

Merge Sort

Medium Accuracy: 54.1% Submissions: 112K+ Points: 4

Participate In Monthly Hiring Challenge conducted by GeeksforGeeks !!

Given an array arr[], its starting position l and its ending position r. Sort the array using merge sort algorithm.

Example 1:

Input:

N = 5

arr[] = {4 1 3 9 7}

Output:

1 3 4 7 9

Example 2:

Input:

N = 10

arr[] = {10 9 8 7 6 5 4 3 2 1}

Output:

C++ (g++ 5.4)

Average Time: 15m

Start Timer

```
1 // } Driver Code Ends
19 class Solution{
20 public:
21 void merge(int arr[], int l, int m, int r) {
22     int i=l , j=m+1 , k=0 , temp[r-l+1];
23     while(i<=m && j<=r){
24         if(arr[i]<=arr[j]){
25             temp[k]=arr[i];
26             i++;k++;
27         }else{
28             temp[k]=arr[j];
29             j++;k++;
30         }
31     }if(i>m){
32         while(j<=r){
33             temp[k]=arr[j];
34             j++;k++;
35         }
36     }else{
37         while(i<=m){
38             temp[k]=arr[i];
39             i++;k++;
40         }
41     }for(int i=l;i<=r;i++){
42         arr[i]=temp[i-l];
43     }
44 }
45 public:
46 void mergeSort(int arr[], int l, int r){
47     if(l<r){
48         int m=(l+r)/2;
49         mergeSort(arr,l,m);
50         mergeSort(arr,m+1,r);
51         merge(arr,l,m,r);
52     }
53 }
```



Custom Input

Compile & Run

Submit

Quick Sort


Medium

Accuracy: 55.23%

Submissions: 118K+

Points: 4



Participate In Monthly Hiring Challenge conducted by GeeksforGeeks !! 

Quick Sort is a Divide and Conquer algorithm. It picks an element as a pivot and partitions the given array around the picked pivot.

Given an array arr[], its starting position is low (the index of the array) and its ending position is high (the index of the array).

Note: The **low** and **high** are inclusive.

Implement the partition() and quickSort() functions to sort the array.

Example 1:

Input:

N = 5

arr[] = { 4, 1, 3, 9, 7 }

Output:

1 3 4 7 9

```
1 // } Driver Code Ends
16 class Solution
17 {
18     public:
19         //Function to sort an array using quick sort algorithm.
20         void quickSort(int arr[], int low, int high)
21         {
22             // code here
23             if(low<high){
24                 int piv = partition(arr,low,high);
25                 quickSort(arr,low,piv-1);
26                 quickSort(arr,piv+1,high);
27             }
28         }
29
30     public:
31     int partition (int arr[], int low, int high)
32     {
33         // Your code here
34         int pivot = arr[high];
35         int left = low;
36         for(int i = low ; i<=high ;i++){
37             if(arr[i]<pivot){
38                 swap(arr[i],arr[left]);
39                 left++;
40             }
41         }
42         swap(arr[left],arr[high]);
43         return left;
44     }
45 };
46
47 // } Driver Code Ends
```

[Custom Input](#)

Compile & Run

Submit