Homework 1.
Due: Monday, May 23, 2022 before 8am EDT.

[DPV] Practice Dynamic Programming Problems

These are practice problems from DPV to help you to become more familiar with DP, these problems will not be graded. It is not compulsory to finish these problems.

[DPV] Problem 6.4 – Dictionary lookup

You are given a string of n characters s[1...n],which you believe to be a corrupted text document in which all punctuation has vanished...

[DPV] Problem 6.8 – Longest common substring

Given two strings x = x1x2...xn and y = y1y2 ...ym, we wish to find the length of their longest common substring...

[DPV] Problem 6.18 – Making change II

Consider the following variation on the change-making problem (Exercise 6.17): you are given denominations x1, x2, . . . , xn, ...

[DPV] Problem 6.19 – Making change k

Given an unlimited supply of coins of denominations x1, x2, . . . , xn, we wish to make change for a value v using at most k coins...

[DPV] Problem 6.20 – Optimal Binary Search Tree

Suppose we know the frequency with which keywords occur in programs of a certain language, for instance ...

[DPV] Problem 6.26 – Alignment

Sequence alignment. When a new gene is discovered, a standard approach to understanding its function is to look through a database of known genes and find close matches...

See next page for homework problems.

DP Homework

Problem 1 Longest Common Sub*!?*

Given two strings $X = x_1, x_2, \ldots, x_n$ and $Y = y_1, y_2, \ldots, y_m$ **give a dynamic programming algorithm** to find the **length** k of the longest string $Z = z_1, \ldots, z_k$ where Z appears as a **substring** of X and as a **subsequence** of Y. Recall, a substring is **consecutive** elements.

For example, for the following input:

$$\begin{aligned} X &= a, \mathbf{b}, \mathbf{d}, \mathbf{b}, \mathbf{a}, b, f, g, d \\ Y &= \mathbf{b}, e, t, f, \mathbf{d}, \mathbf{b}, f, \mathbf{a}, f, r \end{aligned}$$

then the answer is 4 (since, b, d, b, a is a substring of X and it is also a subsequence of Y). You do not need to output the actual substring, just its length.

(Faster (and correct) in asymptotic $O(\cdot)$ notation is worth more credit.)

(a) Define the entries of your table in words. E.g., $T(i)$ or $T(i, j)$ is ....

(b) State recurrence for entries of table in terms of smaller subproblems.

(c) Write pseudocode for your algorithm to solve this problem.

(d) Analyze the running time of your algorithm.

Problem 2 [DPV] 6.17 – Coin changing (unlimited supply of each denomination)

(a) Define the entries of your table in words. E.g., $T(i)$ or $T(i, j)$ is ....

(b) State recurrence for entries of table in terms of smaller subproblems.

(c) Write pseudocode for your algorithm to solve this problem.

(d) Analyze the running time of your algorithm.