

The Knapsack Problem FAQ

- Can you steal fractions of an item?

You can't. With the dynamic-programming solution, you either take the item or not. There's no way for it to figure out that you should take half an item.

- Handling items that depend on each other. Suppose you want to go to Paris, so you add a couple of items on the list. These places take a lot of time, because first you have to travel from London to Paris. That takes half a day. If you want to do all three items, it will take four and a half days. Wait, that's not right. You don't have to travel to Paris for each item. Once you're in Paris, each item should only take a day. So it should be one day per item + half a day of travel = 3.5 days, not 4.5 days. If you put the Eiffel Tower in your knapsack, then the Louvre becomes "cheaper"—it will only cost you a day instead of 1.5 days. How do you model this in dynamic programming?

You can't. Dynamic programming is powerful because it can solve subproblems and use those answers to solve the big problem. Dynamic programming only works when each subproblem is discrete—when it doesn't depend on other subproblems. That means there's no way to account for Paris using the dynamic-programming algorithm.