1 HW #4, Extra credit: Local search

In Problems 4 and 5 on the homework, you were asked to devise algorithms to find a local minimum on complete binary trees and grid graphs. In those problems, you gave an *algorithm.* In this extra credit problem, you will try to show that those algorithms are asymptotically optimal by proving a *lower bound.*

i) For a complete binary tree on $n = 2^d - 1$ vertices, show that *any* algorithm must probe at least $\Omega(\log n)$ vertices of the tree for some input (i.e., in the worst case).

   Here's a proof strategy: Fix some correct algorithm for the problem. The algorithm has a decision procedure for which node it chooses next based on all the past probes and their answers. The algorithm can only depend on the probes it has done so far, and it must be correct on every possible input.

   Suppose the first node the algorithm chooses is v, and you return the answer $x_v = 1$. Now the node v splits the complete binary tree into two pieces A and B. One of these two pieces is larger; say A has more nodes than B. We will *plant* the local minimum in A by setting all the values in B to be increasing as we move into B away from v. In this way, at every step, we try to ensure that the algorithm can only cut off a piece of the graph which has size at most half, meaning the algorithm will have to proceed for $\Omega(\log n)$ steps before it has whittled the graph down to a single node (containing the only local minimum).

   Flesh out this argument and make it rigorous.

ii) Now try to do the following more challenging task: Prove that for the n-by-n grid, any algorithm must make at least $\Omega(n)$ queries in the worst case to find a local minimum.