Homework-2: Selection Problem

Your Name Student ID: 19100000

With the following set of problems we will develop a divide and conquer based algorithm for the general problem of selection of an order statistics of an array.

The $\textsc{Select}(A, k)$ problem asks to find the k smallest element of an array, i.e. an element x which will be at the kth position when A is sorted (or there are $k-1$ numbers in A smaller than x). More formally,

Input An array A of n distinct integers and an integer k.
Output A number $x \in A$, such that $rank_A(x) = k$.

This is a general problem in the sense that many order statistics or ordering problems can be solved with this.

Problem 1. Briefly, describe an algorithm to find the minimum, median, maximum of an array A using a subroutine for $\textsc{Select}(A, k)$.

Problem 2. Briefly, describe an algorithm to sort an array A using a subroutine for $\textsc{Select}(A, k)$.

Divide the array A into $\frac{n}{5}$ groups of 5 successive elements of A. All the groups are of size 5 except for possibly the last one.

Problem 3. Show that the medians for all the groups can be found in total time $O(n)$

Problem 4. Let MoM be the median of medians of all groups. Argue that MoM is not necessarily the median of A. Construct an example array A of 15 numbers for which MoM is not the median. Also construct an example where MoM is the median.

Problem 5. How many elements of A are necessarily smaller (larger) than MoM?

So far we have identified one element of the array, that is reasonably in the middle. Meaning if we partition the array around MoM (all elements smaller than MoM to the left and larger to the right), then both sides will be reasonably small, at most 70% of A. Therefore, we should recursively solve the $\textsc{Select}(B, k')$ subproblem for appropriate B and k'. There are two challenges however, a) We have only identified/defined the MoM, we need an algorithm for that, b) Partitioning A around MoM will take some computational time.

Problem 6. Suggest an algorithm to find MoM of A.
Hint: Make a call to an algorithm for a known problem with appropriate parameters.

Problem 7. Give a linear time algorithm for the problem of partitioning A around MoM.

This leads to a divide and conquer based strategy to solve the $\textsc{Select}(A, k)$ problem recursively.

Problem 8. Give a detailed and precise algorithm that implements the above idea.

Problem 9. Let $T(n)$ be the number of comparisons that your algorithm performs on array A of size n. Derive a recurrence relation for $T(n)$. Prove that $T(n) = O(n)$.

Problem 10. Suppose in the above solution instead of subdividing into groups of size 5, we divide the array into groups of size 7. Derive and prove the recurrence for the runtime of this modified.

Problem 11. Earlier we showed that the runtime of QuickSort depends upon the quality (rank) of the pivot in the corresponding subarray. Suppose we always select pivot by selecting the median of the subarray using the above algorithm. for median. Show that the runtime of this 'modified' algorithm is $O(n \log n)$.

Problem 12. Earlier we showed that if we have a solution for Select(A, k) problem, then we readily get a solution for the Median(A) problem. Suppose we have a linear time solution for Median(A) problem (not based on Select(A, k)), design a linear time algorithm to solve the Select(A, k) problem using the claimed solution