

RECURSION

I Never
Understood
Recursion



Are you someone who understood
recursion in one go? If so, you are a genius;
if not, you are a **genius++**.



Recursion had been a **big source of frustration** for me throughout my programming journey.



When someone explained their recursive reasoning to me, I was able to grasp it, but I always found it **difficult to come up with my own.**



In this post I will explain recursion as if I am
explaining to a 12-year old.



- ➡ Someone in a movie theater asks you what row you're sitting in.
- 🚫 You don't want to count, so you ask the person in front of you what row they are sitting in,
- 👍 knowing that you'll respond one greater than their answer.
- 🔄 The person in front will ask the person in front of them.
- ✅ This will keep happening until word reaches the front row, and it is easy to respond: "I'm in row 1!"
- 😊 From there the correct message(incremented by one each row) will eventually makes it way back to the person who asked.

Why is this a good explanation? It gets across three points:

- Some questions may be naturally recursive, and that may be simpler to solve recursively.
- Recursively, you can ask “how many people are in front of me + 1?” instead of “what row am I in?” with a worst-case scenario of nobody in front of me.
- Additionally, it illustrates the concept of a recursive call stack and how calls are added to and removed from the stack.

Avoid technical terms for a moment and just think like “Do this problem follow same pattern for every output?” if yes, then it can be solved recursively.



And to solve a recursive problem you need practice. Yes, even after understanding how recursion works you'll get stuck at some point. The more you solve problems the more you'll find similar patterns in every problem.