LAB3:Divide and Conquer

Lab. Exercises (LE)

LE3.1 Write a program to recursively implement Binary Search using divide and conquer method. Determine the time required to search an element in an array of n integers. Repeat the experiment for different values of n, the number of elements in the list to be searched and plot a graph of the time taken versus n. The n integers can be generated randomly.Make sure your array is sorted before applying binary search algorithm.

LE3.2 Write a program to sort a given set of elements using the Merge sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

LE3.3 Write a program to sort a given set of elements using the Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

Note:Follow the below table for Questions LE3.2,3.3,HE 3.1,3.3

Sl. No.	Value of n	Time Complexity (Data in Ascending)	Time Complexity (Data in Descending)	Time Complexity (Random Data)
1	5000			
2	10000			
3	15000			
4	20000			
5	25000			
6	30000			
7	35000			
8	40000			
9	45000			
10	50000			

Home Exercises (HE)

HE3.1 Rewrite the program no-2.1 **(Insertion Sort)** with the following details.

i. Compare the best case, worst case and average case time complexity with the same data except time complexity will count the cpu clock time.

ii. Plot a graph showing the above comparison (n, the input data Vs. CPU times for best, worst & average case)

iii. Compare manually program no-2.1 graph vs program no-3.1 graph and draw your inference.

HE3.2 Write a program to use divide and conquer method to recursively implement and to find the maximum and minimum in a given list of n elements.

HE3.3 Write a program to sort a given set of elements using the Randomized Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.