Aim: Write a program in C to sort given list of numbers in ascending order using Merge Sort algorithm and also analyze the complexity of the algorithm.

Objective of the Experiment:
1. Understanding Merge sort algorithm.
2. Analyzing the complexity of the algorithm.

Example:
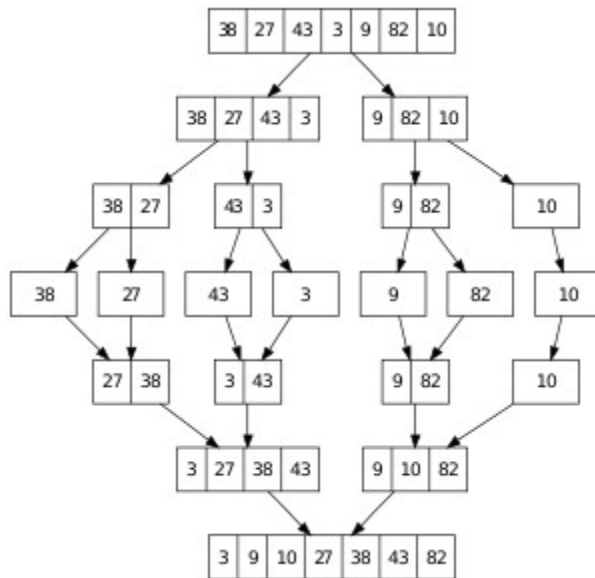
Theory:

Recursive algorithm used for merge sort comes under category of Divide and Conquer. Recursively divide the list in half until each sub-list has one element, then recombine or merge to produce final sorted array.

Algorithm mergeSort (int[] a, int i, int j)
    1. If (i < j) where, i is lower bound and j is upper bound
        1.1. Find mid k;
        1.2. call mergeSort recursively for left side [(a, i, k)];
        1.3. call mergeSort recursively for right side [(a, k+1, j)];
        1.4. call merge (a, i, k, j)

Algorithm merge(int[] a, int l, int m, int u)

        // one integer array c is used for mergeing.
        // first array from l to m
        // second array from m+1 to u
        // merge in c and after merging copy back to array a.
    1. i=l;
    2. j=m+1;
    3. k=0;

4.   if (i <= m && j <= u)

    4.1. if (a[i] < a[j])
        4.1.1.   c[k]=a[i];
        4.1.2.   Increment k and i
    4.2. else
        4.2.1. c[k]=a[j];
        4.2.2. Increment k and j
        4.3. Go back to 4.
5.   if (i <= m)
        5.1. c[k]=a[i];
        5.2. Increment k and i
        5.3. go back to 5.
6.   if (j <= u)
        6.1. c[k]=a[j];
        6.2. Increment k and j
        6.3. go back to 6.
7.   for (i=l,j=0;i<=u;i++,j++)
    copy elements of array c into array a.

.

Post Lab Assignment:
1.   Derive complexity of Merge Sort algorithm.
2.   What is space complexity of Merge sort.