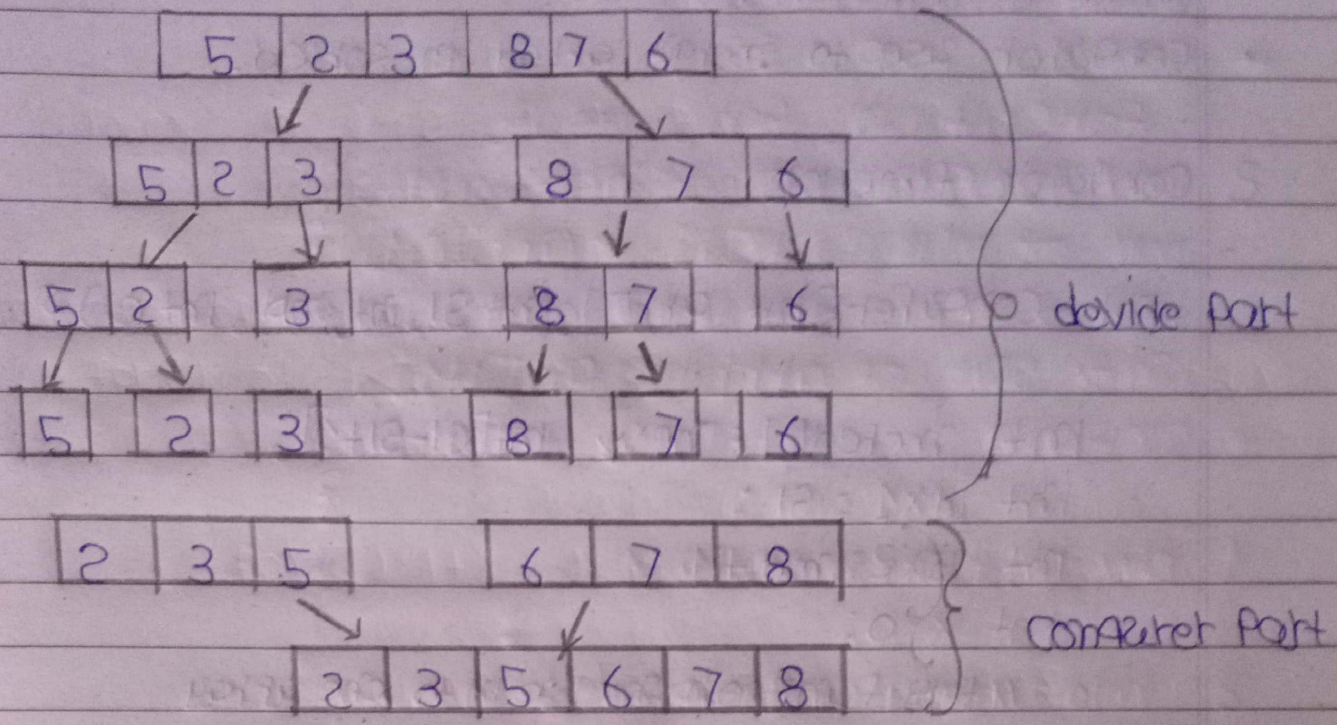
4 Merge Sort

- It is widely used sorting techniques which uses divide & conquer mechanism
- It was first sorting algorithm used on a computer and developed by John Von Neumann.
- In this type of methods the problems which can be broken into smaller problems, solve the small problems and then merge them to get the final answer.
- In this type of techniques, our large list is broken down into smaller lists and the after merge together.
- It is the process of combining two or more sorted array into a third sorted array.
- Devide the array into approximately n/2 sub-arrays of size two and set the element in each array. merging each sub-array with the adjacent sub-array will get another sorted sub-array size four.
- Repeat this process until there is only one array remaining of size n.

3 merge sort




arr → array
si → starting index
ei → end index
mid → middle element

* explain

1 devide function

```
1 int divide(int arr[], int, si, int ei){
    if (si >= ei){
        return ;                    //if starting is bigger then
                                      end index is then return
                }
     devide fnt
    int mid = si + (ei - si)/2;
    devide (arr, si, eid) ;
    devide(arr, mid+1, ei);
    conquer(arr, si, mid, ei);
```

- devide function use to make partition and devide into a single element array.
- conquer use to merge array in sorted.

2. conquer function

```
int conquer(int arr[], int si, mt, mid, int ei){

    int merged[] = new int[ei-si+1];
    int idx1 = si;
    int idx2 = mid+1;
    int x=0;
      // this function merge in to one array
  while(idx1 <= mid && mid+1 <= idx2 <= ei){
          if(arr[idx1] <= arr[idx2]){
            merged[x++] = arr[idx++];
                                    }
          else{
              merged[x++] = arr[idx2++];
              }
                                    }
// this if one array is fully place in sorted array and
  second array have a element then this while condition
  Print full array

  while(idx1 <= mid){
        merged[x++] = arr[idx1++];
                    }
```

```
while(idx2 <= eb){
    merged[x++] = arr[idx2++];
}

for i=0, j=si; i<merged.length; i++, j++){

    arr[j] = merged[i];

}
}
```