Programming Assignment #3

Help Center

Warning: The hard deadline has passed. You can attempt it, but **you will not get credit for it**. You are welcome to try it as a learning exercise.

☐ **In accordance with the Coursera Honor Code, I (francois-guillaume rideau) certify that the answers here are my own work.**

Question 1

In this programming problem and the next you'll code up the knapsack algorithm from lecture.

Let's start with a warm-up. Download the text file here. This file describes a knapsack instance, and it has the following format:

[knapsack_size][number_of_items]

[value_1] [weight_1]

[value_2] [weight_2]

...

For example, the third line of the file is "50074 659", indicating that the second item has value 50074 and size 659, respectively.

You can assume that all numbers are positive. You should assume that item weights and the knapsack capacity are integers.

In the box below, type in the value of the optimal solution.

ADVICE: If you're not getting the correct answer, try debugging your algorithm using some small test cases. And then post them to the discussion forum!

Question 2

This problem also asks you to solve a knapsack instance, but a much bigger one.

Download the text file here. This file describes a knapsack instance, and it has the following

format:

[knapsack_size][number_of_items]

[value_1] [weight_1]

[value_2] [weight_2]

...

For example, the third line of the file is "50074 834558", indicating that the second item has value 50074 and size 834558, respectively. As before, you should assume that item weights and the knapsack capacity are integers.

This instance is so big that the straightforward iterative implemetation uses an infeasible amount of time and space. So you will have to be creative to compute an optimal solution. One idea is to go back to a recursive implementation, solving subproblems --- and, of course, caching the results to avoid redundant work --- only on an "as needed" basis. Also, be sure to think about appropriate data structures for storing and looking up solutions to subproblems.

In the box below, type in the value of the optimal solution.

ADVICE: If you're not getting the correct answer, try debugging your algorithm using some small test cases. And then post them to the discussion forum!

☐ **In accordance with the Coursera Honor Code, I (francois-guillaume rideau) certify that the answers here are my own work.**

Submit Answers Save Answers

You cannot submit your work until you agree to the Honor Code. Thanks!