Divide and conquer method : AlgoLab 3 and Lab 4

Britishers have used divide and conquer method to rule India. It is also used everywhere but you should not use this method except to solve following problems.

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 reccurence relation, master's theorm etc) will be part of algorithm lab.

Lab 3

1. Write a C or a C++ program to find i^{th} Fibonacci number. Make sure that it takes exponential time.

2. Write a C or a C++ program to search an element using Binary search.

3. 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

4. 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 floor(x).

5. 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 $O(logn)$ time.

6. Write a C or a C++ program to sort the given array using Merge sort.

7. 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)

8. Write a C or a C++ program to find i^{th} smallest number in a given array using Randomized Select Algorithm. (CLRS 186)

9. - Write a C or a C++ program to implement Quicksort.
   - 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 and third the least. 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.

Lab 4

10. Write a C or a C++ program to implement Strassen matrix multiplication. It should work for multiplication of two 4×4 matrices.

11. Write a C or a C++ program to implement Karatsuba's integer multiplication. Assume two large n-digit numbers x and y, represented in base B where B is a convenient value to shift with, like 2 in contemporary computers based on the binary system. Choosing a number m less than n, we can write x and y as,
    $x = x_1B^m + x_2,\ \ y = y_1B^m + y_2$ where x_2 and y_2 are less than B^m .
    The product xy can be written as, $xy = (x_1B^m + x_2)(y_1B^m + y_2) = x_1y_1B^{2m} + (x_1y_2 + x_2y_1)B^m + x_2y_2$ The standard method is to multiply the four subproducts separately, then finally shift and add; this can be done in time $O(n^2)$ as done in the class. However, it can be observed that computation of xy can be done in only three multiplications: Let
    $X = x_1y_1,\ Y = x_2y_2$
    Let $Z = (x_1 + x_2)(y_1 + y_2) - X - Y$
    Then, $Z = (x_1y_1 + x_1y_2 + x_2y_1 + x_2y_2) - x_1y_1 - x_2y_2 = x_1y_2 + x_2y_1$
    Thus, $xy = XB^{2m} + Y + ZB^m$ To compute these three products of m-digit numbers, we can recursively employ the above multiplication scheme again. Note that the shifts, additions and subtractions take linear time only. It should Write a code for the above algorithm to multiply two given 52 digit numbers.

12. Write a C or a C++ program to find the closest pair of points in a plane in $O(nlog\ n)$ time. Given set of points $P = p(1), p(2), p(3), \ldots, p(n), n >= 2$ where $p(i) = \{x(i), y(i)\}$, find the closest pair of points. Here, closest refers to the Euclidean distance. (Given in CLRS, NPTEL notes(chpter 9) or google it)