Last time, we started counting trees with n vertices. Out of 2 possible interpretations, ~~we chose~~ I chose the easier one (vertices labelled $1, 2, \ldots, n$) & suggested that the answer is n^{n-2}. Let's now state it rigorously & sketch a proof.

Def. Let $n \in \mathbb{N}$. Then, an <u>n-tree</u> means a simple graph (V, E) such that $V = [n]$ and the multigraph $(V, E, \text{"id"})$ (where "id" sends $e \mapsto e$) is a tree.

(NB: We use simple graphs in order not to count $\boxed{1 \overset{a}{—} 2}$ and $\boxed{1 \overset{b}{—} 2}$ as different.)

Remark: The graph $(\varnothing, \varnothing)$ is not connected and thus not a tree.

So there are no 0-trees.

There is exactly one 1-tree: $\boxed{1}$,

exactly one 2-tree: $\boxed{1 — 2}$.

There are 3 3-trees: $\boxed{2 - 1 - 3}$ $\boxed{1 - 2 - 3}$ $\boxed{1 - 3 - 2}$.

Theorem 6.9. (~~C~~ Cayley's formula). Let n ~~...~~ be a positive integer.

(a) The # of n-trees is n^{n-2}.

(b) Let $n \geq 2$. Let $d_1, d_2, \ldots, d_n$ be n positive integers with $d_1 + d_2 + \cdots + d_n = 2(n-1)$.
Then, the # of ~~t~~ n-trees such that $\deg(i) = d_i \ \forall i \in [n]$ is

$$\frac{(n-2)!}{(d_1-1)!(d_2-1)!\cdots(d_n-1)!}.$$

"Sanity check": Let's check that (b) $\Rightarrow$ (a). Assume $n \geq 2$.
Then, ~~every n-tree~~ for each n-tree, the degrees $\deg 1, \deg 2, \ldots, \deg n$ are positive integers & satisfy
$\deg 1 + \deg 2 + \cdots + \deg n = 2(n-1)$
(because the handshake lemma says $\deg 1 + \deg 2 + \cdots + \deg n = 2 \cdot \underbrace{(\# \text{ of edges})}_{= n-1 \text{ (by Thm. 6.8 } \cancel{k}, \gamma_4)} = 2(n-1)$).

Thus, Thm. 6.5 (b) should yield that the total # of n-trees is

$$\sum_{\substack{d_1, d_2, \ldots, d_n \\ \text{positive int.s,} \\ \text{with } d_1 + d_2 + \ldots + d_n = 2(n-1)}} \frac{(n-2)!}{(d_1 - 1)!\,(d_2 - 1)! \cdots (d_n - 1)!}$$

substitute $a_i = d_i - 1$

$$= \sum_{\substack{a_1, a_2, \ldots, a_n \in \mathbb{N}; \\ \text{with } a_1 + a_2 + \ldots + a_n = n-2}} \frac{(n-2)!}{a_1!\,a_2! \cdots a_n!}$$

$$= \sum_{\substack{a_1, a_2, \ldots, a_n \in \mathbb{N}, \\ \text{with } a_1 + a_2 + \ldots + a_n = n-2}} \frac{(n-2)!}{a_1!\,a_2! \cdots a_n!} 1^{a_1} 1^{a_2} \cdots 1^{a_n}$$

$$= (\underbrace{1 + 1 + \cdots + 1}_{n \text{ 1's}})^{n-2},$$

where we have applied the multinomial formula

$$(x_1+x_2+\dots+x_n)^m = \sum_{\substack{a_1,a_2,\dots,a_n\in\mathbb{N} \\ \text{with } a_1+a_2+\dots+a_n=m}} \frac{m!}{a_1!\,a_2!\dots a_n!} x_1^{a_1} x_2^{a_2} \dots x_n^{a_n}$$

(see [Galvin, Thm. 5.3]), to $m=n-2$ and $x_i=1$ in the last equality. This is, of course n^{n-2}, as Thm. 6.9(a) says.

<u>1st proof of Thm. 6.9.</u> First, use induction to prove (b).
Then, derive (a) from it as above. See [Galvin, §5].

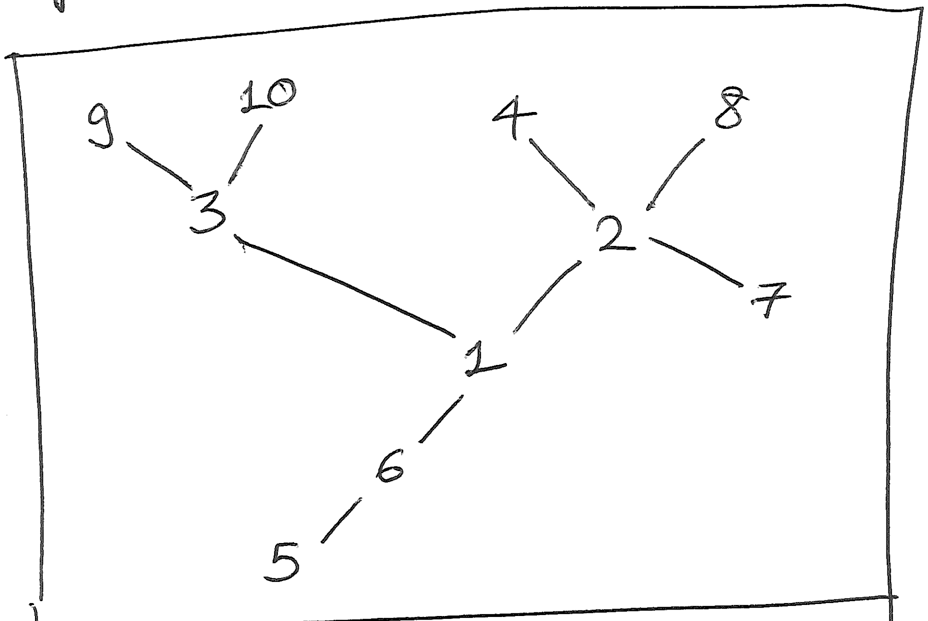
<u>2nd proof of Thm. 6.9.</u> (also in [Galvin, §6]).
Let's prove (a). ((b) is similar.)
WLOG assume $n\geq 2$.

<u>Idea:</u> encode each n-tree as a list of $n-2$ elements of $[n]$,
How? <u>Prüfer code</u> (Heinz Prüfer 1918).

Recall: any tree with ≥ 2 vertices has ≥ 1 leaf (by Thm. 6.8

T_8). (Actually, it has ≥ 2 leaves.)

Running example:

$T =$  and $n = 10$,

To construct the Prüfer code $P(T)$ of a tree T, do the following:

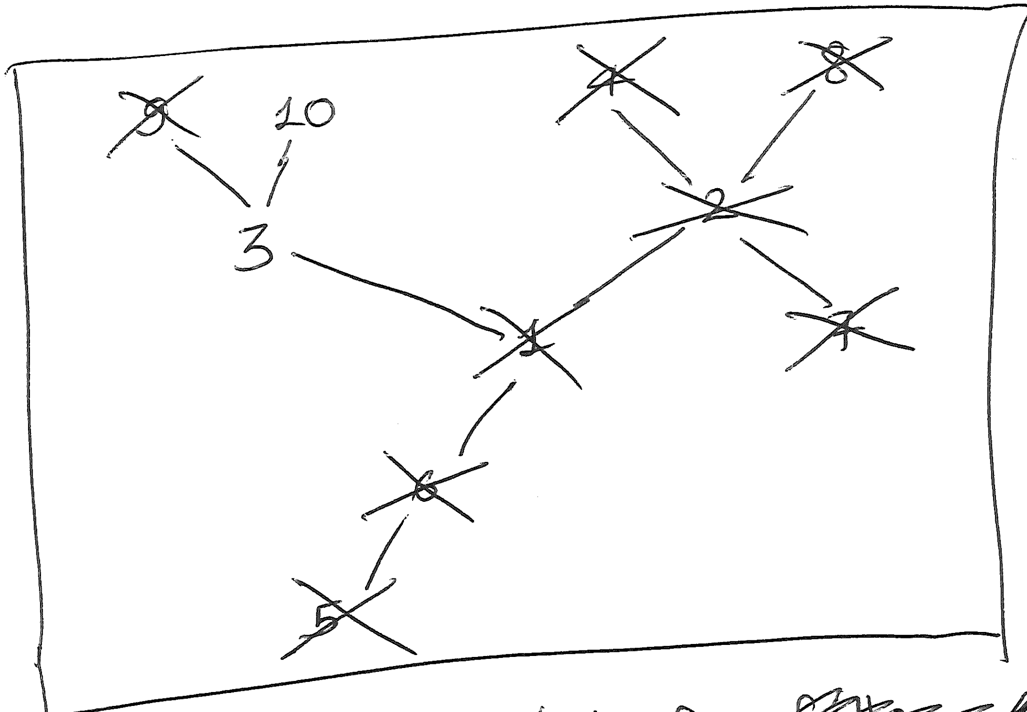
while T has > 2 vertices:

- let ℓ be the smallest leaf of T;
- write down the unique neighbor of ℓ;
- remove ℓ (along with the edge through it) from T (we say that "ℓ falls");

repeat this until T has only 2 vertices left.

In the above example,

$$P(T) = (2, 6, 1, 2, \cancel{8}, 2, 1, 3, \cancel{8}, 3).$$



Note that T stays a tree throughout the algorithm.

Generally, $P(T)$ is the list of ~~fallen leaves~~ neighbors written down (NOT the list of fallen leaves).

__Claim:__

$$\{n\text{-trees}\} \longrightarrow [n]^{n-2},$$
$$T \longmapsto P(T)$$

is a bijection.

Why?
We need a way to reconstruct T from $P(T)$.

First step: For each $i \in [n]$, we have

$$\underbrace{\deg_T i}_{\substack{= \deg i \\ \text{with respect} \\ \text{to } T}} = \left(\#\text{ of times } i \text{ occurs in } P(T)\right) + 1.$$

(Proof idea: If i is not one of the 2 surviving vertices, then, before i falls, all but one neighbors of i must have fallen. If i is one of the 2 survivors, argue similarly.)

Thus, from $P(T)$, we can reconstruct $\deg_T i$ for all $i \in [n]$.

$\Rightarrow$ We know which $i \in [n]$ are leaves.

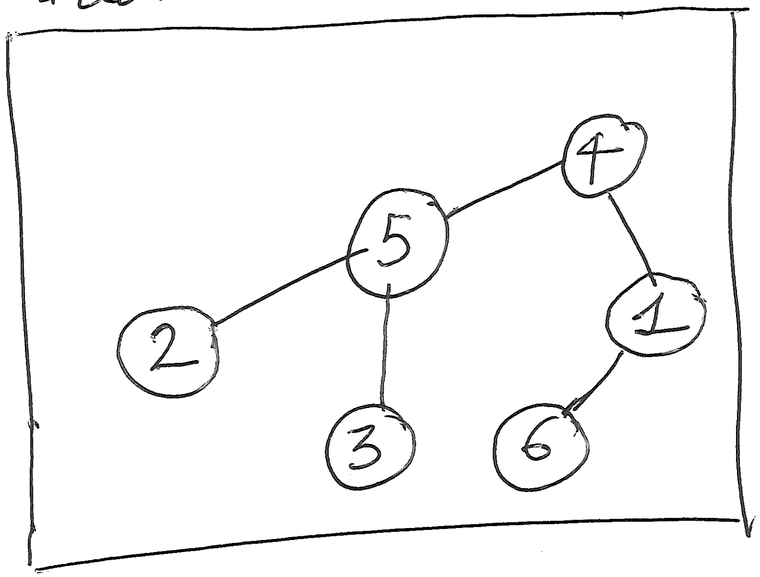
$\Rightarrow$ we know the smallest leaf of T.

$\Rightarrow$ we know what leaf was the first to fall in the algorithm, and we also know its unique neighbor.

Let T' be the tree obtained after the first leaf of T has fallen. Then, $P(T')$ is the list $P(T)$ without its first entry. By recursion, we can reconstruct T' from $P(T')$. (Note: T' does not contain the fallen leaf.) Having done that, we get T by attaching the first fallen leaf to its unique neighbor. $\Rightarrow$ P is injective.

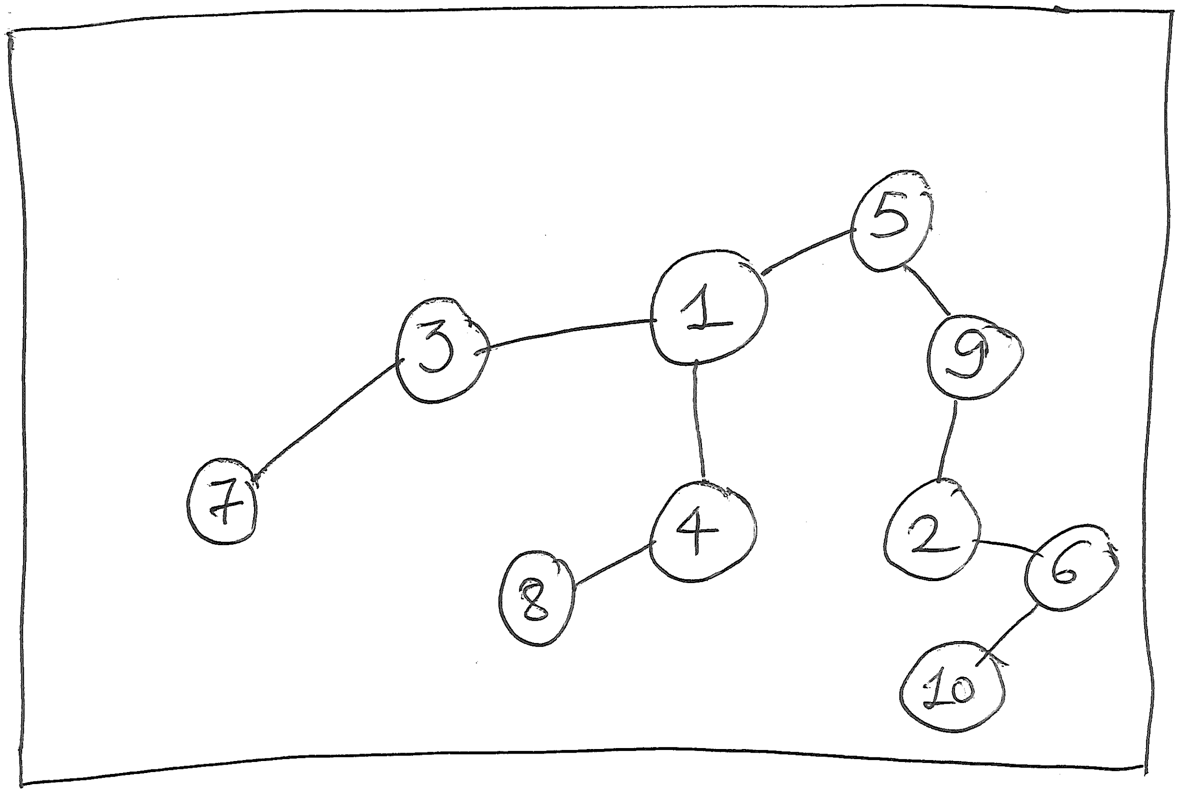
Example: $n=6$, $P(T) = (5, 5, 4, 1)$

$\Rightarrow T =$



(Note: The two survivors are adjacent, so we know where to attach 6.)

Example: $n = 10$, $P(T) = (3, 1, 4, 1, 5, 9, 2, 6)$.

$\Rightarrow T =$ 

We can also show that P is surjective.

$\Rightarrow P$ is a bijection.

$\Rightarrow |\{n\text{-trees}\}| = |[n]^{n-2}| = n^{n-2}$. This proves Thm. 6.9(a).

$\square$

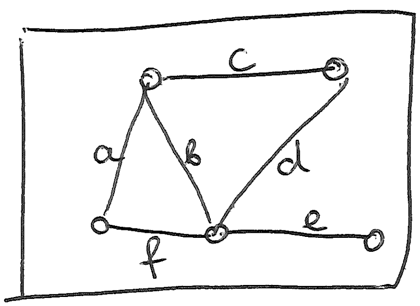
(b) Same idea.

Now, we will briefly consider spanning trees.

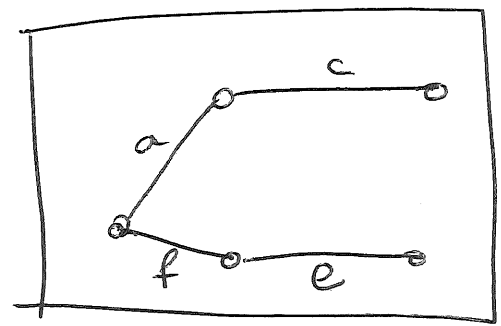
Idea: If G is a connected graph, then there are ways (usually, several of them) to remove some edges from G such that what remains is a tree.

Such a tree is called a spanning tree of G.

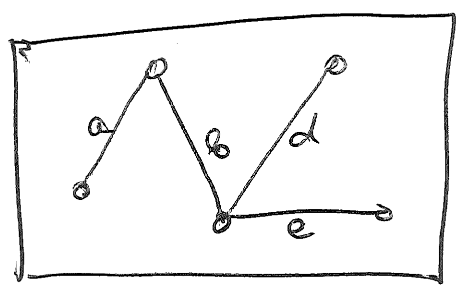
Example:



has spanning trees



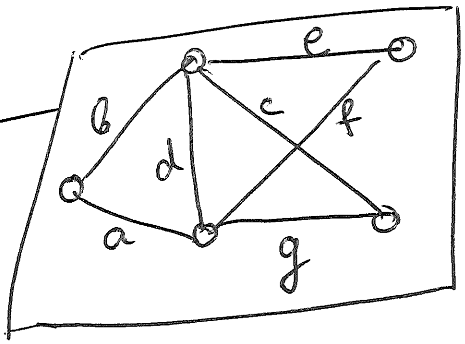
and



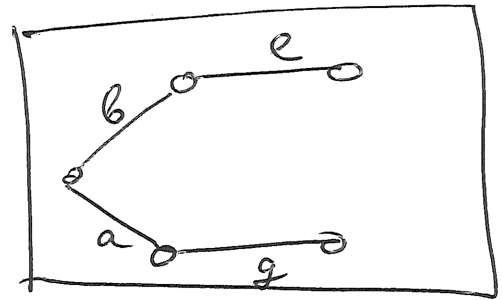
and others.

~~Idea: Let G be a graph. A spanning tree of G is (informally) a way to remove edges from G such that what remains is a tree~~

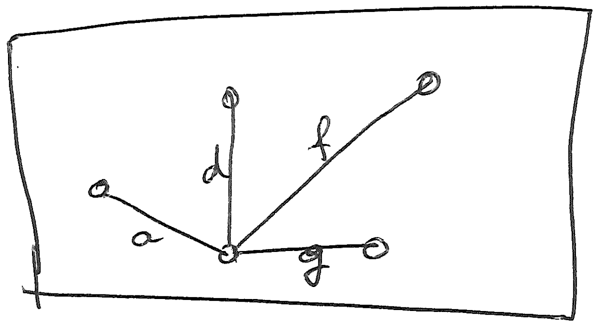
Example:



has spanning trees

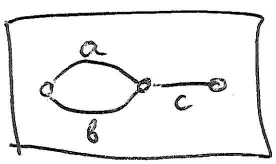


and

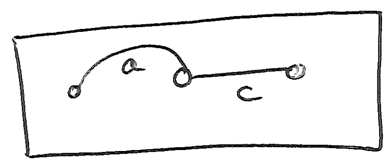


and various others.

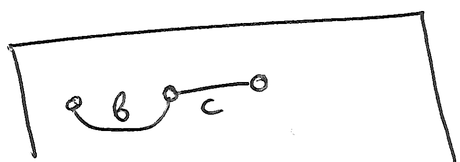
Example:



has spanning trees



and



.

To make this rigorous, we define:

Def. Let G be a graph.

(a) A subgraph of G is a graph H such that:

- each vertex of H is a vertex of G;
- each edge of H is an edge of G;
- each edge of H connects the same two vertices in H as in G.

(b) A subgraph of G is spanning if it contains each vertex of G.

(c) A spanning tree of G is a spanning subgraph of G that is a tree.

Examples: (a) If G is a tree, then the only spanning tree of G is G itself.

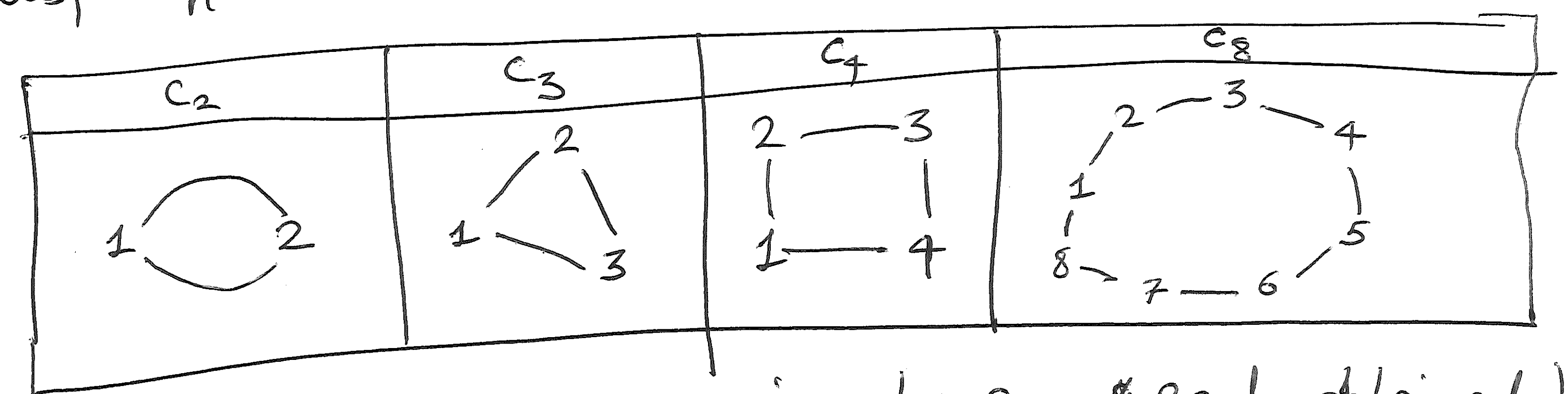
(b) Let $n \geq 2$. The cycle graph C_n has vertices $1, 2, \ldots, n$ and edges $12, 23, 34, \ldots, (n-1)n, n1$. (this is shorthand

for: an edge with endpoints 1, 2;
— // — 2, 3;
— // — 3, 4;

. . . . ;

an edge with endpoints $n-1, n$;
— // — $n, 1$.

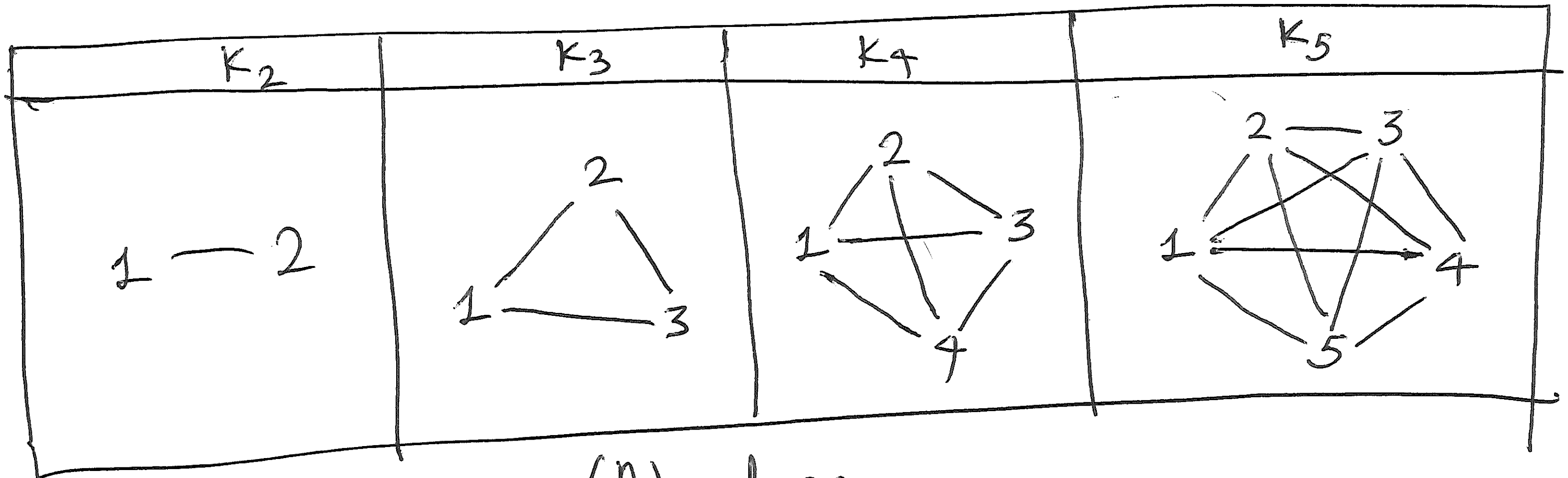
(If $n=2$, then we treat the two edges as different.)
Thus, C_n has n vertices & n edges.



Then, C_n has n spanning trees, each obtained by removing 1 edge.

(c) Let $n \in \mathbb{N}$. The <u>complete graph</u> K_n has vertices $1, 2, .., n$ and an edge between any two distinct vertices:

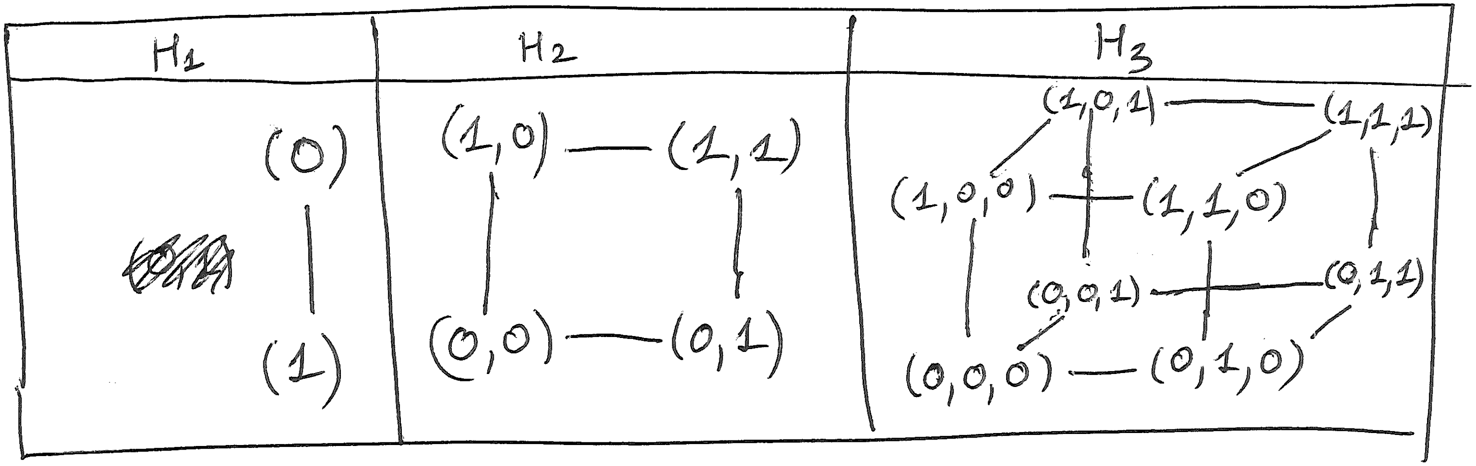
$K_n = ([n], \mathcal{P}_2([n]))$ as simple graph.



Note: K_n has $\binom{n}{2}$ edges.

The spanning trees of K_n are the n-trees. Thus, there are n^{n-2} of them (by Thm. 6.9(a)).

(d) Let $n \geq 1$. The <u>hypercube graph</u> H_n has ~~edges~~ vertex set $\{0,1\}^n$, with two vertices $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$ being adjacent if & only if the lists $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$ differ in only one entry. (It is a simple graph.)



H_n has 2^n vertices.

The # of spanning trees of H_n is $2^{2^n-n-1} \cdot \prod_{i=1}^{n} i^{\binom{n}{i}}$.

[Stanley, Algebraic Combinatorics, Example 9.12].

<u>Theorem 6.10 (Matrix-tree theorem, Kirchhoff).</u> Let G be a graph with n vertices $1, 2, \dots, n$. The <u>reduced Laplacian</u> of G is defined as the $(n-1)\times(n-1)$-matrix L whose (i,j)-th entry is

$$\begin{cases} \deg i, & \text{if } i=j; \\ -(\#\text{ of edges } i - j), & \text{if } i \neq j. \end{cases}$$

Then, the # of spanning trees of G is $\det L$.

For proofs, see [Stanley, Algebraic Combinatorics, Ch. 9].

<u>Example:</u> Redderive Thm. 6.9(a) using Thm. 6.10.

Let $n \geq 1$. Consider the complete graph K_n.

Setting $G = K_n$ in Thm. 6.10 we get

$$L = \underbrace{\left.\begin{pmatrix} n-1 & -1 & -1 & \cdots & -1 \\ -1 & n-1 & -1 & \cdots & -1 \\ -1 & -1 & n-1 & \cdots & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & \cdots & n-1 \end{pmatrix}\right\} n-1}_{n-1},$$

so $\det L = n^{n-2}$ (by some elementary linear algebra),

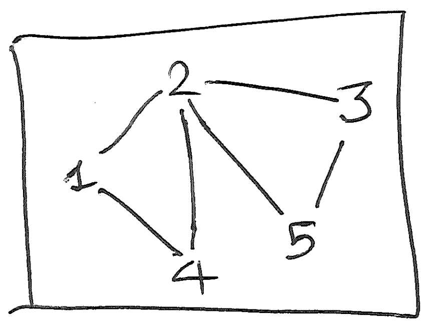
So Thm. 6.10 yields that the # of spanning trees of K_n is n^{n-2}. This is Thm. 6.9(a).

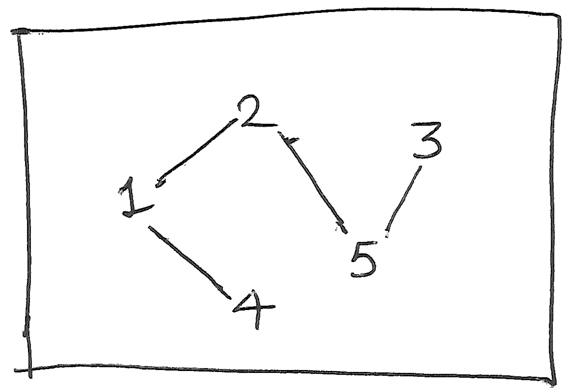
Thm. 6.11. Every connected graph G has a spanning tree.

1st proof. Construct a spanning tree of G using Kruskal's algorithm:

- Start with $(V, \emptyset)$, where V = vertex set of G.
- Keep adding edges from G to this graph, as long as this is possible without creating cycles.
- Once you're out of addable edges, you found a spanning tree of G. $\square$

Example:

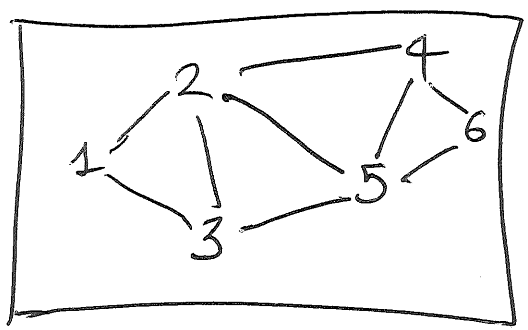
$G =$ 

$\rightsquigarrow$ 

<u>2nd proof.</u> Construct a spanning tree of G using the reverse-deletion algorithm:

- Start with G.
- Keep removing edges from the graph, as long as this does not disconnect the graph.
- Once you're out of removable edges, you found a spanning tree of G. □

<u>Example:</u>

G = 

⇝