CSDS 310: Algorithms Fall 2024

Assignment 3: Dynamic Programming

Instructor: Orhan Özgüner Due: Sunday, November 3 before 11:59 PM

Provide a dynamic programming solution to the following questions. You do not have to follow these steps:

Here are guided steps to help you while thinking about a solution:

(i) Identify the "last" question you need to answer in developing a solution.

(ii) Define and prove optimal substructure.

(iii) Define subproblems, express the solution to the overall problem in terms of the subproblems.

(iv) Formulate a recursive solution to the subproblems. Do not forget to specify the base case(s).

(v) Characterize the runtime of the resulting procedure assuming that you would implement your solution using a bottom-up procedure.

(vi) Provide the pseudo code of the bottom-up procedure you use to compute the value of the optimal solution, as well as the procedure for reconstructing the optimal solution.

Problem 1

We are given n types of coin denominations with integer values v_1, v_2, ..., v_n. Given an integer t, we would like to compute the minimum number of coins to make change for t (i.e., we would like to compute the minimum number of coins that add up to t, where repetitions are allowed). We know that one of the coins has value 1, so we can always make change for any amount of money t. For example, if we have coin denominations of 1, 2, and 5, then the optimal solution for $t = 9$ is 5, 2, 2.

Problem 2

Given two strings $X =< x_1, x_2, ..., x_m >$ and $Y =< y_1, y_2, ..., y_n >$, the edit distance between X and Y is defined as the minimum number of edit operations (*replacement*, *insertion*, or *deletion* of a character) required to convert X to Y. For example, the edit distance between $X = esteban$ and $Y = stephen$ is 4, comprising of 1 deletion (e), 1 insertion (h), and 2 replacements ($b \rightarrow p$ and $a \rightarrow e$). We would like to compute the edit distance between two given strings.

Problem 3

We are given n currencies and an exchange rate r_{ij} for any pair of currencies i an j. Namely, if we exchange 1 unit of currency i with currency j, we receive r_{ij} units of currency j. If we are given a source currency s and a target currency t, then we can go through a path of different currencies to reach t from s so as to maximize our profit. The markets can also charge an exchange fee depending on the number of exchanges we make. For example if the exchange fee is $f(k)$ for making k exchanges and we start with 1 unit of currency s, then the path of exchanges $s \to t$ will yield $r_{st} - f(1)$ units of currency t, whereas the path of exchanges $s \to i \to j \to t$ will yield $r_{si} \times r_{ij} \times r_{jt} - f(3)$ units of currency t. The problem is to find the sequence of exchanges that will maximize the amount of target currency t we can obtain for a given source currency s.