Design and Analysis of Algorithms (CST 226-2)

Laboratory Assignment 01(UWU/CST/15/058)

Q1

1. **Divide and Conquer**

   a. Breaking it into sub problems that are themselves smaller instances of the same type of problem.(The divide step just computes the middle of the Subarray,which takes constant time D(n)=O(1)).

   b. Recursively solving these sub problems.(each of size n/2 ,which contributes 2T(n/2) to the running time).

   c. Appropriately combining their answer. We have already that the MERGE procedure or an n element subarray takes time O(n).

2. Algorithm Merge_Sort();

```
/*a[0:n-1] is an array of n elements.*/
MergeSort (a, beg, end)
Begin
        If beg< end then
                Set mid = (beg+end)/2;
                Call Merge (a, beg, mid);
                Call Merge (a,mid+1, end);
                Call MergeSortedArray(a,beg,mid+1,end)
        Endif
End
```

Q2

1. **Algorithm: Algorithm for quicksort Algorithm A[]is an array and p and r are variables**

```
Quicksort(A,p,r) {//Sorting Algorithm
        if (p < r) {
                q <- Partition(A,p,r)
                Quicksort(A,p,q)
                Quicksort(A,q+1,r)
        }
 }
 Partition(A,p,r)  //Partition Algorithm
         x <- A[p]
         i <- p-1
         j <- r+1
         while (True) {
                repeat
                        j <- j-1
                until (A[j] <= x)
                repeat
                        i <- i+1
                until (A[i] >= x)
                if (i A[j]
                else
        return(j)
 }
}
```

2. **Pivot** is an element which divides the array into two halves in such a way that elements in the left half are smaller than pivot and elements in the right are greater than **pivot**.

   Pick a **pivot** element:
   It has the many different versions of QuickSort that pick a pivot in different ways.

   I. Always pick first element as pivot
   II. Always pick the last element as pivot
   III. Pick a random element
   IV. Pick a median as pivot