Dynamic Programming

The Knapsack Problem

Advanced - java

Imagine that you are a thief breaking into a house. There are so many valuables to steal, but you’re just one person who can only carry so much. Each item has a weight and value, and your goal is to maximize the total value of items while remaining within the weight limit of your knapsack. Create a `knapsack()` method that takes in:

- the total amount of weight you can carry
- an array of the weights of all of the items
- an array of the values of all of the items

and returns the maximum value that you will be able to carry.

For example, let’s say your knapsack can carry `10` units of weight. The item weights are `[3, 6, 8]` and their values are `[50, 60, 100]`. Your knapsack function should return `110` since you can carry the first and second items, whose values total `110`. If you tried to carry the third item, which has the value of `100`, you wouldn’t be able to fit anything else in the knapsack.

This challenge was reported to have been asked at interviews with Amazon. If you’ve covered the material in Pass the Technical Interview with Java or an equivalent, you should be able to solve this challenge. If you have trouble, try refreshing your knowledge with its Knapsack problem walkthrough.