

LAB 8- MERGE SORT

Anitej Prasad

1BM19CS194

4-D

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void merge(int arr[], int l, int m, int r)
{
    int i, j, k;
    int n1 = m - l + 1;
    int n2 = r - m;

    int L[n1], R[n2];

    for (i = 0; i < n1; i++)
        L[i] = arr[l + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[m + 1 + j];

    i = 0;
    j = 0;
    k = l;
    while (i < n1 && j < n2) {
```

```
    if (L[i] <= R[j]) {  
        arr[k] = L[i];  
        i++;  
    }  
    else {  
        arr[k] = R[j];  
        j++;  
    }  
    k++;  
}
```

```
while (i < n1) { //Copy the remaining elements of L[]  
    arr[k] = L[i];  
    i++;  
    k++;  
}
```

```
while (j < n2) { //Copy the remaining elements of R[]  
    arr[k] = R[j];  
    j++;  
    k++;  
}  
}
```

```

void mergeSort(int arr[], int l, int r)
{
    if (l < r) {

        int m = l + (r - l) / 2;

        mergeSort(arr, l, m); // Sort first half
        mergeSort(arr, m + 1, r); // Sort second half

        merge(arr, l, m, r); //merge the sorted halves
    }
}

```

```

void generateArray(int arr[],int n) {
    for(int i=0;i<n;i++)
        arr[i] = (rand()%100) - 50; // -50 to 50
}

```

```

void test(int arr[], int n) {
    int i=0;
    for(i=0;i<n-1;i++) {
        if(arr[i]>arr[i+1])
        {
            printf(" ERROR \n");
            return;
        }
    }
}

```

```

    }
}
printf(" OK \n");
}

void main() {
    srand(time(NULL));
    printf("-----Merge Sort-----\n\n");

    int n =0;
    int s,e,step;
    printf("Enter start length : ");
    scanf("%d",&s);
    printf("Enter end length : ");
    scanf("%d",&e);
    printf("Enter step length : ");
    scanf("%d",&step);

    for(int i=s;i<=e;i=i+step) {
        int arr[i];
        generateArray(arr, i);
        double res;
        clock_t start, end;
        start=clock();
        mergeSort(arr,0,i-1);
        end=clock();
    }
}

```

```
    res = ((double) (end - start)) / CLOCKS_PER_SEC;
    printf("\nn = %d\nTime taken by merge sort : %lf",i,res);
    test(arr, i);
}
printf("\n\n");
}
```