Homework #2
Introduction to Algorithms
601.433/633
Spring 2020

Due on: Tuesday, February 13th, 12pm
Format: Please start each problem on a new page.
Where to submit: On Gradescope, please mark the pages for each question

February 7, 2020

1 Problem 1 (12 points)

Given a list of n integers $x_1, \ldots, x_n$ (possibly negative), find the indices $i, j \in [n]$ ($i \neq j$) such that $x_i \cdot x_j$ is maximized. Your algorithm must run in $O(n)$ time.

2 Problem 2 (12 points)

Let S be an array of integers $\{S[1], S[2], \ldots, S[n]\}$ such that $S[1] < S[2] < \cdots < S[n]$. Design an algorithm to determine whether there exists an index i such at $S[i] = i$. For example, in $\{-1, 2\}$, $S[2] = 2$.

Your algorithm should work in $O(\log n)$ time. Prove the correctness of your algorithm.

3 Problem 3 (13 points)

We say a 3-tuple of positive real numbers (x_1, x_2, x_3) is *legal* if a triangle can have sides of length x_1, x_2 and x_3. Given a list of n positive real numbers $\{x_1, \ldots, x_n\}$, count the number of unordered 3-tuples (x_i, x_j, x_k) that are legal. For example, for the numbers $\{3, 5, 8, 4, 4\}$, $(3, 4, 5)$ is a legal tuple while $(4, 4, 8)$ is not.

Your algorithm should run in $O(n^2)$ time. Prove correctness of your algorithm.

EDIT: You may give an $O(n^2 \log n)$ time algorithm and get full-credit.

4 Problem 4 (13 points)

You are given one unsorted integer array A of size n. You know that A is almost sorted, that is it contains at most m inversions, where inversion is a pair of indices (i, j) such that $i < j$ and $A[i] > A[j]$.

1. To sort array A you applied algorithm Insertion Sort. Prove that it will take at most $O(n + m)$ steps.

2. What is a maximum possible number of inversions in the integer array of size n?