Dynamic Programming Project (Spring 2023/2024)

Gandalf the Wizard

In the mystical realm of Middle-earth, Gandalf, the famous wizard revered for his magical prowess, is often sought after by kings and lords to craft enchanted artifacts and items of great power. Among his most esteemed patrons is Lord Elrond, a noble ruler known for his generosity and protection of Gandalf.

Whenever Gandalf completes an extraordinary task for Lord Elrond, the lord rewards him with chests (boxes) filled with precious treasures, including rare metals and enchanted gems. Gandalf will have to select one of these chests as his reward. Each chest holds a variety of these valuables, such as gold, silver, diamonds, and sapphires. Each type of metal and gem possesses a specific weight and value, which remains constant throughout Middle-earth. Some chests contain only one type of treasure, while others may contain a mix of different types.

Gandalf faces the challenge of determining the value of each chest without opening it. However, he can know the weight of the chest quickly. Armed with knowledge of the weight and value of each type of treasure in Middle-earth, Gandalf seeks to estimate the minimum possible value of a chest based solely on its weight.

The task at hand is to help Gandalf calculate this worst-case (minimum) value for a chest, given its weight and the known characteristics of the metals and gems. Each chest's contents could potentially vary in configuration, leading to different potential values. Gandalf's strategy is to always assume the scenario where the chest contains the least valuable configuration possible.

For instance, imagine a chest weighing 0.5 kg. Considering the following details about treasures in Middle-earth:

- A Gold piece weights 15 grams, has a value of 25 Gondor Coins
- A Diamond piece weighs 5 grams, has a value of 20 Gondor Coins
- A Silver piece weight 25 grams, has a value of 30 Gondor Coins

Here are some possible values for the chest:

1. 600 Gondor Coins (20 Silver pieces).
2. 2000 Gondor Coins (100 Diamond pieces).
3. 845 Gondor Coins (33 Gold pieces + 1 Diamond piece).

4. 950 Gondor Coins (10 Gold pieces, 20 Diamond pieces and 10 Silver pieces)

And many other possibilities/configurations

The least possible value of the chest among all those possibilities is 600. This is the value Gandalf will have to assume, so that he knows that the chest is at least (in the worst case) is worth 600 Gondor Coins.

Your objective is to assist Gandalf in efficiently determining this worst-case (minimum) value for any given chest, leveraging his knowledge of weights and values of the treasures in Middle-earth. Dynamic programming offers a strategic approach to solve this problem effectively without resorting to exhaustive enumeration of all potential chest contents (brute force).

The Input Format:

Chest Weight (in grams)
Number of gems/metals in the kingdom
List of gem/metal piece information. For each one you have first the piece weight then its value.

The Output Format:

The minimum possible value for the chest.

The above example would be presented as follows on the hackerrank:

Input:

```
500
3
15 25
5 20
25 30
```

Output:

```
600
```

Here is a reminder of the outline of your solution:

Part1: Divide & Conquer

1- Define the value returned by the function f which we want to optimize.
2- Define the parameters which f depends on.
3- Draw the recursion tree for f using the values from the example above.
4- Write the recursive (divide and conquer) code to solve the question.

Part2: Dynamic Programming

5- Draw the table and determine the dependencies between the table cells.
6- Determine the direction of movement within the table.
7- Write the Dynamic programming code which fills the table(s).
8- Write the code that will print the final answer.

You are requested to submit a report that explains each of the steps above which also includes graphs and figures to explain your solution and the rationale behind it. Also, you need to submit your working code on the hackerrank website as usual. The link will be provided on Moodle/FB Group.

One the hackerrank you submit the dynamic programming code, however, in the report you must show both the divide and conquer code and the dynamic programming code.

Important:

To solve this task you are not allowed to copy/be inspired by any piece of code from the internet, ChatGPT-like systems or from a colleague or from anyone or any place.

If any percentage of resemblance is found between your code and a code listed on the internet (even if the code is a solution for a different problem), it will be considered cheating.

You are only allowed to check the code of the three problems we studied at class and the last years problems I posted on Moodle.

If it has been proven that you cheated on this task (no matter how small the percentage is), you will get zero in the final exam mark.

One final hint to help you with this question: This question is very similar to one of the 3 DP examples we studied in class.

Dr. Samer Arandi