DAT600: Algorithm Theory

Dynamic and Greedy Algorithms

Assignment – 2: Compulsory

Code of Conduct

The principle is trust, participation and collaboration for better learning.
So please, do your best to learn and don't try to cheat. It is allowed and encourage to collaborate with others but for the purpose of learning, not just copy-pasting. You are allowed to consult and get inspired by external resources, but you should mention them in your report. NB! You may be asked to explain your work to the student assistents and/or the course responsible. Typically via a random selection, or some other relevant reasons such as the permission to use a good quality work as a good example, or of course in case of suspicion of plagiarism .

Delivery

Write a short and concise report in which you solve the following tasks. Add code sections when ever necessary, and a link to your GitHub repository if you find more convenient.

I Problem Matrix Chain Multiplication

Define a problem for matrix chain multiplication with 5 matrices. For example from A1..A5 with their respective dimensions. For an example see page 376 from the text book.

1. Solve the parenthesization problem by hand using the recursive formula to fill the memoisation tables m and s.
2. Implement a dynamic programming program that solves the problem for any number of matrices.
3. Does it exist a greedy choice that could apply to this problem?

II Fractional and 0-1 Knapsack

Pages 426-427 from the text book present the knapsack problem. Write a program that takes as input the possible units (kg, price), and the maximum capacity of the knapsack, and does the following:

1. Automatically generates 0-1 knapsack problems with their corresponding solutions.
2. Automatically generates fractional knapsack problems with their corresponding solutions.

III Problem Greedy + Dynamic (coin change)

Given an array of coins \$ c*1* $< c2 < \ldots <$ c_n \$, the objective is to determine the fewest coins needed to achieve a total of N.

1. Propose a greedy solution to minimize the number of coins required for a given total N.
2. Some currency systems may pose challenges for a greedy approach. For instance, with $coins = [1, 5, 11]$, a greedy solution for $N = 15$ will not yield the minimum number of coins $11 + 1 + 1 + 1 + 1 = 15$ is 5 coins but an optimal solution would be $5 + 5 + 5 = 15$ which is 3 coins.
3. Devise a new solution that accommodates any currency system, ensuring an optimal global solution for the minimum number of coins required. (Note: You can always assume the currency system has a coin equal to 1).
4. Find out if the Norwegian coin system is greedy, $coins = [1, 5, 10, 20]$.
5. What is the running time of these two algorithms?