ADA LAB **Kunal Sachdeva**

06-04-2021 **19115045**

4th semester CSE

MERGE SORT

```
#include <bits/stdc++.h>
using namespace std;
void merge(int arr[], int l, int m, int r)
{
        int n1 = m - l + 1;
        int n2 = r - m;
        int L[n1], R[n2];
        for (int i = 0; i < n1; i++)
                L[i] = arr[l + i];
        for (int j = 0; j < n2; j++)
                R[j] = arr[m + 1 + j];
        int i = 0;
        int j = 0;
        int 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)
        {
                arr[k] = L[i];
                i++;
                k++;
        }
        while (j < n2) {
                arr[k] = R[j];
                j++;
                k++;
        }
}
void mergeSort(int arr[],int l,int r){
        if(l>=r){
                return;
        }
        int m =l+ (r-l)/2;
        mergeSort(arr,l,m);
        mergeSort(arr,m+1,r);
        merge(arr,l,m,r);
}
void printArray(int A[], int size)
{
        for (int i = 0; i < size; i++)
                cout << A[i] << " ";
}
int main()
{
        int arr[] = {13,10,12,89,1,3,5,1,56};
        int arr_size = sizeof(arr) / sizeof(arr[0]);
        cout << "Given array is \n";
        printArray(arr, arr_size);
        mergeSort(arr, 0, arr_size - 1);
        cout << "\nSorted array is \n";
        printArray(arr, arr_size);
        return 0;
}
```

Time Complexity:

In sorting n objects, merge sort has an average and worst-case performance of O(n log n).

```
#include <bits/stdc++.h>
using namespace std;
void merge(int arr[], int l, int m, int r)
{
    int n1 = m - l + 1;
    int n2 = r - m;
    int L[n1], R[n2];
    for (int i = 0; i < n1; i++)
        L[i] = arr[l + i];
    for (int j = 0; j < n2; j++)
        R[j] = arr[m + 1 + j];
    int i = 0;
    int j = 0;
    int 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)
    {
        arr[k] = L[i];
        i++;
        k++;
    }
    while (j < n2) {
        arr[k] = R[j];
        j++;
        k++;
    }
}
void mergeSort(int arr[],int l,int r){
    if(l>=r){
        return;
    }
    int m =l+ (r-l)/2;
    mergeSort(arr,l,m);
    mergeSort(arr,m+1,r);
    merge(arr,l,m,r);
}
```

```
void printArray(int A[], int size)
{
    for (int i = 0; i < size; i++)
        cout << A[i] << " ";
}
int main()
{
    int arr[] = {13,10,12,89,1,3,5,1,56};
    int arr_size = sizeof(arr) / sizeof(arr[0]);
    cout << "Given array is \n";
    printArray(arr, arr_size);
    mergeSort(arr, 0, arr_size - 1);
    cout << "\nSorted array is \n";
    printArray(arr, arr_size);
    return 0;
}

```

Success #stdin #stdout 0s 5504KB comments (0)

stdin copy

```
Standard input is empty
```

stdout copy

```
Given array is
13 10 12 89 1 3 5 1 56
Sorted array is
1 1 3 5 10 12 13 56 89
```