

Divide and conquer method : AlgoLab 3

- Note: (1) You have to use only divide and conquer method to solve following program.
(2) In final exam how to find time complexity (using recurrence relation, master's theorem etc) will be part of algorithm lab.

1. Given a sorted array in which all elements appear twice (one after one) and one element appears only once. Find that element in $O(\log n)$ complexity.
Example: Input: $arr[] = \{1, 1, 3, 3, 4, 5, 5, 7, 7, 8, 8\}$
Output: 4
2. Given an integer x , Write a C or a C++ program to find square root of it in $O(\log(n))$. If x is not a perfect square, then return $\text{floor}(x)$.
3. Let $X[1..n]$ and $Y[1..n]$ be two given arrays, each containing n numbers already in sorted order. Write a C or C++ program to find the median of all the $2n$ elements in arrays X and Y in $O(\log n)$ time.
4. Let $A[0 : n - 1]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an inversion of A . Write a C or C++ program that determines the number of inversions in any permutations of n elements in $O(n \lg n)$ worst case time. (CLRS 39)
5. Write a C or a C++ program to find i^{th} smallest number in a given array using Randomized Select Algorithm. (CLRS 186)
6.
 - Write a C or a C++ program to implement Randomized Quicksort. (Randomize the Quicksort algorithm by using a random number generator. Instead of always using $A[p]$ (the first element of the subarray) as the pivot, choose randomly element from the subarray $A[p .. r]$. You can do it by exchanging element $A[p]$ with an element chosen at random from $A[p .. r]$).
 - Write a C or a C++ program to implement Randomized Quicksort by removing tail recursion from Quicksort. (Implement Randomized Quicksort algorithm by avoiding the second recursive call to the Quicksort function).

Note: As you know first algorithm takes more time, second less than first . To realise it use array sizes of 100, 200, . . . , 500 integers. For each size, run the same code say 10,00 times inside a loop for different inputs and estimate the time taken by the program for one run.