Divide and Conquer I

1. 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. Give an $O(n \log n)$-time divide-and-conquer algorithm for this (where each call to f is counted as 1 operation).
2. (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.
3. [KT-Chapter 5] You are given n nonvertical lines in the plane, labeled $L_1, L_2, \dots L_n$, with the i^{th} line specified by the equation $y = a_i x + b_i$. We will make the assumption that no three of the lines all meet at a single point. We say line L_i is uppermost at a given x-coordinate x_0 if its y-coordinate at x_0 is greater than the y-coordinates of all the other lines at x_0: $a_i x_0 + b_i > a_j x_0 + b_j$ for all $j \neq i$. We say line L_i is visible if there is some x-coordinate at which it is uppermost--intuitively, some portion of it can be seen if you look down from $y = \infty$. Give an algorithm that takes n lines as input and in $O(n \log n)$ time returns all of the ones that are visible.
4. [KT-Chapter 5] Consider a n-node complete binary tree T, where $n = 2^d - 1$ for some d. Each node v of T is labelled with a real number x_v. You may assume that the real numbers labelling the nodes are all distinct. A node v in T is a local minimum ff the label x_v is less than the label x_w for all nodes w that are joined to v by an edge. You are given such a complete binary tree T, but the labeling is only specified in the following implicit way: 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.
5. [KT-Chapter 5] Suppose now that you're given a $n \times n$ grid graph G. (A $n \times n$ grid graph is just the adjacency graph of a $n \times n$ chessboard. To be completely precise, it is a graph whose node set is the set of all ordered pairs of natural numbers (i, j), where $1 \leq i \leq n$ and $1 \leq j \leq n$; the nodes (i, j) and (k, l) are joined by an edge iff and only iff $|i - k| + |j - l| = 1$. We use some of the terminology of the previous question. Again, each node v is labelled by a real number x_v; you may assume that all these labels are distinct. Show how to find a local minimum of G using only $O(n)$ probes to the nodes of G. (Note that G has n^2 nodes.)
6. You are given a sequence of n numbers $a_1, a_2, \dots, a_n$. Find $i, j,\ i \leq j$ such that the sum $a_i + a_{i+1} + \dots + a_j$ is maximum. Note that the numbers are not all positive.