

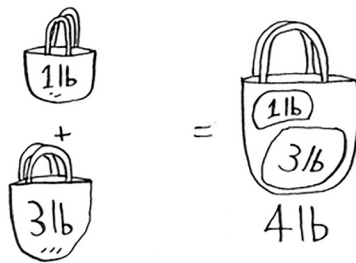
That's impractical for any reasonable number of goods. In chapter 8, you saw how to calculate an *approximate* solution. That solution will be close to the optimal solution, but it may not be the optimal solution.

So how do you calculate the optimal solution?

Dynamic programming

Answer: With dynamic programming! Let's see how the dynamic-programming algorithm works here. Dynamic programming starts by solving subproblems and builds up to solving the big problem.

For the knapsack problem, you'll start by solving the problem for smaller knapsacks (or "sub-knapsacks") and then work up to solving the original problem.



Dynamic programming is a hard concept, so don't worry if you don't get it right away. We're going to look at a lot of examples.

I'll start by showing you the algorithm in action first. After you've seen it in action once, you'll have a lot of questions! I'll do my best to address every question.