

Algorithm Design and Analysis-CS403

Programming Assignment

Total: 15 Marks.

1 1st Programming Assignment:

1. Given an array of n elements find its maximum and minimum elements using divide and conquer approach. The number of comparison step in the algorithm should be $(\frac{3n}{2} - 2)$. 2 Marks.
2. Given an array of n elements find its minimum and second minimum elements using divide and conquer approach. The number of comparison step in the algorithm should be $(n + \log n - 2)$. 2 Marks.
3. Given an array of n elements find median using divide and conquer approach. The running time complexity of the algorithm should be $O(n)$. 3 Marks.
4. Write an algorithm for quick sort that uses the algorithm for computing median in $O(n)$ time (stated in Question 3) as a subroutine, and give the sorted array in $O(n \log n)$ complexity. 2 Marks.
5. Write an algorithm for merge-sort that give the sorted array in $O(n \log n)$ complexity. 2 Marks.
6. An array $A[1 \dots n]$ is said to have a majority element if more than half of its entries are the same. Given an array, the task is to design an efficient algorithm to tell whether the array has a majority element, and, if so, to find that element.
 - Show how to solve this problem in $O(n \log n)$ time. (*Hint: Split the array A into two arrays A_1 and A_2 of half the size. Does knowing the majority elements of A_1 and A_2 help you figure out the majority element of A ? If so, you can use a divide-and-conquer approach.*) 2 Marks.
 - Can you give a linear-time algorithm? (*Hint: Here's another divide-and-conquer approach: – Pair up the elements of A arbitrarily, to*

get $n/2$ pairs; Look at each pair: if the two elements are different, discard both of them; if they are the same, keep just one of them. Show that after this procedure there are at most $n/2$ elements left, and that they have a majority element if and only if A does.).
2 Marks.