a dominating set (as long as G has no degree-0 vertices). But the converse is not true.

Proposition 8.4.3. Let G be a loopless multigraph.
Let m be the largest size of a matching in G.
Let c be the smallest size of a vertex cover in G.
Then, $m \leq c$.

Proof. Consider a matching M of size m and a vertex cover C of size c (these exist by assumption). Now, we build a map $f : M \to C$ by letting $f(m)$ be an endpoint of m that belongs to C (this always exists, since C is a vertex cover; if there are two such endpoints, just choose one). This map f is injective, since M is a matching. So we get $|M| \leq |C|$, that is, $m \leq c$. □

In general, the inequality can be strict: We can have $m < c$. (For example, for $G = C_3$, we have $m = 1$ and $c = 2$.) König's result is that $m = c$ holds at least when G is bipartite:

Theorem 8.4.4 (König's theorem)**.** Let (G, X, Y) be a bipartite graph.
Let m be the largest size of a matching in G.
Let c be the smallest size of a vertex cover in G.
Then, $m = c$.

Both Hall's and König's theorems follow easily from the following theorem:

Theorem 8.4.5 (Hall–König matching theorem, aka HKMT)**.** Let (G, X, Y) be a bipartite graph. Then, there exist a matching M of G and a subset U of X such that

$$|M| \geq |N(U)| + |X| - |U| .$$

Why?

Proof of Hall's marriage using the HKMT. The HKMT yields that there exist a matching M of G and a subset U of X such that

$$|M| \geq |N(U)| + |X| - |U|.$$

Consider these M and U. By the Hall condition, we have $|N(U)| \geq |U|$. Hence,

$$|M| \geq |N(U)| + |X| - |U| \geq |U| + |X| - |U| = |X|.$$

This easily yields that M is X-complete (since otherwise, the M-edges would have fewer than $|X|$ many X-vertices going around them, but that would violate the distinctness of their endpoints). So G has an X-complete matching, qed. □

Proof of König's theorem using the HKMT. The HKMT yields that there exist a matching M of G and a subset U of X such that

$$|M| \geq |N(U)| + |X| - |U|.$$

Consider these M and U. Set $C := (X \setminus U) \cup N(U)$. Then, $|C| = |N(U)| + |X| - |U|$ (since $X \setminus U$ is disjoint from $N(U)$). Furthermore, C is a vertex cover of G (why?). Hence, $|C| \geq c$. But M is a matching, so $|M| \leq m$. Hence,

$$m \geq |M| \geq |N(U)| + |X| - |U| = |C| \geq c.$$

Combining this with $m \leq c$ (which we know from a proposition above), we obtain $m = c$. This proves König's theorem. □

So it will suffice to prove the HKMT.

Incidentally, the HKMT is not significantly stronger than either the HMT or König. You can (with a bit of work) derive it from either of the two. (For details, see the end of Lecture 24 in Spring 2022).

We shall now see theorems that are consequences of the HMT (often equivalent to it) but look rather different, in particular not always coming from graph theory.

8.5. Systems of representatives

The following equivalent form of the HMT doesn't look like a graph theory result at all:

Theorem 8.5.1 (existence of SDR)**.** Let $A_1, A_2, \ldots, A_n$ be any n sets. Assume that the union of any p of these sets has size $\geq p$, for each $p \in \{0, 1, \ldots, n\}$. (In other words, assume that

$$\left|A_{i_1} \cup A_{i_2} \cup \cdots \cup A_{i_p}\right| \geq p \qquad \text{for any } 1 \leq i_1 < i_2 < \cdots < i_p \leq n.$$

)

Then, we can find n **distinct** elements

$$a_1 \in A_1, \qquad a_2 \in A_2, \qquad \ldots, \qquad a_n \in A_n.$$

Proof. First, we WLOG assume that $A_1, A_2, \ldots, A_n$ are finite (otherwise, replace the infinite ones among them by some arbitrarily chosen n-element subsets).

Also, WLOG assume that no integer belongs to any of $A_1, A_2, \ldots, A_n$ (otherwise, rename the respective elements).

Now, let $X = \{1, 2, \ldots, n\}$ and $Y = A_1 \cup A_2 \cup \cdots \cup A_n$. Then, X and Y are disjoint finite sets.

We define a simple graph G as follows:

- The vertices are the elements of $X \cup Y$.
- A vertex $x \in X$ is adjacent to a vertex $y \in Y$ if and only if $y \in A_x$.

Thus, (G, X, Y) is a bipartite graph. The assumption

$$\left|A_{i_1} \cup A_{i_2} \cup \cdots \cup A_{i_p}\right| \geq p \qquad \text{for any } 1 \leq i_1 < i_2 < \cdots < i_p \leq n$$

ensures that it satisfies the Hall condition. Therefore, by the HMT (Hall's marriage theorem), we conclude that G has an X-complete matching. This matching must have the form

$$\{\{1, a_1\}, \{2, a_2\}, \ldots, \{n, a_n\}\}$$

where $a_1, a_2, \ldots, a_n$ are distinct. Thus, $a_1 \in A_1$ and $a_2 \in A_2$ and $\cdots$ and $a_n \in A_n$, so we are done. □

There is also a different set-theoretical restatement of the HMT, called the existence of an **SCR** (system of common representatives).

8.6. Regular bipartite graphs

The HMT gives a necessary and sufficient condition for the existence of an X-complete matching in a bipartite graph. In the more restrictive setting of **regular bipartite graphs** – i.e., bipartite graphs for which every vertex has the same degree –, this can be greatly simplified: Such a matching always exists! We shall soon prove this, but first let us get the definitions in order:

Definition 8.6.1. Let $k \in \mathbb{N}$. A multigraph G is said to be k**-regular** if all its vertices have degree k.

For example:

- A graph is 1-regular if and only if it is a disjoint union of copies of P_2.
- A graph is 2-regular if and only if it is a disjoint union of copies of C_n for various n (allowing $n = 1$ and $n = 2$ in particular). Nice and reasonably easy exercise.
- The 3-regular graphs are called **cubic graphs** and cannot really be classified.

Example 8.6.2. Any Kneser graph $K_{S,k}$ is $\dbinom{|S| - k}{k}$-regular.

Proof. Exercise. □

Proposition 8.6.3. Let $k > 0$. Let (G, X, Y) be a k-regular bipartite graph (i.e., a bipartite graph such that G is k-regular). Then, $|X| = |Y|$.

Proof. Each edge of G has exactly one endpoint in X. Thus,

$$|\operatorname{E}(G)| = \sum_{x \in X} \deg x = \sum_{x \in X} k = k \cdot |X| .$$

Similarly, $|\operatorname{E}(G)| = k \cdot |Y|$. Comparing these, we get $k \cdot |X| = k \cdot |Y|$. Dividing by k, we obtain $|X| = |Y|$ (since $k > 0$). □

Theorem 8.6.4 (Frobenius matching theorem)**.** Let $k > 0$. Let (G, X, Y) be a k-regular bipartite graph (i.e., a bipartite graph such that G is k-regular). Then, G has a perfect matching.

Proof. First, we claim that each subset A of X satisfies $|N(A)| \geq |A|$.

Indeed, let A be a subset of X. Consider the edges of G that have at least one endpoint in A. We shall call such edges "A-edges". How many are there?

On the one hand, each A-edge contains exactly one vertex in A (since $A \subseteq X$). Hence,

$$(\text{\# of } A\text{-edges}) = \sum_{x \in A} \deg x = \sum_{x \in A} k = k \cdot |A| .$$

On the other hand, each A-edge contains exactly one vertex in $N(A)$ (since $N(A) \subseteq Y$). Hence,

$$(\text{\# of } A\text{-edges}) \leq \sum_{y \in N(A)} \deg y = \sum_{y \in N(A)} k = k \cdot |N(A)| .$$

Hence,

$$k \cdot |A| = (\text{\# of } A\text{-edges}) \leq k \cdot |N(A)| .$$

Since $k > 0$, we can divide by k and conclude that $|A| \leq |N(A)|$, that is, $|N(A)| \geq |A|$.

So we have proved that the Hall condition is satisfied. Hence, the HMT yields that G has an X-complete matching. Let M be this matching.

I claim that M is perfect. Why? Recall that $|X| = |Y|$ by the previous proposition. Since M is X-complete, we have $|M| = |X| = |Y|$. Therefore, M must use every vertex in Y, and hence is Y-complete as well. Thus, M is perfect, qed. □

8.7. Latin squares

An application of the Frobenius matching theorem is the study of Latin squares:

Definition 8.7.1. Let $n \in \mathbb{N}$. A **Latin square** of order n is an $n \times n$-matrix M such that:

1. The entries of M are the numbers $1, 2, \ldots, n$, each appearing exactly n times.
2. In each row of M, the entries are distinct.
3. In each column of M, the entries are distinct.

(Add one more condition and you get the notion of a Sudoku!)

Example 8.7.2. Here is a Latin square of order 5:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 1 \\ 3 & 4 & 5 & 1 & 2 \\ 4 & 5 & 1 & 2 & 3 \\ 5 & 1 & 2 & 3 & 4 \end{pmatrix}.$$

Similarly, for each $n \in \mathbb{N}$, the matrix $\left(c_{i+j-1}\right)_{1\leq i\leq n,\ 1\leq j\leq n}$ with

$$c_k = \begin{cases} k, & \text{if } k \leq n; \\ k-n, & \text{if } k > n \end{cases}$$

is a Latin square of order n.

This example (along with its many permutations) shows that Latin squares of each order exist, but it doesn't even begin to classify them. There are many more. What would be a good algorithm to generate general Latin squares?

We can try this: Build a Latin square row by row, starting with the top row. Each next row needs to consist of distinct numbers, and each of its entries needs to differ from all the entries above it (in the same column). You should expect to sometimes get stuck (i.e., being unable to add the next row).

Surprisingly, you never get stuck (until you have all n rows). For example:

$$\begin{pmatrix} 2 & 4 & 1 & 5 & 3 \\ 1 & 3 & 4 & 2 & 5 \\ 5 & 2 & 3 & 4 & 1 \\ 3 & 5 & 2 & 1 & 4 \\ 4 & 1 & 5 & 3 & 2 \end{pmatrix}.$$

Why does this always work? And more importantly, how do we actually choose each new row, short of trying all $n!$ permutations?

Proposition 8.7.3. Let $n \in \mathbb{N}$ and $k \in \{0, 1, \ldots, n-1\}$. Then, any $k \times n$ **Latin rectangle** (i.e., any $k \times n$-matrix that contains the entries $1, 2, \ldots, n$, each exactly k times, and satisfies the conditions 2 and 3 from the definition of a Latin square) can be extended to a $(k+1) \times n$ Latin rectangle by inserting an appropriately chosen new row at the bottom.

Proof. Let M be a $k \times n$ Latin rectangle. We want to find a new row that we can append to M at the bottom, so that

- all its entries are distinct, and
- each of its entries differs from all the entries above it.

Let $X = \{1, 2, \ldots, n\}$ and $Y = \{-1, -2, \ldots, -n\}$. Let G be the simple graph with vertex set $X \cup Y$, where a vertex $i \in X$ is adjacent to a vertex $-j \in Y$ if and only if the number j does not appear in the i-th column of M. Thus, we are looking for an X-complete matching in G. (If we find such a matching

$$\{\{1, -a_1\}, \{2, -a_2\}, \ldots, \{n, -a_n\}\},$$

then we can append the row $(a_1, a_2, \ldots, a_n)$ to our Latin rectangle.)

To find such an X-complete matching, we apply the Frobenius theorem (with $n-k$ instead of k). To do so, we need to show that G is $(n-k)$-regular. This is because:

- Each column is missing exactly $n-k$ elements. So each vertex in X has degree k.
- Each element of $\{1, 2, \ldots, n\}$ appears in exactly k columns. So each vertex in Y has degree k.

Thus, the Frobenius theorem applies (we are using $n-k > 0$ here), and we are done. □

Next time, we will actually prove the HKMT (and thus get all the other theorems today as corollaries).

Lecture 19

9. Network flows

Today I will give an introduction to **network flows** and their optimization. Books like Ford/Fulkerson (see 2022 Lecture 26 for precise references) say much more and (in particular) talk about applications. We will just prove the most basic theorem and use it to prove the Hall-König matching theorem and another theorem called the Menger theorem (see 2022 Lecture 27 for much more about that).

9.1. Definition

Recall that $\mathbb{N} = \{0, 1, 2, \ldots\}$.

Definition 9.1.1. A **network** consists of

- a multidigraph $D = (V, A, \psi)$;
- two distinct vertices $s, t \in V$ called the **source** and the **sink**, respectively;
- a function $c : A \to \mathbb{N}$, called the **capacity function**.

For an example, see the picture on the blackboard. (The label on an arc a is the number $c(a)$.)

Remark 9.1.2. I do not require that $\deg^- s = 0$ and $\deg^+ t = 0$, but this is often satisfied in applications.

Also, $c(a)$ can be 0, but in practice usually isn't.

Definition 9.1.3. Let N be a network consisting of a multidigraph $D = (V, A, \psi)$, a source $s \in V$, a sink $t \in V$ and a capacity function $c : A \to \mathbb{N}$. Then:

1. For any arc $a \in A$, the number $c(a)$ is called the **capacity** of a.
2. For any subset S of V, we let $\overline{S}$ denote the subset $V \setminus S$ of V.
3. If P and Q are two subsets of V, then $[P, Q]$ shall mean the set of all arcs of D whose source belongs to P and whose target belongs to Q. Thus,
$$[P, Q] := \{a \in A \mid \psi(a) \in P \times Q\}.$$

4. If P and Q are two subsets of V, and if $d : A \to \mathbb{N}$ is any function, then we set
$$d(P,Q) := \sum_{a \in [P,Q]} d(a) \in \mathbb{N}.$$
(In particular, we can apply this to $d = c$, thus getting $c(P,Q) = \sum_{a \in [P,Q]} c(a)$.)

Example 9.1.4. In the example on the blackboard, let $P = \{s, u\}$ and $Q = \overline{P} = \{v, w, x, y, t\}$. Thus,
$$[P,Q] = \{uv,\ ux,\ sw\}.$$
Hence,
$$c(P,Q) = c(uv) + c(ux) + c(sw) = 2 + 1 + 2 = 5.$$

Definition 9.1.5. Let N be a network consisting of a multidigraph $D = (V, A, \psi)$, a source $s \in V$, a sink $t \in V$ and a capacity function $c : A \to \mathbb{N}$.

A **flow** (on N) means a function $f : A \to \mathbb{N}$ with the following properties:

- We have $0 \leq f(a) \leq c(a)$ for each arc $a \in A$. This condition is called the **capacity constraints**.
- For any vertex $v \in V \setminus \{s, t\}$, we have
$$f^-(v) = f^+(v),$$
where
$$f^-(v) := \underbrace{\sum_{\substack{a \in A \text{ is an arc} \\ \text{with target } v}} f(a)}_{\text{inflow into } v} \qquad \text{and} \qquad f^+(v) := \underbrace{\sum_{\substack{a \in A \text{ is an arc} \\ \text{with source } v}} f(a)}_{\text{outflow from } v}.$$
This condition is called the **conservation constraints**.

If $f : A \to \mathbb{N}$ is a flow and $a \in A$ is an arc, then the nonnegative integer $f(a)$ is called the **arc flow** of f on a.

You can think of a flow as ...

- ... traffic along one-way roads ($c(a)$ is the capacity of road a, while $f(a)$ is the hourly traffic on road a);
- ... water (or oil) flowing through pipes ($c(a)$ is the capacity of pipe a, while $f(a)$ is the actual flow);

- ... money getting transferred between bank accounts (going from s to t, with all the other vertices acting as middlemen).

Real-life applications are somewhat different.

Remark 9.1.6. Flows on a network N can be viewed as a generalization of paths on the underlying digraph D. Indeed, if $\mathbf{p}$ is a path from s to t on the digraph $D = (V, A, \psi)$ that underlies a network N, then we can define a flow $f_{\mathbf{p}}$ on N as follows:

$$f_{\mathbf{p}}(a) = \begin{cases} 1, & \text{if } a \text{ is an arc of } \mathbf{p}; \\ 0, & \text{if not} \end{cases} \qquad \text{for each } a \in A,$$

provided that all arcs of $\mathbf{p}$ have capacity ≥ 1.

Definition 9.1.7. Let N be a network consisting of a multidigraph $D = (V, A, \psi)$, a source $s \in V$, a sink $t \in V$ and a capacity function $c : A \to \mathbb{N}$.

Let $f : A \to \mathbb{N}$ be an arbitrary map (e.g., a flow on N). Then:

1. For each vertex $v \in V$, we set

$$f^-(v) := \underbrace{\sum_{\substack{a \in A \text{ is an arc} \\ \text{with target } v}} f(a)}_{\substack{\text{called the} \\ \textbf{inflow} \text{ into } v}} \qquad \text{and} \qquad f^+(v) := \underbrace{\sum_{\substack{a \in A \text{ is an arc} \\ \text{with source } v}} f(a)}_{\substack{\text{called the} \\ \textbf{outflow} \text{ from } v}}.$$

2. We define the **value** of the map f to be the number $f^+(s) - f^-(s)$. This value is called $|f|$.

Now we can state an important optimization problem, known as the **maximum flow problem**: Given a network N, find a flow of maximum value.

Example 9.1.8. For any network N, we can define the **zero flow** on N. This is the flow $0_A : A \to \mathbb{N}$ that sends each arc $a \in A$ to 0. This flow has value $|0_A| = 0$.

Example 9.1.9. Finding a maximum matching in a bipartite graph is a particular case of the maximum flow problem.

Indeed, let (G, X, Y) be a bipartite graph. Then, we can transform it into a network as follows:

- Add two new vertices s and t.

- Turn each edge e of G into an arc $\overrightarrow{e}$ whose source is the X-endpoint of e and whose target is the Y-endpoint of e.
- Add an arc from s to each $x \in X$.
- Add an arc from each $y \in Y$ to t.
- Assign to each arc the capacity 1.

Now, the flows on this network N are in bijection with the matchings of G. Namely, if f is a flow on N, then the set

$$\left\{e \in \operatorname{E}(G) \mid f\left(\overrightarrow{e}\right) = 1\right\}$$

is a matching of G. Conversely, if M is a matching of G, then we obtain a flow f on N by assigning the arc flow 1 to all arcs of the form $\overrightarrow{e}$ where $e \in M$, as well as assigning the arc flow 1 to every new arc that joins s or t to a vertex matched in M. All other arcs are assigned the arc flow 0.

If a flow f corresponds to a matching M under this bijection, then $|f| = |M|$. Thus, finding a maximum matching is equivalent to finding a maximum-value flow.

9.2. Some basic properties of flows

Before we solve the maximum flow problem, let us prove some basic facts about flows:

Proposition 9.2.1. Let N be a network consisting of a multidigraph $D = (V, A, \psi)$, a source $s \in V$, a sink $t \in V$ and a capacity function $c : A \to \mathbb{N}$.

Let $f : A \to \mathbb{N}$ be a flow on N. Then,

$$\begin{aligned} |f| &= f^+(s) - f^-(s) \\ &= f^-(t) - f^+(t). \end{aligned}$$

Proof. We have

$$\sum_{v \in V} f^+(v) = \sum_{v \in V} f^-(v),$$

since both sides are just $\sum_{a \in A} f(a)$. However, the conservation constraints yield that

$$\sum_{v \in V \setminus \{s,t\}} f^+(v) = \sum_{v \in V \setminus \{s,t\}} f^-(v) \qquad \text{(term by term)}.$$

Subtracting this equality from the previous one, you obtain

$$f^+(s) + f^+(t) = f^-(s) + f^-(t).$$

In other words,

$$f^{+}(s) - f^{-}(s) = f^{-}(t) - f^{+}(t).$$

But the LHS here is $|f|$. So the RHS is $|f|$ as well. □

Proposition 9.2.2. Let N be a network consisting of a multidigraph $D = (V, A, \psi)$, a source $s \in V$, a sink $t \in V$ and a capacity function $c : A \to \mathbb{N}$.

Let $f : A \to \mathbb{N}$ be a flow on N. Let S be a subset of V. Then:

1. We have
$$f\left(S, \overline{S}\right) - f\left(\overline{S}, S\right) = \sum_{v \in S} \left(f^{+}(v) - f^{-}(v)\right).$$

2. Assume that $s \in S$ and $t \notin S$. Then,
$$|f| = f\left(S, \overline{S}\right) - f\left(\overline{S}, S\right).$$

3. Assume that $s \in S$ and $t \notin S$. Then,
$$|f| \leq c\left(S, \overline{S}\right).$$

4. Assume that $s \in S$ and $t \notin S$. Then, $|f| = c\left(S, \overline{S}\right)$ if and only if
$$\left(f(a) = 0 \text{ for all } a \in \left[\overline{S}, S\right]\right) \qquad \text{and}$$
$$\left(f(a) = c(a) \text{ for all } a \in \left[S, \overline{S}\right]\right).$$

Proof. See 2022 Lecture 26 Proposition 1.2.2. □

9.3. The max-flow-min-cut theorem

One more definition, before we meet the hero of today's story:

Definition 9.3.1. Let N be a network consisting of a multidigraph $D = (V, A, \psi)$, a source $s \in V$, a sink $t \in V$ and a capacity function $c : A \to \mathbb{N}$. Then:

1. A **cut** of N means a subset of A that has the form $\left[S, \overline{S}\right]$, where S is a subset of V satisfying $s \in S$ and $t \notin S$.

2. The **capacity** of a cut $\left[S, \overline{S}\right]$ is defined to be $c\left(S, \overline{S}\right) = \sum_{a \in \left[S, \overline{S}\right]} c(a) \in \mathbb{N}$.

Now, part of the previous proposition says that the value of any flow f can never be larger than the capacity of any cut $\left[S, \overline{S}\right]$. In particular, the maximum value of a flow is $\leq$ to the minimum capacity of a cut.

It turns out that this inequality actually is an equality! And moreover, it is not hard to compute both a maximum-value flow and a minimum-capacity cut:

Theorem 9.3.2 (max-flow-min-cut theorem)**.** Let N be a network consisting of a multidigraph $D = (V, A, \psi)$, a source $s \in V$, a sink $t \in V$ and a capacity function $c : A \to \mathbb{N}$. Then,

$$\max \{|f| \mid f \text{ is a flow}\} = \min \left\{ c\left(S, \overline{S}\right) \mid S \subseteq V \text{ and } s \in S \text{ and } t \notin S \right\}.$$

In other words, the maximum value of a flow is the minimum capacity of a cut.

Moreover, the **Ford-Fulkerson algorithm** lets you compute both flow and cut.

Proof idea. (See Spring 2022 Lecture 26 for details.)

The idea is to start with the zero flow 0_A and gradually improve it, increasing its value. Once we can no longer do this, we will have a flow of maximum value and also find a cut whose capacity equals this value.

How to improve a flow f ? The simplest way is to find a path from s to t that is not used to its full capacity (more precisely, each arc of this path has at least 1 unit of unused capacity in f). Then, we can increase $f(a)$ by 1 for each arc a of this path, and we obtain a better flow (i.e., a flow of higher value).

But such a path doesn't always exist, even if f is not maximum-value. Thus, we have to make our method subtler. Namely, we pick a "zig-zag path" (i.e., a path that can use arcs both forward and backward) from s to t with the property that every forward arc a has at least 1 unit of unused capacity (i.e., satisfies $f(a) < c(a)$) and every backward arc a has at least 1 unit of arc flow (i.e., satisfies $f(a) > 0$). Then, we increase $f(a)$ by 1 for every forward arc a of this path, and decrease $f(a)$ by 1 for every backward arc a of this path. As a result, our flow remains a flow (not hard to check), but has higher value (higher by 1), thus is an improvement.

Two questions:

1. How do we find such a "zig-zag path"?

2. What if there is none?

The answers to these questions are related. Namely, zig-zag paths are just regular paths in a different digraph: the **residual digraph** D_f. This residual digraph D_f has the same vertices as D, but its arcs are:

- the arcs $a \in A$ that satisfy $f(a) < c(a)$, as well as

- the reversals of all arcs $a \in A$ that satisfy $f(a) > 0$.

The paths of D_f are precisely the "zig-zag paths" of D that satisfy the required capacity condition (i.e., that satisfy $f(a) < c(a)$ for each forward arc a and $f(a) > 0$ for each backward arc a). Thus, we can answer question 1 since we have an algorithm for finding a path in a digraph.

As to question 2: What if there is no zig-zag path? I.e., what if D_f has no path from s to t ? In that case, we let

$$S = \left\{v \in V \mid D_f \text{ has a path from } s \text{ to } v\right\}.$$

Then, $S \subseteq V$ and $s \in S$ and $t \notin S$. Hence, $\left[S, \overline{S}\right]$ is a cut of N. Now, it is not hard to see that $f\left(S, \overline{S}\right) = c\left(S, \overline{S}\right)$ (since $f(a) = c(a)$ for each $a \in \left[S, \overline{S}\right]$) and $f\left(\overline{S}, S\right) = 0$ (since $f(a) = 0$ for each $a \in \left[\overline{S}, S\right]$). Thus, using the previous proposition, we get $c\left(S, \overline{S}\right) = |f|$. However, every flow on N has value $\leq c\left(S, \overline{S}\right)$ (again by the previous proposition). Thus, $c\left(S, \overline{S}\right) = |f|$ shows that the flow f has maximum possible value. Likewise, it shows that the cut $\left[S, \overline{S}\right]$ has minimum possible capacity. Thus, if there is no zig-zag path, then we are done.

So our algorithm to find a maximum-value flow is as follows:

- Start with the zero flow 0_A.
- Look for a zig-zag path. If there is one, then use it to improve the flow.
- Repeat until there is no longer a zig-zag path available. At that point, your flow is maximum-value.

This is called the **Ford-Fulkerson algorithm**. (Note that this will always terminate, since every step increases the value of the flow by at least 1, but the value of the flow is bounded from above.) □

Remark 9.3.3. The max-flow-min-cut theorem works also if we replace $\mathbb{N}$ by $\mathbb{Q}_+$ or $\mathbb{R}_+$, but the proof gets trickier. For $\mathbb{Q}_+$, the algorithm still works, because essentially it is the same as $\mathbb{N}$ up to common denominator. For $\mathbb{R}_+$, the algorithm might run in an endless loop without getting close to the maximum value. There is a way to fix it by choosing not some random zig-zag path, but a shortest possible zig-zag path at every step (provided, of course, that you modify the algorithm to increase the arc flows not by 1 but by the appropriate largest possible amounts). This is known as the **Edmonds-Karp version of the Ford-Fulkerson algorithm**.

See Lecture 26 for:

- a proof of the Hall-König matching theorem using the max-flow-min-cut theorem (and thus of Hall's marriage theorem).

See Lecture 27 for:

- Menger's theorems.