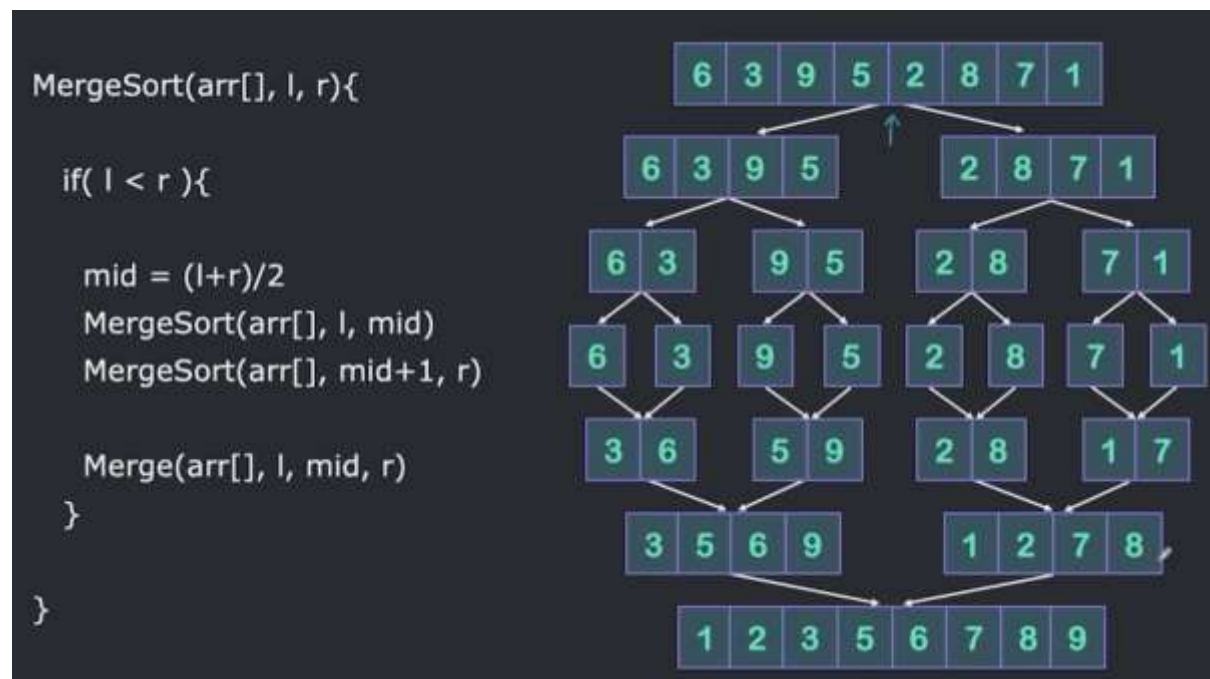


Day – 26 of the 101 days coding challenge

⇒ Merge Sort

➔ It's also a type of sorting algorithm that applies the concept of divide and conquer.

➔ Example-



➔ After dividing and merging we would be having two array

➔ First array's first elements will point by the one pointer and another one from another pointer then will compare each.

Code:

```
#include<iostream>
```

```
using namespace std;
```

```
void merge(int arr[], int l, int mid, int r){
```

```
    int n1 = mid-l+1;
```

```
    int n2=r-mid;
```

```
int a[n1];
```

```
int b[n2];
```

```
for(int i=0;i<n1;i++){
```

```
    a[i]=arr[l+i];
```

```
}
```

```
for(int i=0;i<n2;i++){
```

```
    b[i]=arr[mid+1+i];
```

```
}
```

```
int i=0;
```

```
int j=0;
```

```
int k=l;
```

```
while(i<n1 && j<n2){
```

```
    if(a[i]<b[j]){
```

```
        arr[k] = a[i];
```

```
        k++;
```

```
        i++;
```

```
    }
```

```
    else{
```

```
        arr[k] = b[j];
```

```
        k++;

        j++;
    }
}
```

```
while(i<n1){
    arr[k]=a[i];
    k++;
    i++;
}
```

```
while(j<n2){
    arr[k]=b[j];
    k++;
    j++;
}
```

```
}
```

```
void mergeSort(int arr[], int l, int r)
```

```
{
```

```
    if(l<r){
        int mid = (l+r)/2;
        mergeSort(arr, l, mid);
        mergeSort(arr, mid+1, r);
```

```

        merge(arr,l,mid,r);
    }
}

int main()
{
    int arr[] = {5,4,3,2,1};
    mergeSort(arr,0,4);
    for(int i=0;i<5;i++){
        cout<<arr[i]<<" ";
    }

    cout<<endl;

    return 0;
}

```

Output:

```

1 2 3 4 5
-----
Process exited after 0.08128 seconds with return value 0
Press any key to continue . . .

```