Algorithms

Assignment: Order Statistics

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

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

Grade

Good Luck!
1. Let \(A[1], A[2], \ldots, A[n]\) be an array containing \(n\) positive integers.

Describe an efficient algorithm to find 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 complexity of your algorithm? Explain. If you know the exact complexity write it. Otherwise, you may use the “\(O, \Omega, \Theta\)" notation.
2. Let $A = A[1], \ldots, A[n]$ be an array of $n \geq 4$ distinct keys.

Describe an efficient algorithm to find the three smallest keys in $A$. Use words.

What is the worst case number of comparisons performed by your algorithm. Try to find an exact number. Ignore floors and ceilings. Explain your answer.
3. Design an efficient algorithm to find the median of 5 distinct keys.
4. Let $A$ be an array containing $n$ very large positive integers not necessarily distinct. A *majority* is a number that appears at least $\lceil(n + 1)/2\rceil$ times in the array (note that there can be at most one majority). Describe an $O(n)$-time algorithm that finds a majority in $A$ *if exists*. Explain why your algorithm has time complexity $O(n)$. 
5. Let $A = [A_1 < A_2 < \cdots < A_n]$ be a sorted array containing $n$ distinct positive integers and let $k$ be a positive integer. Describe an $O(n)$-time algorithm that finds if exist two indices $1 \leq i, j \leq n$ such that $A_i + A_j = k$. Explain why your algorithm has time complexity $O(n)$. 
6. Let $A = [A_1 < A_2 < \cdots < A_n]$ be a sorted array containing $n$ distinct negative and positive integers. Describe an $O(\log(n))$-time algorithm that finds if exists an index $1 \leq i \leq n$ such that $A_i = i$. Explain why your algorithm has time complexity $O(\log(n))$. 