EECS 3101 Fall '17: Design and Analysis of Algorithms

Assignment 3

Weight: 5%, Due: Nov 17, in the drop box by 1 pm

Notes:

1. The assignment can be handwritten or typed. It MUST be legible.
2. You must do this assignment individually.
3. Submit this assignment only if you have read and understood the policy on academic honesty on the course web page. If you have questions or concerns, please contact the instructor.
4. Use the dropbox near the EECS main office to submit your assignments. No late submissions will be accepted. Please do not send files by email unless you have the instructor's permission to do so.

Problem 1

Professor X believes that a student's score in a programming course like EECS 1022 is negatively correlated with their score in a discrete mathematics course like EECS 1019. Professor Y would like to disprove this claim by showing that in his class there is a large number of students who show a positive correlation in these courses. More precisely, Professor Y would like to construct the largest possible sequence of students in his class whose scores in both courses are in increasing order. Design a dynamic programming algorithm for this problem to help Professor Y. For simplicity you can assume all scores are distinct and they need not be integers.

Note: The input is a sequence of score-pairs of the form $(EECS1022score, EECS1019score)$. Your algorithm should output the largest number of pairs (students) that can be ordered in the manner described above.

(3 points) Solution without using LCS

Design a dynamic programming algorithm for this problem and indicate how you can construct the sequence using the algorithm. You must write a recursive equation for the solution and indicate how it is used to fill a table and reach the final answer.

You need not prove optimal substructure for this problem.

(2 points) Solution using LCS

Design a solution using the solution to the LCS problem covered in class. Your solution should not define a recursive equation for the length etc., but rather transform this problem to be an instance of the LCS problem.

Problem 2

A machine for custom cutting of rods has the following cost structure. To cut a rod of length k into 2 pieces (i.e. make a single cut) of any size, the cost is proportional to k. Suppose a company buys rods of fixed length L and cuts each into n pieces of lengths $k_1, ..., k_n$ using the machine described above. Clearly, the order in which the cuts are made determines the total cost. E.g. if it wants to cut pieces of length 3,5,12 from a rod of length $L = 20$, the cost could be 20+17 or 20 +8 depending on the order of the 2 cuts needed.

Devise a dynamic programming algorithm that, given the lengths of the n pieces, determines the cheapest cost of cutting those pieces in time $O(n^3)$.

For proving correctness, you need to show optimal substructure. You must also show why the running time is $O(n^3)$.