CMPSCI 611 FALL '17: HOMEWORK 1
DUE 8PM, FRIDAY SEPTEMBER 22ND

- Homework may be completed in group of size with at most four students. You're not allowed to use material from the web or talk about the homework with anybody outside your collaboration group (aside from the lecturer or TA.)
- Solutions should be typed and uploaded as a pdf to `gradescope.com` (instructions to follow).
- To get full marks, answers must be sufficiently detailed, supported with rigorous proofs. Faster algorithms will typically get more marks than slower algorithms.

Question 1. Given a list of n distinct numbers $A = [a_1, a_2, \ldots, a_n]$, we say that pair a_i and a_j are *inverted* if $i < j$ but $a_i > a_j$. Let $\nu(A)$ be the number of pairs that are inverted. For example, if A is sorted in increasing order then $\nu(A) = 0$ whereas, if A is sorted in decreasing order then $\nu(A) = n(n-1)/2$. Design a divide and conquer algorithm that computes $\nu(A)$ that takes $O(n \log n)$ time. Assume that the sequence A is given in an array such that we can test whether $a_i < a_j$ or $a_j < a_i$ in unit time for any $i, j \in [n] = \{1, 2, \ldots, n\}$.

Question 2. Let A and B be two sorted lists each containing n values. For simplicity assume that n is a power of 2 and that all $2n$ elements are distinct. Design a divide and conquer algorithm for finding the nth smallest element in $A \cup B$. You need to prove that the algorithm is correct and that the running time is as claimed. The algorithm should run in $O(\log n)$ time assuming that it only take $O(1)$ time to compare two values.

Question 3. Given a list of n numbers $A = [a_1, a_2, \ldots, a_n]$, we say A is *boring* if more than $n/2$ of the values are the same. Design a divide and conquer algorithm that determines whether or not A is boring. To get full credit, the algorithm should run in $O(n)$ time assuming that it only take $O(1)$ time to compare two values.

Question 4. Let $T = (V, E)$ be a balanced binary tree with nodes V and edges E that is rooted at node r, and let $w(u)$ be the weight of node u. All weights are positive. A vertex cover is a subset U of V such that every edge has at least one endpoint in U. Give a divide and conquer algorithm to compute the weight of the minimum weight vertex cover in T. (Note that it's possible to design more efficient algorithms for this problem that aren't based on divide and conquer but, for the purposes of this homework, design a divide and conquer algorithm.)

Question 5. You are given a set of n non-vertical lines in the plane, where the i-th line, ℓ_i, is described by two real numbers a_i and b_i such that ℓ_i consists of all points (x, y) that satisfy $y = a_i x + b_i$. We say that a line ℓ dominates at t, if the y-coordinate of ℓ is larger at $x = t$ than the y-coordinate of any other line at $x = t$. We say that line ℓ is visible if there is some t at which ℓ dominates.

Design an $\Theta(n \log n)$ time algorithm that outputs a list of all the lines that are visible. Prove that the algorithm is correct and that the running time is as claimed. You may assume that no three lines all meet at a single point.