CS/ECE 374 A (Spring 2024)
Homework 6 (due Mar 7 Thursday at 10am)

Instructions: As in previous homeworks.

Note: In dynamic programming problems, you should follow all the steps below, unless stated otherwise:

1. first give a clear, precise definition of the subproblems (i.e., what the recursive function is intended to compute);
2. then derive a recursive formula to solve your subproblems (including base cases), with justification/explanation;
3. specify a valid evaluation order;
4. give pseudocode to evaluate your recursive formula bottom-up (using tables, and loops instead of recursion); and
5. analyze the running time;
6. and if we explicitly ask for it, give pseudocode to output an optimal solution rather than just the optimal value.

Do not jump directly to pseudocode. Do not skip step 1!

Problem 6.1: For a sequence of distinct numbers $B = \langle b_1, b_2, \ldots, b_\ell \rangle$, define its associated string $\sigma(B) = c_1 c_2 \cdots c_{\ell-1} \in \{/, \backslash\}^*$ where $c_i =$ / if $b_i < b_{i+1}$, and $c_i = \backslash$ if $b_i > b_{i+1}$.

Consider the following problem: given a sequence of distinct numbers $A = \langle a_1, a_2, \ldots, a_n \rangle$, find a subsequence $B = \langle a_{i_1}, a_{i_2}, \ldots, a_{i_\ell} \rangle$ with $i_1 < i_2 < \cdots < i_\ell$ of maximum length, such that $\sigma(B)$ does not contain /\/ nor \/\ as a substring.

(Example: For the input sequence $A = \langle 9, 13, 3, 11, 15, 4, 6, 10, 7, 1, 5, 2 \rangle$, one feasible solution is the subsequence $B = \langle 9, 3, 11, 15, 10, 7, 1, 5 \rangle$ of length ~~7~~ 8, since its string $\sigma(B) =$ \//\\\/ avoids the pattern substrings /\/ and \/\. ~~(I think this solution is optimal.)~~ (There is a better solution of length 9 (see discussion on Ed), which is optimal, probably?) On the other hand, $B = \langle 9, 3, 11, 15, 4, 10, 7, 1, 5 \rangle$ is not feasible, since $\sigma(B) =$ \//\/\\/.)

(Fun Fact: there always exists a solution of length at least $n/2$ for any $n > 7$. This was first proved in 2018; the proof is far from obvious... you don't need to know this.)

(a) (80 pts) Design and analyze an efficient dynamic programming algorithm to compute the optimal length. Aim for $O(n^2)$ running time for full credit.

(Hint: consider defining $L(i, /\backslash)$ to be the maximum length of a subsequence B of $\langle a_1, a_2, \ldots, a_i \rangle$ subject to the constraints that the last element of B is a_i, and $\sigma(B)$ ends in /\ and avoids /\/ and \/\. Define other subproblems similarly...)

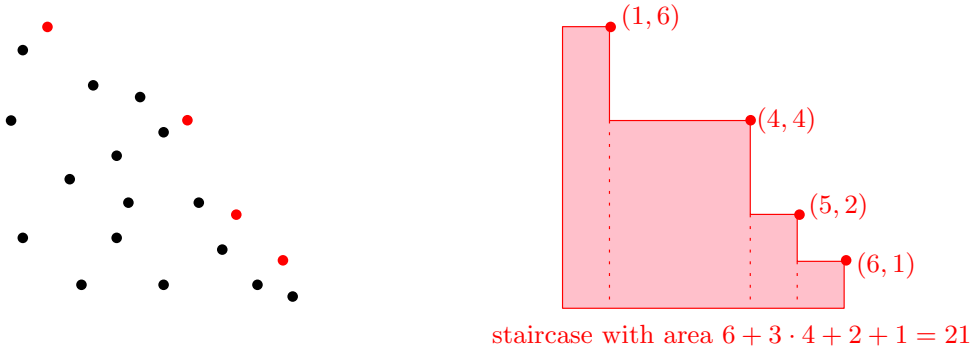
(b) (20 pts) Give pseudocode to output an optimal subsequence.

Problem 6.2: For a set Q of points with positive x- and y-coordinates in 2D, define STAIRCASE(Q) to be the region

$$\{(x, y) : \ 0 \le x \le q.x \text{ and } 0 \le y \le q.y \text{ for some } q = (q.x, q.y) \in Q\}.$$

Consider the following problem: given a set P of n points with positive distinct x- and y-coordinates in 2D, and given a number $k \le n$, find a subset $Q \subseteq P$ of at most k points such that the area of STAIRCASE(Q) is maximized.

(The higher-dimensional version of this problem is useful as a way to identify "important" points in a large dataset. Below is an example of a feasible solution Q, shown in red, for $k = 4$.)



(a) (80 pts) Design and analyze an efficient dynamic programming algorithm to compute the optimal area (you don't have to output an optimal subset). Aim for $O(n^2k)$ running time for full credit.

(Hint: Why may we assume that Q is decreasing, i.e., the points of Q are of the form $q_1, \ldots, q_\ell$ with $q_1.x < \cdots < q_\ell.x$ and $q_1.y > \cdots > q_\ell.y$?)

(b) (20 pts) Instead of maximizing the area, suppose we change the problem to finding a subset $Q \subseteq P$ that maximize the number of points of P inside STAIRCASE(Q). Describe the changes to your recursive formula (no need to give complete pseudocode again), and show that the algorithm can be implemented in $O(n^3)$ time (do not assume that k is a constant).

(Bonus: A correct solution that has running time $O(n^2k)$ or better instead of $O(n^3)$ may receive up to 5 extra points.)