

```
#include <bits/stdc++.h>
using namespace std;
```

```
// int indexOfMaxElement(int arr[],int lo,int hi){
//   int maxEle=arr[lo];
//   int maxIndex=lo;
//   for(int i=lo;i<=hi;i++){
//     if(arr[i]>maxEle){
//       maxEle=arr[i];
//       maxIndex=i;
//     }
//   }
//   return maxIndex;
// }
```

```
//Merge Sort
```

```
vector<int> MergeTwoSortedArrays(vector<int>&arr1,vector<int>&arr2){
    int i=0;
    int j=0;
    vector<int>resArr;

    while(i<arr1.size()&&j<arr2.size()){
        if(arr1[i]<=arr2[j]){
            resArr.push_back(arr1[i]);
            i++;
        }
        else{
            resArr.push_back(arr2[j]);
            j++;
        }
    }
    while(i<arr1.size()){
        resArr.push_back(arr1[i]);
        i++;
    }

    while(j<arr2.size()){
        resArr.push_back(arr2[j]);
        j++;
    }
    return resArr;
}

vector<int> mergeSort(int arr[],int lo,int hi){
    //base case
    if(lo==hi){
```

```

        vector<int>ar;
        ar.push_back(arr[lo]);
        return ar;
    }
    int mid=(lo+hi)/2;
    vector<int>a1=mergeSort(arr,lo,mid);
    vector<int>a2=mergeSort(arr,mid+1,hi);
    vector<int>result=MergeTwoSortedArrays(a1,a2);
    return result;
}
int main() {
    int arr[6]={3,7,4,9,2,1};
    int n=6;

    //bubble sort
    // bool swap;
    // for(int i=1;i<=n;i++){
    //     swap=false;
    //     for(int j=1;j<=n-i;j++){
    //         if(arr[j-1]>arr[j]){
    //             swap(arr[j-1],arr[j]);
    //             swap=true;
    //             // int temp;
    //             // temp=arr[j-1];
    //             // arr[j-1]=arr[j];
    //             // arr[j]=temp;
    //         }
    //     }
    //     if(swap==false){
    //         break;
    //     }
    // }

    //selection sort
    // for(int i=0;i<n;i++){
    //     int maxIndex=indexOfMaxElement(arr,0,n-i-1);
    //     int indexToBeSwapped=n-i-1;
    //     swap(arr[maxIndex],arr[indexToBeSwapped]);
    // }

    // for(int i=0;i<n;i++){
    //     cout<<arr[i]<<" ";
    // }

    // int a1[4]={1,3,5,7};
    // int n1=4;

    // int a2[6]={2,7,8,9,11,12};

```

```

// int n2=6;
vector<int>ans=mergeSort(arr,0,n-1);
for(int i=0;i<ans.size();i++){
    cout<<ans[i]<<" ";
}
return 0;
}

```

Time of Merge Sort: $O(n \log n)$

Time of Bubble Sort: $O(n)$ (when array is sorted) else  $O(n^2)$

Time of Selection Sort: $O(n^2)$

Space of Merge Sort: $O(n_1+n_2)$ +recursion stack space for function calls

Space of Bubble Sort: $O(1)$

Space of Selection Sort: $O(1)$