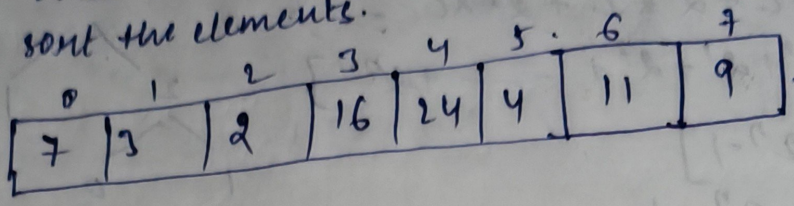
* Divide & Conquer:

(*) Divide and Conquer:

(1.) Merge sort: It is basically a divide & conquer technique to sort the elements.

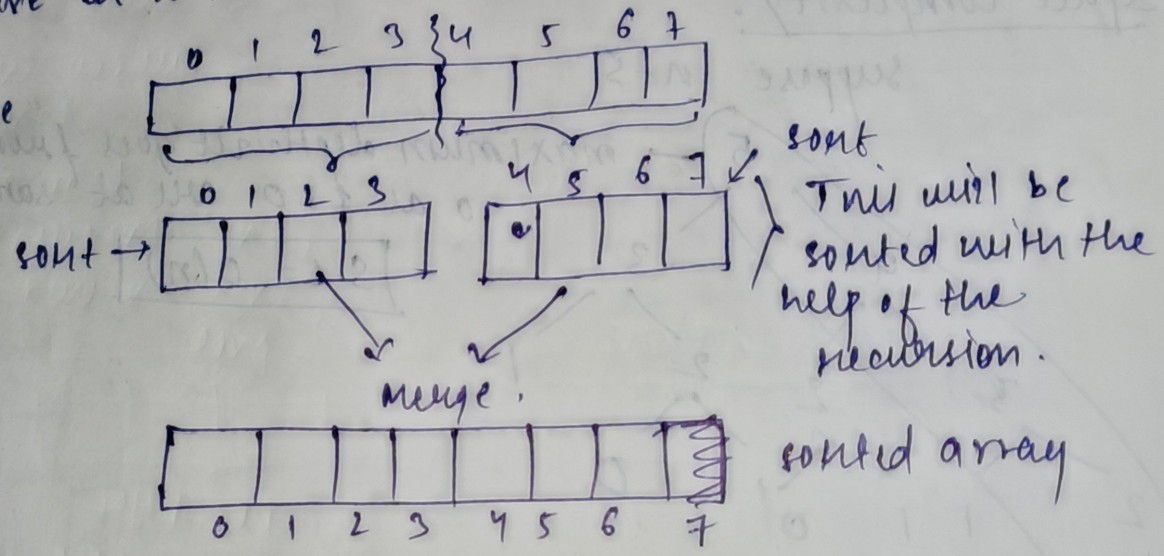
ex:



Merge sort Algorithm:

(*) Divide the array into two parts from middle

(*) Next leave it to the recursion to sort the remaining two parts.



Step1:

1. Find the mid part of the array.
2. Break into two half
3. Ask recursion to sort the two half arrays.
4. Then merge the two half.

good ques. → To understand this we need to understand a question i.e Merge 2 sorted array.

input: Array1: 2 4 6

Array2: 3 5 7 9 11

ans: 2 3 4 5 6 7 9 11 → output

This can be done with two pointer approach

array:1 | 2 | 4 | 6 | (i, ik, k)

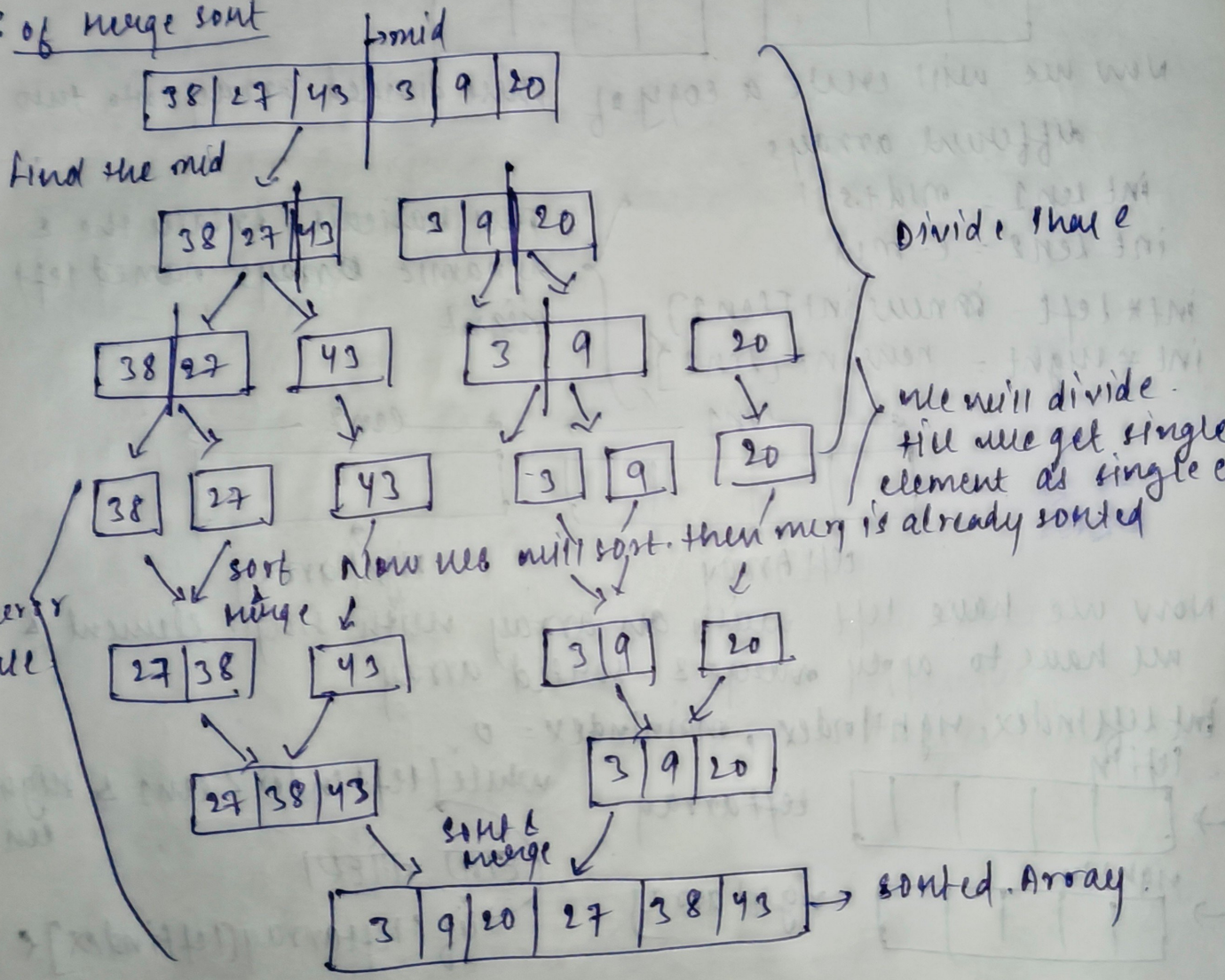
array:2 | 3 | 5 | 7 | 9 | 11 | (jk, jk, j)

2<3 ✓ i++ कर दो
3<4 ✓ j++ कर दो
4<5 ✓ i++ कर दो
5<6 ✓ j++ कर दो
6<7 ✓ i++ कर दो

ans → | 2 | 3 | 4 | 5 | 6 | 7 | 9 | 11 |

Now array 1 is completed so we will copy all the remaining elements of array 2 in ans array.

ex: of merge sort



④ In this basically we undertand the code:

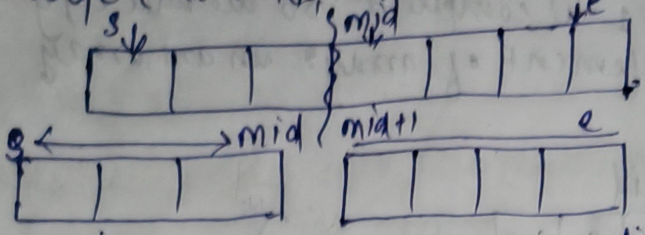
```
mergesort(int arr[], int start, int end){
    if(start >= end){ return }   → when single ele Then return कर देना
    int mid = (start + end)/2   → merge sort to divide the array into 2 equal part
```

3rd dividing the array into two equal parts

```
mergesort(arr, start, mid);
mergesort(arr, mid+1, end);
merge(arr, start, end)
```

This will basically recursion which will run till the array contains only single element.

(*) Now merge(int arr[], int start, int end) {

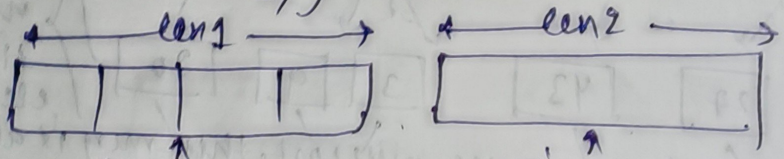


→ we have divided into 2 parts as mid = (s+e)/2.

Now we will create a copy of these divided array into two different arrays

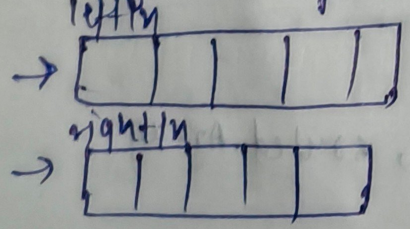
```
int len1 = mid+s+1
int len2 = e-mid
int* left = new int[len1];
int* right = new int[len2];
```

This basically create the 2 dynamic arrays named left & right



(*) Now we have left with an array with single element & we have to apply merge & sorted array

```
int leftIndex, rightIndex, mainIndex = 0
```




```
while(leftIndex < len1 & rightIndex < len2)
```

होना चाहिए

```
if(leftarray[leftIndex] < rightarray[rightIndex])
    arr[mainIndex] = leftarray[leftIndex];
```

(*) To copy elements of both divided array in left & right array we can

```
int k = start
for(int i=0; i<len1; i++){
    left[i] = arr[k]
    k++;
}
```

```
} else for right
int k = mid+1;
for(int i=0; i<len2; i++){
    right[i] = arr[k];
    k++;
}
```