CSCI 3100: Algorithms
Homework 2

Required Problems

1. Consider an n-node complete binary tree; here, by complete I mean perfectly balanced, so $n = 2^d - 1$ 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, to keep things simple.) 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 when looking for a minimum!)

   More explicitly, you are given this 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.

2. Now, do problem 32 from chapter 1 of the textbook: find a local minimum in an (unsorted) array, also in $O(\log n)$ time.

3. Call a sequence $X[1..n]$ bitonic if there is an index i with $1 < i < n$ such that the prefix $X[1..i]$ is increasing and the suffix $X[(i+1)..n]$ is decreasing. Give a simple recursive definition for the function $lbs(A)$, which gives the length of the longest bitonic subsequence of an arbitrary array A of integers.

   Note: I don't want an algorithm here, so for once you don't have to give me a running time! (It will be exponential anyway, at least to start.) Just give a simple recursive formulation, and an explanation of why your formulation will compute the correct thing. Go double check text segmentation and (especially) longest increasing subsequence for an example of what I mean by a simple recursive formulation.