BCSE204L- Design and Analysis of Algorithms
Practice Problem Sheet 2
(Divide-Conquer-Combine Strategy)

Practice makes you Perfect

Inputs required for PPS2:

- Chapter 3 of CLRS book

Design of Algorithm

- Understand the problem
- Develop the logic to solve the problem
- Validate the logic(developed by you) with simple illustration.
- Write the pseudocode

Analysis of Algorithm

- Proof of Correctness (PoC) of the algorithm
- Computation of running time $T(n)$
- Best-case, average-case running and worst-case
- Time-complexity of the algorithm- Expression of asymptotic growth of $T(n)$ with $\vartheta, O, \Omega, o, \omega$ notations

Note: Every question should answer the 'design' and 'analysis' component of the algorithm. 'Design' component should involve all the steps described as above. 'Analysis' component should involve all the steps described as above.

1. Design a DCC based algorithm to compute the product of two complex numbers *aibi* and $c + id$, and analyse the same.
2. Given an array of integers A, Design a DCC based algorithm to compute the maximum integers in the array and analyse them.

3. In a ‘Maximum subarray Problem’, you are given an array A of integers and the task is to find the nonnegative contiguous subarray of A such that the sum of the values in the subarray is maximum. Here, there is no restriction on the size of the subarray that has maximum sum. Based on this, we propose a new problem : ‘Maximum value Maximal subarray Problem’ where in the array of non-negative integers is taken as an input and the task is to return the non-negative contiguous subarray whose sum is maximum and the length of the subarray is minimal. Design a pseudocode for the Maximum value Maximal subarray Problem.
4. Q.no.1, discusses the ‘Maximum value Maximal Subarray Problem’. On the same lines,

   (a) Design a pseudocode for ‘Maximum value Minimal Subarray Problem’.
   (b) Design a pseudocode for ‘Minimum value Minimal Subarray Problem’.
   (c) Design a pseudocode for ‘Minimum value Maximal Subarray Problem’.

5. In a ‘Maximum subarray Problem’, you are given an array A of integers and the task is to find the nonnegative contiguous subarray of A such that the sum of the values in the subarray is maximum. Here, the subarray returned by the code is ‘contiguous’ (cells of the subarray should be adjacent). Based on this, we propose a new problem : ‘Maximum non-contiguous subarray Problem’ where in the array of non-negative integers is taken as an input and the task is to return the non-negative non-contiguous subarraay whose sum is maximum and the subarray need not be contiguous.
6. Consider an array of $A[1, 2, 3, ..., n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then we call the the pair (i, j) as an inversion of A. For example, the five inversions in the array $A :< 2, 3, 8, 6, 1 >$ are $(1, 5), (2, 5), (3, 4), (3, 5), (4, 5)$. Given an array of numbers, design the pseudocode and the corresponding code that follows ‘Divide-Conquer-Combine’ strategy for computing the number of inversions in the array and analyse the same.

7. Given two square matrices of same size, Design an DCC based algorithm to compute the product of two matrices and analyse the same.
8. Merge-sort algorithm (discussed in the class) works by partitioning the input array A recursively into two halves. Here, the partition is based on the position in the array. Instead, design a new algorithm A' where partioning is based on the values in the input array. Compare the performance of A' with that of A.
9. Given n 2-dimensional points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, Design an algorithm that follows the 'Divide-Conquer-Combine' startegy to
   (a) arrange the $n-points$ in an increasing order of the x coordinates
   (b) arrange the $n-points$ in an increasing order of the y coordinates
   (c) arrange the $n - points$ in decreasing order of the value $(x - coordinate + y - coordinate)/2)$ coordinates