Algorithms

Assignment: Order Statistics and Sorting

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

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

<table>
<thead>
<tr>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

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

   Describe an efficient algorithm to find the minimum positive difference between any two integers in the array.

   What is the complexity of your algorithm? Explain.
   Describe an efficient algorithm to find the maximum difference between any two integers in the array.
   What is the complexity of your algorithm? Explain.
3. Design an efficient algorithm to find the median of 5 distinct keys.
4. Design an efficient algorithm to sort 5 distinct keys.
5. 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.
6. 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 $O(n)$. 


7. An array $A$ of $n \geq 1$ distinct positive integers is *bitonic* if there exists an index $1 \leq i \leq n$ such that:


Describe an algorithm that uses a linear number ($O(n)$) of comparisons to sort bitonic arrays with very large integers (you *cannot* use radix-sort like algorithms). Note that if the array is bitonic the index $i$ is not known. Justify your answer.