

Algorithms Homework 2

Dynamic Programming & Greedy Algorithm

Due data: May 6, 2025 23:59

0-1 Knapsack Problem is one of the most famous problem. This problem can be used in many different domains, especially for the optimization problems. The other well known reason for this problem is that it's a *NP-Complete* problem. In most of the case we can only find an inefficient algorithm for $O(nW)$ or $O(2^n)$ time to find the optimal solution. In some case we may use greedy algorithm to find an approximate solution. If the optimal solution find a set of items with total profit: 100, and the greedy method only find a set of items with total profit: 50. Then greedy algorithm is a $\frac{100}{50} = 2$ approximation of the optimal solution.

In this problem, you are asked to implement:

- the bottom-up method ($O(nW)$),
- the top-down method ($O(2^n)$) and
- the greedy method ($O(n \log n)$) (choosing by profit/weight)

In your implementation, you can compare the running time and quality of the greedy algorithm on:

- different number of items. (e.g. 10, 20, 30)
- different weight range (e.g. [1-1], [1-10], [1-10] \times 10, [1-1000], ...)

You need to submit a zip/rar file with

- A folder with name *code* includes all of your codes (.cpp, .h, and etc.)
 - Including all the versions you implemented.
- A pdf file with name report.pdf.

In your report, at least you need to explain

- The contains for each of your source code (what you implement)
- The different input setting you chose to test your programs.
- The experiment results of the different input.

You may also

- Try different things may affect the performance.
 - Different types/orders for the input data

Note: you may see the following page to measure the running time of your program
<https://www.geeksforgeeks.org/measure-execution-time-function-cpp/>