

NTUST, CSIE

Algorithms (CS3001301), Spring 2021

Homework 4

Due date: May 28 (Friday)

Total: 10 pts

Problem 4.1. In the code of LOOKUP-CHAIN (checking the slides or p.388 from the textbook), if the first two lines of code, namely,

```
1  if  $m[i, j] < \infty$ 
2      return  $m[i, j]$ 
```

are removed, what will happen? (2pts)

Problem 4.2. Consider a variant of the matrix-chain multiplication problem in which the goal is to parenthesize the sequence of matrices so as to maximize, rather than minimize, the number of scalar multiplications. Does this problem exhibit optimal substructure? (2pts)

Problem 4.3. Determine an LCS of $\langle 1, 0, 1, 0, 0, 1, 0, 1 \rangle$ and $\langle 0, 1, 0, 1, 1, 0, 1, 1, 0 \rangle$. Moreover, as discussed in our class, generalize the recursive formula used in the textbook Eq. 15.9 (p.393) to the following one:

$$c[i, j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0, \\ \max(c[i-1, j-1] + (x_i == y_j), c[i, j-1], c[i-1, j]) & \text{if } i, j > 0, \end{cases}$$

($x_i == y_j$ is 1 if $x_i = y_j$ or 0 if not), so that all LCS's of two sequences can be found. Use the recursive formula to find all the LCS's. (2pts)

Problem 4.4. Give an $O(n^2)$ -time algorithm to find the longest monotonically increasing subsequence of a sequence of n numbers. Discuss two cases separately: (1) simple increasing subsequences (2) strictly increasing subsequences. (2pts)

Problem 4.5. Not just any greedy approach to the activity-selection problem produces a maximum-size set of mutually compatible activities. Give an example to show that the approach of selecting the activity of least duration from among those that are compatible with previously selected activities does not work. Do the same for the approaches of always selecting the compatible activity that overlaps the fewest other remaining activities and always selecting the compatible remaining activity with the earliest start time. (2pts)