CS3230: Design and Analysis of Algorithms
Semester 2, 2021-22, School of Computing, NUS

Practice Problem Set
Dynamic Programming & Greedy Algorithms

April 1, 2022

Instructions

- This problem set is **completely optional**. There is no need to submit the solutions.
- Post on the LumiNUS forums if you will face any problem while solving the questions.

Question 1: In the lecture we have seen dynamic programming algorithm that computes LCS between two input strings. Now suppose you have m input strings each of length n. Can you extend the dynamic programming algorithm covered in the lecture to find a largest subsequence that is common to all the m input strings? (Try to achieve running time of $O(mn^m)$.)

Question 2: Given two strings x and y the *edit distance*, denoted by $ed(x, y)$, between them is the minimum number of character insertion, deletion and substitution operations (all of these operations are also known as edit operations) required to transform x into y. Modify the LCS algorithm in such a way so that given two strings x and y of length n and m respectively, your algorithm will output the value of $ed(x, y)$ in $O(mn)$ time. Can you also output the corresponding sequence of edit operations within the same time bound?

Question 3: Suppose a seller wants to cut a rod into several pieces and then sells them to his/her customers. The main objective of the seller is to maximize his/her earning. Given a rod of length n and a list of prices of rod of length i for all $1 \leq i \leq n$, find an optimal way to cut the rod into smaller pieces in order to maximize the earning. (Try to design a dynamic programming algorithm.)

Question 4: You are given an array of n positive integers with total sum S and a value k. Design a dynamic programming algorithm to find (if exists) k (disjoint) partitions of the input array so that the sum of numbers in each partition is S/k.

Question 5: In the lecture we have already discussed the problem of changing $\$n$ with smaller denomination coins, and designed a dynamic programming algorithm. Now consider the following greedy strategy for the same problem with coin denominations $\{Q, D, N, P\}$: suppose the amount left to change is n; take the largest coin that is no more than n; subtract this coin's value from n, and repeat.

Either give a counterexample, to prove that this algorithm can output a non-optimal solution, or prove that this algorithm always outputs an optimal solution.

Question 6: You are given n events where each takes one unit of time. Event i will provide a profit of g_i dollars ($g_i > 0$) if started at or before time t_i where t_i is an arbitrary real number. (Note: If an event is not started by t_i then there is no benefit in scheduling it at all. All events can start as early as time 0.) Also only one event can be scheduled at any particular time interval.

Give as efficient algorithm as you can, to find a schedule that maximizes the total profit.

Question 7: Given a set of keys $1, 2, \cdots, n$, where the key i has weight w_i. The weight of the key reflects how often the key is accessed, and thus heavy keys should be higher in the tree. The Optimal Binary Search Tree problem is to construct a binary-search tree for these keys, in such a way that

$$wac(T) = \sum_{i=1}^{n} w_i d_i$$

is minimized, where d_i is the depth of key i in the tree (note: here we assume the root has a depth equal to one). This sum is called the *weighted access cost (wac)*. Consider the greedy heuristic for Optimal Binary Search Tree: for keys $1, 2, \cdots, n$, choose as root the node having the maximum weight. Then repeat this for both the resulting left and right subtrees.

Apply this heuristic to keys $1, 2, 3, 4, 5$ with respective weights $50, 40, 20, 30, 40$. Show that the resulting tree does not yield the minimum weighted access cost.

Question 8: Given a set $\{x_1 \leq x_2 \leq \cdots \leq x_n\}$ of points on the real line, determine the smallest set of unit-length closed intervals that contains all of the points x_i.

Can you design an $O(n)$ time greedy algorithm for the above problem?