CS3100: Algorithms
Homework 1

1. You are interested in analyzing some hard-to-obtain data from two separate databases. Each database contains n numerical values, so that there are $2n$ total, and you may assume that no two are the same. You'd like to determine the median value of this set of $2n$ values, which we define to be the n^{th} value.

   However, the situation is complicated by the fact that you can only access these values through queries to the databases. In a single query, you can specify a value k to one of the two databases, and the chosen database will return the k^{th} smallest value that it contains. Since queries are expensive, you would like to compute the median using as few queries as possible.

   Given an algorithm that finds the median value using at most $O(\log n)$ queries. Be sure to specify the algorithm, the analysis for the number of queries, and a justification (i.e. a proof) that your algorithm returns the median value.

2. An *inversion* in an array $A[1..n]$ is a pair of indices (i, j) such that $i < j$ and $A[i] > A[j]$. The number of inversions in an n-element array is between 0 (if the array is sorted) and $\binom{n}{2}$ (if the array is in reverse sorted order, so that every possible pair is inverted).

   Describe and analyze an algorithm to count the number of inversions in an n-element array in $O(n \log n)$ time. [Hint: Don't reinvent the wheel! Can you modify mergesort somehow to track these?]

3. Consider an n-node complete binary tree, where $n = 2^d - 1$ for some value d, which we call the *depth* of the tree. Every node v of T is labeled with a real number x_v. (You may assume that the real numbers labeling the nodes are all distinct.) A node v of T is a *local minimum* if the label x_v is less that the label x_w for all nodes w that are jointed to v by an edge. (Note that you consider both the parent and the children here!)

   You are given a complete binary tree T, but the labeling is only specified implicitly: for each node v, you can determine the value x_v by *probing* the node v. Show how to find a local minimum of T using only $O(\log n)$ probes to the nodes of T.