2^{nd} Assignment
CS430 Introduction to Algorithm, Fall 2019

Instructor: Edward Chlebus
TA: Haohua Du, Yidan Li

October 19, 2019

This homework is due at 11:59pm on October 31, 2019

Assignment Instruction

- Team work is allowed, max 4 students per team.
- ONLY team leader submits the PDF version of the assignment to the Blackboard. You also HAVE TO include all the team members' full name and A-number in the first page of the submission.
- Late submissions won't be accepted.
- All solutions should be explained.
- Extra credits would be given for nonstandard original solutions.

Problem 1 (15pts)

Do problem 13-4 (a) and (b) on page 333 in CLRS. Justify your answer.

Problem 2 (10pts)

Consider the code for Randomized-Selection on page 216 in CLRS. Support someone carelessly implemented the code, but omitted the "-1" on line 8, that it typing q instead of q - 1.

Does the corrupted code still work (that is, correctly find the i^{th} smallest element) always, some-times, or never? Explain your answer.

Problem 3 (25pts)

Coins of various values are placed on the cells of an $n \times m$ chess board. Let the upper left corner cell be $(1, 1)$ and the lower right cell be (n, m); cell (i, j) has coins valued at c_{ij} . A robot starts at cell $(1, 1)$ and can move only to the right or down on the board.

1. Give a dynamic programming algorithm expressed recursively without memoization to determine the path the robot should follow to maximize the total value of the coins collected as the robot wanders on the board from cell $(1, 1)$ to cell (n, m). Analyze the time required and give corresponding pseudocode.

2. Give the algorithm iteratively with memoization. Analyze the time required and give corresponding pseudocode

Problem 4 (25pts)

We are given a sequence of n numbers, $a_1, a_2, \cdots, a_n$ and want to find the *longest increasing subsequence* (LIS); that is, we want to find indices $i_1 < i_2 < \cdots < i_m$ such that $a_{i_j} < a_{i_j+1}$ and m is as large as possible. For example, given the sequence 5, 2, 8, 6, 3, 6, 9, 7 we have an increasing subsequence 2, 3, 6, 9 and there is no longer increasing subsequence.

1. Give a recursive dynamic programming recurrence (just give the function) for the LIS of a sequence $a_1, a_2, \cdots, a_n$. (Hint : Let L_i be the length of the LIS in $a_1, a_2, \cdots, a_i$, let A_i be index of the smallest possible largest element in that increasing subsequence, and let B_i be index of the second-largest element in that increasing subsequence. Express L_i recursively. You may assume a dummy element $a_0 = -\infty$)

2. Give the algorithm iteratively with memoization. Analyze the time required and give corresponding pseudocode.