CCPS616 Lab 5 – Dynamic 0-1 Knapsack

Preamble

In this lab you will write a C++ program that finds an optimal solution to the 0-1 knapsack problem discussed in class. Your implementation must be from the ground up without linking any external libraries. Your lab should be completed in a single cpp file called **knapsack.cpp**, and should compile at the command with the provided tester as follows:

```
g++ -o lab5 knapsack.cpp lab5.cpp
```

Lab Description

In this week's lab you will implement two algorithms for solving the 0-1 knapsack problem. The first algorithm will be a simple O(2^n) brute force search. For the small data sets we'll be testing on, this won't be too costly. The second solution will use dynamic programming to achieve pseudo-polynomial time complexity.

Input to your program will be represented the same as Lab #4. An integer indicating the maximum weight you can carry, and an array of integer pairs representing each item's weight and value. Below, the max weight is 5, the first item has a weight of 2 and a value of 1, the second has a weight of 3 and a value of 3, the third has a weight of 4 and a value of 5.

```
5
[ (2, 1), (3, 3), (4, 5) ]
```

Your task is to choose those items that maximize the value in your knapsack without going over the weight limit. In this case, the optimal choice is taking the third item alone. Results will be returned in the form of an array of integer pairs representing the weights and values of those items chosen. Details of the data representation are found in the accompanying lab5.cpp file.

Testing

Your program will be tested on a series of small examples with known results to verify the correctness of your algorithm. After this, you will be tested on randomly generated data. The size of the input will remain relatively small to ensure the brute force approach remains tractable. The result of your brute force implementation will be used to verify the correctness of your dynamic programming results in these random trials. As in previous labs, build your solution on the provided template (knapsack.cpp).

Submission

Submit your source file (**knapsack.cpp**) on D2L.

Labs may be submitted individually, or in groups of up to **two**. If you submit as a group, be sure to include both names in the D2L submission as well as your source code.