COL 351 Aug 21-25, 2023

TUTORIAL SHEET 4

1. We have n nuts and n bolts. The nuts (and the bolts) are of different sizes. Each bolt fits in exactly 1 nut. We would like to match the nuts with the bolts which fits into them. Since the dimensions of the nuts and the bolts are so small, we can not really tell if a nut (or a bolt) is bigger than another nut (or bolt). So the only operation that is allowed is comparing a nut and a bolt – with such a comparison we can distinguish between three cases, namely, the nut fits a larger bolt, or the nut fits a smaller bolt, or the nut fits this bolt. Give a randomized algorithm which matches nuts with bolts. The expected number of comparisons (of a nut with a bolt) done by this algorithm should be $O(n \log n)$.

2. **[KT-Chapter5]** We are interested in analyzing some hard to obtain data from two databases. Each database contains n numerical values (so there are $2n$ values in total). Assume that these values are distinct. We would like to determine the median of these $2n$ values, which we define as the n^{th} smallest value. However, the only way to access these values is through queries to the databases. In a single query, we specify a value k to one of the two databases, and the chosen database returns the k^{th} smallest value that it contains. Give an algorithm which finds the median value using $O(\log n)$ queries only.

3. Given an array of n objects, you need to decide if there is an object which is present more than $n/2$ times. The only operation by which you can access the objects is a function f, which given two indices i and j, outputs whether the objects at positions i and j in the array are identical or not. Given an $O(n \log n)$-time algorithm for this (where each call to f is counted as 1 operation).

4. (**Dasgupta, Papadimitriou, Vazirani Chapter 2**)You are given an infinite array $A[]$ in which the first n cells contain integers in sorted order and the rest of the cells are filled with ∞. You are not given the value of n. Describe an algorithm that takes an integer x as input and finds a position in the array containing x, if such a position exists, in $O(\log n)$ time.

5. (**Jeff Erickson**) Suppose we are given an array $A[1...n]$ with the special property that $A[1] \geq A[2]$ and $A[n-1] \leq A[n]$. We say that an element $A[x]$ is a local minimum if it is less than or equal to both its neighbors, or more formally, if $A[x-1] \geq A[x]$ and $A[x] \leq A[x+1]$. We can obviously find a local minimum in $O(n)$ time by scanning through the array. Describe and analyze an algorithm that finds a local minimum in $O(\log n)$ time. It is possible that the array has many local minima, you are required to find any one.