Math 221 Winter 2023, Lecture 1: Introduction

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

0.1. What is this about?

My name is Darij Grinberg.

This is a course on **discrete mathematics**. To us, discrete mathematics means the mathematics of finite, discrete objects: integers, finite sets, occasionally some more complex creatures such as graphs and polynomials. Integer sequences, while theoretically infinite, are also included since one usually makes statements about finite pieces of the sequence. Much of linear algebra logically belongs to discrete mathematics, but there are separate courses entirely devoted to it, so we won't touch on it here.

Discrete mathematics is in contrast to **continuous mathematics**, which studies real numbers, continuous functions and infinite sets. This mostly begins with analysis (or calculus, which is its less rigorous variant).

So this course will introduce you to the major topics of discrete mathematics:

- **mathematical induction** and **recursion**;
- **elementary number theory** (the properties of divisibility, prime numbers, coprimality, possibly applications like the RSA cryptosystem);
- basic **enumerative combinatorics** (counting and binomial coefficients);
- basic **graph theory**.

We will neither go very deep nor be fully rigorous about everything. There are deeper, more specific classes on most of these subjects:

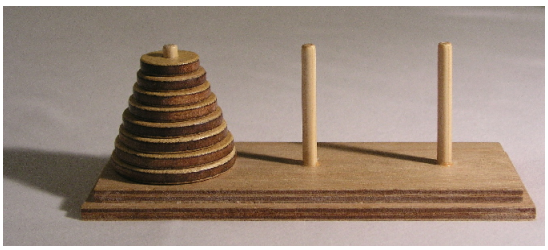
- Math 220 is a deeper introduction to mathematical proof.
- Math 222 is a quarter-length introduction to enumerative combinatorics. (I have notes for this on my website.)
- Math 530 is an introduction to graph theory. (I have notes for this on my website.)
- CS 303 is a course on cryptography.
- I plan to teach elementary number theory as a class next Fall (Math T480).

1. Induction and recursion

1.1. The Tower of Hanoi

Let me start with a puzzle called the **Tower of Hanoi**.

You have 3 pegs (or rods). The first peg has n disks stacked on it. The n disks have n different sizes, and they are stacked in the order of their size, with the smallest one on top. Here is how this looks like for $n = 8$ (with the 3 pegs numbered $1, 2, 3$ from left to right):



(image by User:Evanherk on Wikipedia, licensed under the CC Attribution-Share Alike 3.0 Unported License).

You can make a certain kind of moves ("**Hanoi moves**"): You can take the topmost disk from one peg and move it on top of another peg. However, you are only allowed to do this if this disk is smaller than the other disks currently on the latter peg; in other words, you must never stack a larger disk atop a smaller disk.

Your **goal** is to move all n disks onto the third peg.

This game can actually be played online, e.g., at `https://codepen.io/eliortabeka/pen/yOrrxG`. (Be warned that this site has $n = 7$ hardcoded into it. But you can easily fix this by modifying "disksNum = 3" and changing "minMoves = 127" to "minMoves = 0". Also note that the game allows you to win by moving all disks to peg 2 as well, but this is clearly not a significant difference.)

Let us analyze the case $n = 3$. In this case, one strategy to win the game (i.e., achieve the goal) is as follows:

1. Move the smallest disk from peg 1 to peg 3.
2. Move the middle disk from peg 1 to peg 2.
3. Move the smallest disk from peg 3 to peg 2.
4. Move the largest disk from peg 1 to peg 3.
5. Move the smallest disk from peg 2 to peg 1.
6. Move the middle disk from peg 2 to peg 3.

7. Move the smallest disk from peg 1 to peg 3.

So we can win in 7 moves for $n = 3$.

What about other values of n ? The questions we can ask are the following:

Question 1.1.1. **1.** Can we always win the game?
2. If so, then what is the smallest # of moves[1] we need to make?

Let us record the answers for small values of n:

- For $n = 0$, we win in 0 moves (since all disks – of which there are none – are on peg 3 already). This sounds very pedantic and pointless, but it's not a bad start.
- For $n = 1$, we win in 1 move (just move the single disk directly).
- For $n = 2$, we win in 3 moves. Fewer moves are not enough, for fairly simple logical reasons: We need 1 move to free the largest disk, then 1 move to move it to peg 3, then 1 more move to get the other disk on top of it.
- For $n = 3$, we win in 7 moves. But do we need 7 moves, or can we do with less?
- For $n = 4$, what happens?

Solving the problem by brute force gets harder and harder as n grows. But we can try to analyze our strategy for $n = 3$ and see if there is a pattern behind it.

We observe that the largest disk moves only once, and its move is right in the middle of the strategy. So our strategy for $n = 3$ can be summarized as follows:

1.–3. Move the two smaller disks from peg 1 onto peg 2.

4. Move the largest disk from peg 1 onto peg 3.

5.–7. Move the two smaller disks from peg 2 onto peg 3.

Moreover, the moves 1–3 in this strategy are essentially a Tower of Hanoi game played only with the two smaller disks, except that the goal is not to move them to peg 3 but to move them to peg 2 (but this doesn't matter, because the two games are clearly "isomorphic" – i.e., the roles of pegs 2 and 3 are swapped but otherwise everything is the same). The largest disk stays at the bottom of peg 1 all the time and thus does not prevent any of the moves (since all the other disks are smaller than it and thus can fit on top of it).

[1] The symbol "#" means "number".

Move 4 moves the newly liberated largest disk from peg 1 onto peg 3.

Moves 5–7 are again a little Tower of Hanoi game for the two smaller disks, except that now they have to be moved from peg 2 to peg 3. Again, the largest disk (which is now on the bottom of peg 3) does not interfere with any of the moves.

Now the logic behind the above strategy has become clear (and also easier to memorize).

Does this help us solve the $n = 4$ case?

Yes! We can win in 15 moves by a strategy that has the same structure:

1.–7. Move the three smaller disks from peg 1 onto peg 2. (This is a little Tower of Hanoi game for these three smaller disks. The largest disk rests at the bottom of peg 1 and does not interfere.)

8. Move the largest disk from peg 1 onto peg 3.

9.–15. Move the three smaller disks from peg 2 onto peg 3. (This is again a little Tower of Hanoi game for these three smaller disks. The largest disk rests at the bottom of peg 3 and does not interfere.)

Thus, we don't just have a strategy for $n = 3$ and one for $n = 4$, but actually a "meta-strategy" that lets us win the game for n disks if we know how to win it for $n - 1$ disks. We will still call this "meta-strategy" a strategy.

Let us summarize what we gain from this strategy.

Definition 1.1.2. For any integer $n \geq 0$, we let m_n denote the # of moves needed to win the Tower of Hanoi game with n disks. If the game cannot be won with n disks, then we set $m_n = \infty$ (where ∞ is not a number but just a symbol).

Thus, both of our Questions 1 and 2 boil down to computing m_n.

Here is a table of small values of m_n obtained using our strategy:

n	0	1	2	3	4	5	6	7	8
m_n	0	1	3	7	15	31	63	127	255

Note that these values are easily computed using our strategy, because in order to win the game for a given n, we have to win it for $n - 1$, then make one extra move, then win it for $n - 1$ again. So we get $m_n = m_{n-1} + 1 + m_{n-1} = 2m_{n-1} + 1$ (for $n \geq 1$).

Right?

Not so fast! We have proved that, e.g., the game can be won in 127 moves for $n = 7$. We have not proved that it cannot be won in fewer moves. So the

formula $m_n = 2m_{n-1} + 1$ has been proved not for the # of moves needed to win, but rather for the # of moves needed to win **using our strategy**. Maybe there is a better strategy that wins for $n = 7$ in (say) 109 moves?

So what we really have proved is the following:

Proposition 1.1.3. Let n be a positive integer. If m_{n-1} is an integer (i.e., if $m_{n-1} \neq \infty$), then $m_n \leq 2m_{n-1} + 1$.

To gain some writing experience, let us write out the proof in detail:

Proof. Assume that m_{n-1} is an integer. Thus, we can win the game for $n-1$ disks in m_{n-1} moves. Let S be the strategy (i.e., the sequence of moves) needed to do this. So the strategy S moves $n-1$ disks from peg 1 onto peg 3 in m_{n-1} moves.

Let S_{23} be the same strategy as S, but with the roles of pegs 2 and 3 swapped. Thus, S_{23} moves $n-1$ disks from peg 1 onto peg 2 in m_{n-1} moves.

Let S_{12} be the same strategy as S, but with the roles of pegs 1 and 2 swapped. Thus, S_{12} moves $n-1$ disks from peg 2 onto peg 3 in m_{n-1} moves.

Now, we proceed as follows to win the game with n disks:

A. We use strategy S_{23} to move the $n-1$ smaller disks from peg 1 onto peg 2. (This is allowed because the largest disk rests at the bottom of peg 1 and does not interfere with the movement of smaller disks.)

B. We move the largest disk from peg 1 onto peg 3. (This is allowed because this disk is free (i.e., there are no disks on top of it) and because peg 3 is empty, since all the other disks are on peg 2.)

C. We use strategy S_{12} to move the $n-1$ smaller disks from peg 2 onto peg 3. (Again, this is allowed since the largest disk rests at the bottom of peg 3 and does not interfere.)

This strategy wins the game (for n disks) in $m_{n-1} + 1 + m_{n-1} = 2m_{n-1} + 1$ many moves. So the game for n disks can be won in $2m_{n-1} + 1$ many moves. In other words, $m_n \leq 2m_{n-1} + 1$. This proves Proposition 1.1.3. □

Now, let us see if the inequality $m_n \leq 2m_{n-1} + 1$ that we have proved is an equality or just an inequality – i.e., whether the above strategy is optimal or there is a faster one. I claim it is the former:

Proposition 1.1.4. Let n be a positive integer. If m_{n-1} is an integer (i.e., if $m_{n-1} \neq \infty$), then $m_n = 2m_{n-1} + 1$.

Proof. Again, assume that m_{n-1} is an integer.

We need to show that $m_n = 2m_{n-1} + 1$. It suffices to show that $m_n \geq 2m_{n-1} + 1$ (since Proposition 1.1.3 yields $m_n \leq 2m_{n-1} + 1$, and we can combine these two inequalities to get $m_n = 2m_{n-1} + 1$). In other words, it suffices to show that any winning strategy for n disks has at least $2m_{n-1} + 1$ many moves.

So let us consider a winning strategy T for n disks. Somewhere during the strategy T, the largest disk has to move (since it starts out on peg 1 but has to end up on peg 3). Let us refer to these moves (the ones that move the largest disk) as the **special moves**. There may be several special moves or just one, but as we just said, there has to be **at least** one.

Before the first special move can happen, the smallest $n - 1$ disks have to be moved away from peg 1 (since they would otherwise block the largest disk from moving). Moreover, these smallest $n - 1$ disks must all be moved onto the same peg (since otherwise, both pegs 2 and 3 would be occupied, and then the largest disk would have nowhere to move). Thus, before the first special move can happen, we must have won the Tower of Hanoi game for $n - 1$ disks. Hence, before the first special move can happen, we already need to have made m_{n-1} moves (since m_{n-1} is the smallest # of moves that can win the game for $n - 1$ disks).

Now, consider what happens **after the last special move**. This last special move necessarily moves the largest disk to peg 3 (since that's where this disk has to come to rest). After that, we still need to move all the other disks onto peg 3. At the time we are making the last special moves, these other disks must all be on the same peg (since they can be neither on the peg from which the largest disk is moving, nor on the peg to which it is moving[2]). Therefore, after the last special move, we still need to move all the remaining $n - 1$ disks from one peg to another. And this is again tantamount to winning the game for $n - 1$ disks. So this again needs at least m_{n-1} moves.

So in total, we know that our strategy T needs to have

1. at least m_{n-1} moves before the first special move,
2. at least one special move, and
3. at least m_{n-1} moves after the last special move.

Thus, it needs to have at least $m_{n-1} + 1 + m_{n-1} = 2m_{n-1} + 1$ many moves in total. This proves $m_n \geq 2m_{n-1} + 1$. □

Proposition 1.1.4 confirms the table we have carelessly made before:

n	0	1	2	3	4	5	6	7	8
m_n	0	1	3	7	15	31	63	127	255

[2]because in either case, they would block the move of the largest disk

Obviously, you can keep using Proposition 1.1.4 to compute $m_9, m_{10}, m_{11}, \ldots$. Indeed, the equation

$$m_n = 2m_{n-1} + 1 \tag{1}$$

is what is called a **recursive formula** for the numbers m_n. This means a formula that allows you to compute m_n using the previous values $m_0, m_1, \ldots, m_{n-1}$. In our case, we only need the direct predecessor m_{n-1}, so this is a particularly convenient recursive formula.

Still, can we perhaps do better? Can we find an **explicit formula** – i.e., one that gives us m_n directly?

Some of you have already guessed such a formula:

$$m_n = 2^n - 1.$$

Is there a way to see this without guessing? Let's try applying the recursive formula (1) again and again, simplifying each time:

$$\begin{aligned}
m_n &= 2m_{n-1} + 1 && \text{(by (1))} \\
&= 2\left(2m_{n-2} + 1\right) + 1 && \text{(by (1), applied to } n-1\text{)} \\
&= 4m_{n-2} + 2 + 1 \\
&= 4\left(2m_{n-3} + 1\right) + 2 + 1 && \text{(by (1), applied to } n-2\text{)} \\
&= 8m_{n-3} + 4 + 2 + 1 \\
&= 8\left(2m_{n-4} + 1\right) + 4 + 2 + 1 && \text{(by (1), applied to } n-3\text{)} \\
&= 16m_{n-4} + 8 + 4 + 2 + 1 \\
&= \cdots && \text{(keep going until you reach } m_0\text{)} \\
&= 2^n \underbrace{m_0}_{=0} + 2^{n-1} + 2^{n-2} + \cdots + 2^0 \\
&= 2^{n-1} + 2^{n-2} + \cdots + 2^0 \\
&= 2^0 + 2^1 + 2^2 + \cdots + 2^{n-1}.
\end{aligned}$$

I claim that the right hand side is $2^n - 1$. Next time, we will see why.