

Course/ Batch: BTech/ SCSET
Course Code: CSET244
Semester: Even
Session: 2024-2025

Course Type: Core
Course Name: Design and Analysis of Algorithms

Tutorial Assignment: Week 5

Tutorial title: Divide and Conquer

Q1. Step 1: Sort the given array using merge sort algorithm

Step 2: current element = Assign the first element of the sorted array into the first cell of the new array
/* the new array contains elements without duplicates */

Step 3: Traverse the sorted array from the second index up to the last index /* single pass traversal */
3.1 if the current element is the same as the previous one
 skip it.
3.2 else
 copy it to the new array
 update the current element

Merge sort time complexity is $O(n \log n)$.

Time complexity for removing duplicate from the array is $O(n)$.

Time complexity of total process is $O(n \log n) + O(n) \approx O(n \log n)$.

Q2. Step 1: Sort the Array using Merge sort in $O(n \log n)$

/* If an element appears more than $n/2$ times, it must occupy a contiguous block in the sorted array. */

Step 2: Scan the sorted array and check each element whether it appears $> n/2$. /* Single pass scan */

2.1: if such an element found
 return that element

2.2 else
 print("There exists no such element that appears more than $n/2$ times")

Merge sort time complexity is $O(n \log n)$.

Time complexity for linear search is $O(n)$.

Time complexity of the entire process is $O(n \log n) + O(n) \approx O(n \log n)$.

Q3. $n \geq 64$

Q4. $O(n^2 \log n)$

Q5. Zero