

CS 161: Recitation 3 (Fall 2016)

Question 1

We are given a sorted list of values S and a target value k , and we wish to find an index i^* such that $S[i^*] = k$. Suppose our computer can only compare elements in the following way: given two values x and y , we can tell whether $x > y$, $x < y$, or $x = y$ in constant time.

Argue that in this *comparison-based model of computation*, any algorithm for finding i^* must take $\Omega(\log n)$ time.

Question 2

Random sort, also known as bogosort, attempts to sort a list by randomly permuting all the elements and then checking if the list is sorted. It repeats this process until it successfully generates the sorted list. In this problem, assume that the operation of randomly permuting n elements takes $O(n)$ time, and assume that the list we are attempting to sort consists of distinct integers.

What is the expected running time of the algorithm for worst-case input? Is the running time of the algorithm bounded?