Algorithms

Assignment: Order Statistics

Name: .................................................................

Id: .................................................................

Grade

Good Luck!
1. Let $A = [A_1, A_2, \ldots, A_n]$ be an unsorted array of $n \geq 1$ positive integers. Design an efficient algorithm that finds the maximum difference between any two integers in the array. In other words, compute $M = \max_{1 \leq i,j \leq n} \{A_i - A_j\}$.

What is the exact worst-case number of comparisons made by your algorithm? If you don’t know the exact complexity you may use the “$O, \Omega, \Theta$” notation.

Justify the correctness and complexity of your algorithm.
2. Let $A = [A_1, A_2, \ldots, A_n]$ be an unsorted array of $n \geq 4$ distinct integers. Design an efficient algorithm that finds the first, second, and third largest integers in $A$. What is the worst case number of comparisons made by your algorithm?

Use words to describe the algorithm. Try to find the exact or almost exact number of comparisons. Ignore floors and ceilings.

Justify the correctness and complexity of your algorithm.
3. Design an algorithm that finds the median of 5 distinct keys with at most 6 comparisons. Justify the correctness of your algorithm.
4. For an odd $n \geq 1$, let $A = [A_1, A_2, \ldots, A_n]$ be an unsorted array of $n$ positive integers that are not necessarily distinct. A majority is an integer that appears at least $(n + 1)/2$ times in the array. Design a linear-time algorithm that finds a majority in $A$ if exists.

*Hint:* Note that there can be at most one majority.

Justify the correctness and complexity of your algorithm.
5. Let $A = [A_1 < A_2 < \cdots < A_n]$ be a sorted array of $n \geq 1$ distinct positive integers and let $k$ be a positive integer. Design a linear time algorithm that finds, if exist, two indices $1 \leq i, j \leq n$ such that $A_i + A_j = k$.

Justify the correctness and complexity of your algorithm.