Assignment 3
COSC 3P03: Design and Analysis of Algorithms
Fall 2019

Due: Nov. 18, (Monday), 5:00 PM.

What to submit:

Hard copy: for all questions, submit a hard copy including (when applicable) the definition of your cost function, recurrence, justification that the principle of optimality holds, algorithm (high level description, not actual code), analysis. test results.

Soft copy: for programming questions Q1 and Q2, submit a soft copy for each of your programs as well as test results (on Sandcastle: submit3p03).

1. (30) Let S be a sequence of n distinct integers stored in an array as array elements $S[1], S[2], \cdots, S[n]$. Use the technique of dynamic programming to find the length of a longest ascending subsequence of entries in S and print out one such a sequence. For example, if the entries of S are 11, 17, 5, 8, 6, 4, 7, 12, 3, then one longest ascending subsequence is 5, 6, 7, 12 with length 4. Please follow the general steps when designing an algorithm using the technique of dynamic programming:

- define an appropriate function;
- write the recurrence;
- show that the principle of optimality holds with respect to the function;
- design an algorithm using the recurrence;
- (programming) implement your algorithm that asks the user for input and then computes the optimal result.
- be sure to test your program on several input.

Your algorithm must run in $O(n^2)$ time. Explain why your algorithm has a complexity of $O(n^2)$.

2. (30) Implement the algorithm discussed in class to find an optimal way to multiply n matrices. Please note that you need to follow the following steps:

- ask the user for an input, n and $r_0, r_1, \cdots, r_n$ that represent the dimensions of the n matrices.
- implement the dynamic programming algorithm covered in class.
- then print out the optimal way of multiplying these n matrices. For example, if $n = 4$ and the optimal way is to multiply M_2 and M_3 first, followed by multiplying the product just obtained with M_4, followed by multiplying M_1 with the last product, you should print out $(M_1 \times ((M_2 \times M_3) \times M_4))$.

3. (20) Fill the table used by the dynamic programming method to find optimal order to multiply n matrices. Please note that you should try to do it without writing a program (a program should only be used to verify your answers): $n = 5$, $r_0 = 8$, $r_1 = 3$, $r_2 = 2$, $r_3 = 19$, $r_4 = 18$, $r_5 = 7$. **Important:** Show your work by writing down how each entry m_{ij} in the cost matrix is computed.

4. (20) Two character strings may have many common substrings. Substrings are required to be contiguous in the original string. For example, *photograph* and *tomography* have several common substrings of length one (i.e., single letters), and common substrings *ph*, *to*, and *ograph* as well as all the substrings of *ograph*. The maximum common substring length is 6. Let $X = x_1x_2 \cdots x_m$ and $Y = y_1y_2 \cdots y_n$ be two character strings. Using dynamic programming, design an algorithm to find the maximum common substring length for X and Y using dynamic programming. Please follow the general steps for designing a dynamic programming solution as in Q1 (other than the actual programming part).