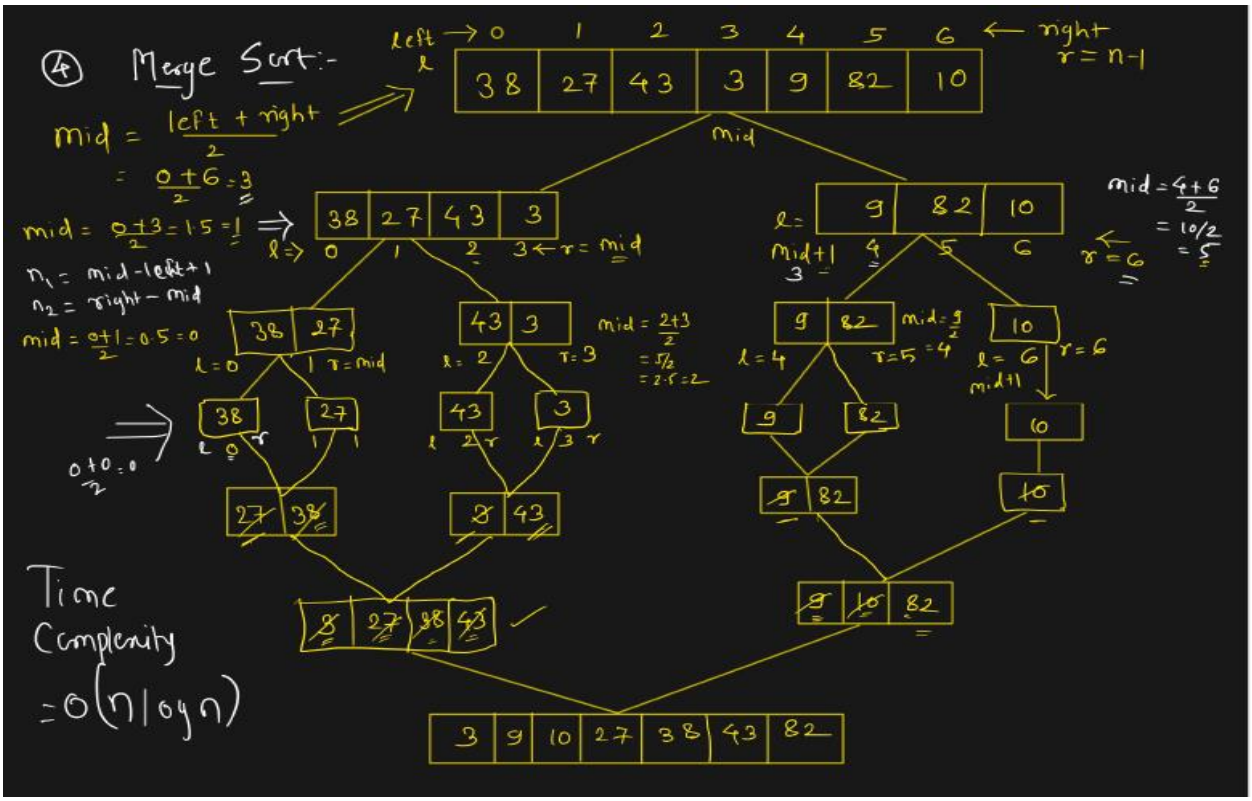
// Merge Sort Algorithm :-




```
#include<iostream>
using namespace std;
void combine(int arr[],int n,int left,int right,int mid)
{
   int n1=mid-left+1; //find out size of left array
   int n2=right-mid;  //find out size of right array
   int L[n1]; //Create left array
   int R[n2]; //Create right array
   for(int i=0;i<n1;i++)  //enter the ele in left array
   {
      L[i]=arr[left+i]; //left array elements should starts from left
   }
   for(int j=0;j<n2;j++)  //enter the ele in right array
   {
      R[j]=arr[mid+1+j]; //right array elements should starts from mid+1
   }
   int i=0,j=0,k=left;
   /* i-represent the indexing for left array
   j-represent the indexing for right array
```

```
   k -represent the indexing for merge array  */
  while(i<n1 && j<n2)
  {
    if(L[i]<R[j])
      arr[k++]=L[i++];
    else
      arr[k++]=R[j++];
  }
  while(i<n1)
    arr[k++]=L[i++];
  while(j<n2)
    arr[k++]=R[j++];
}
void mergeSort(int arr[],int n,int left, int right)
{
   if(left<right)
   {
      int mid=(left+right)/2; //find out mid for each stage
      mergeSort(arr,n,left,mid); //Divide the array in left sub array
      mergeSort(arr,n,(mid+1),right); //Divide the array in right sub array
      //Divide of original array until we getting single single ele in new sub array
      combine(arr,n,left,right,mid); //then finally we combine on comparison basis
   }
}
void initialize(int arr[],int n)
{
   cout<<"Enter the array elements=";
   for(int i=0;i<n;i++)
   {
      cin>>arr[i];
   }
}
void display(int arr[],int n)
{
   for(int i=0;i<n;i++)
   {cout<<arr[i]<<" "; }
   cout<<endl;
}
int main()
```

```
{
   int n;
   cout<<"Selec the array size=";
   cin>>n;
   int arr[n];
   initialize(arr,n);
   cout<<endl;
   cout<<"Array elements before sorting=";
   display(arr,n);
   cout<<endl;
   cout<<"Array elements after sorting=";
   int left=0,right=n-1;
   mergeSort(arr,n,left,right);
   display(arr,n);
   return 0;
}
```

```
Selec the array size=7
Enter the array elements=38 27 43 3 9 82 10

Array elements before sorting=38 27 43 3 9 82 10

Array elements after sorting=3 9 10 27 38 43 82
PS D:\itviewtech\DSA Problems\sorting>
```

//Binary Search

```
#include<iostream>
using namespace std;
int binarySearch(int arr[], int n, int low,int high, int search_ele)
{
   while(low<=high)
   {
      int mid=low+(high-low)/2;
      if(search_ele==arr[mid])
         return mid;
      else if(search_ele>arr[mid])
         low=mid+1;
      else
         high=mid-1;
   }
   return -1;
}
void initialize(int arr[],int n)
```

```
{
 cout<<"Enter the array elements=";
   for(int i=0;i<n;i++)
   {  cin>>arr[i]; }
}
void display(int arr[],int n)
{
   for(int i=0;i<n;i++)
   { cout<<arr[i]<<" ";  }
   cout<<endl;
}
int main()
{
   int n;
   cout<<"Selec the array size=";
   cin>>n;
   int arr[n];
   initialize(arr,n);
   cout<<endl;
   cout<<"Array elements are=";
   display(arr,n);
   cout<<endl;
   int low=0,high=n-1;
   int search_ele;
   cout<<"Enter the ele to search=";
   cin>>search_ele;
   int index=binarySearch(arr,n,low,high,search_ele);
   if(index!=-1)
      cout<<"Element is present at index="<<index;
   else
      cout<<"Element is not present in the array";
   return 0;
}
```

Output:-

```
Selec the array size=5
Enter the array elements=1 2 4 5 8

Array elements are=1 2 4 5 8

Enter the ele to search=5
Element is present at index=3
```