Longest Common Substring

You've seen one dynamic programming problem so far. What are the takeaways?

- Dynamic programming is useful when you're trying to optimize something given a constraint. In the knapsack problem, you had to maximize the value of the goods you stole, constrained by the size of the knapsack.
- You can use dynamic programming when the problem can be broken into discrete subproblems, and they don't depend on each other.

It can be hard to come up with a dynamic-programming solution. That's what we'll focus on in this section. Some general tips follow:

- Every dynamic-programming solution involves a grid.
- The values in the cells are usually what you're trying to optimize. For the knapsack problem, the values were the value of the goods.
- Each cell is a subproblem, so think about how you can divide your problem into subproblems. That will help you figure out what the axes are.

Computer scientists sometimes joke about using the **Feynman algorithm**. The Feynman algorithm is named after the famous physicist Richard Feynman, and it works like this:

1. Write down the problem.
2. Think real hard.
3. Write down the solution.

So is dynamic programming ever really used? Yes:

- Longest common subsequence and longest common substring are examples of dynamic programming, which can be solved with a grid.
- Biologists use the longest common subsequence to find similarities in DNA strands. They can use this to tell how similar two animals or two diseases are. The longest common subsequence is being used to find a cure for multiple sclerosis.
- Have you ever used diff (like git diff)? Diff tells you the differences between two files, and it uses dynamic programming to do so.
- We talked about string similarity. Levenshtein distance measures how similar two strings are, and it uses dynamic programming. Levenshtein distance is used for everything from spell-check to figuring out whether a user is uploading copyrighted data.
- Have you ever used an app that does word wrap, like Microsoft Word? How does it figure out where to wrap so that the line length stays consistent? Dynamic programming!