1. (10 points) Define **majority element** as one which occurs (strictly) more than \( \frac{n}{2} \) times in an array. We want to find if an input array (unsorted) has such an element and find the element if it exists. Describe a divide-and-conquer algorithm to do this using only comparisons that decide if an element is equal to or bigger or smaller than another element. What happens if we allow only comparisons of the type "Is \( A[i] = A[j] \)"? (For example, if the array does not consist of numbers but objects this will happen). Does the worst case complexity remain the same? Show all the recurrence relations that arise and their solution.