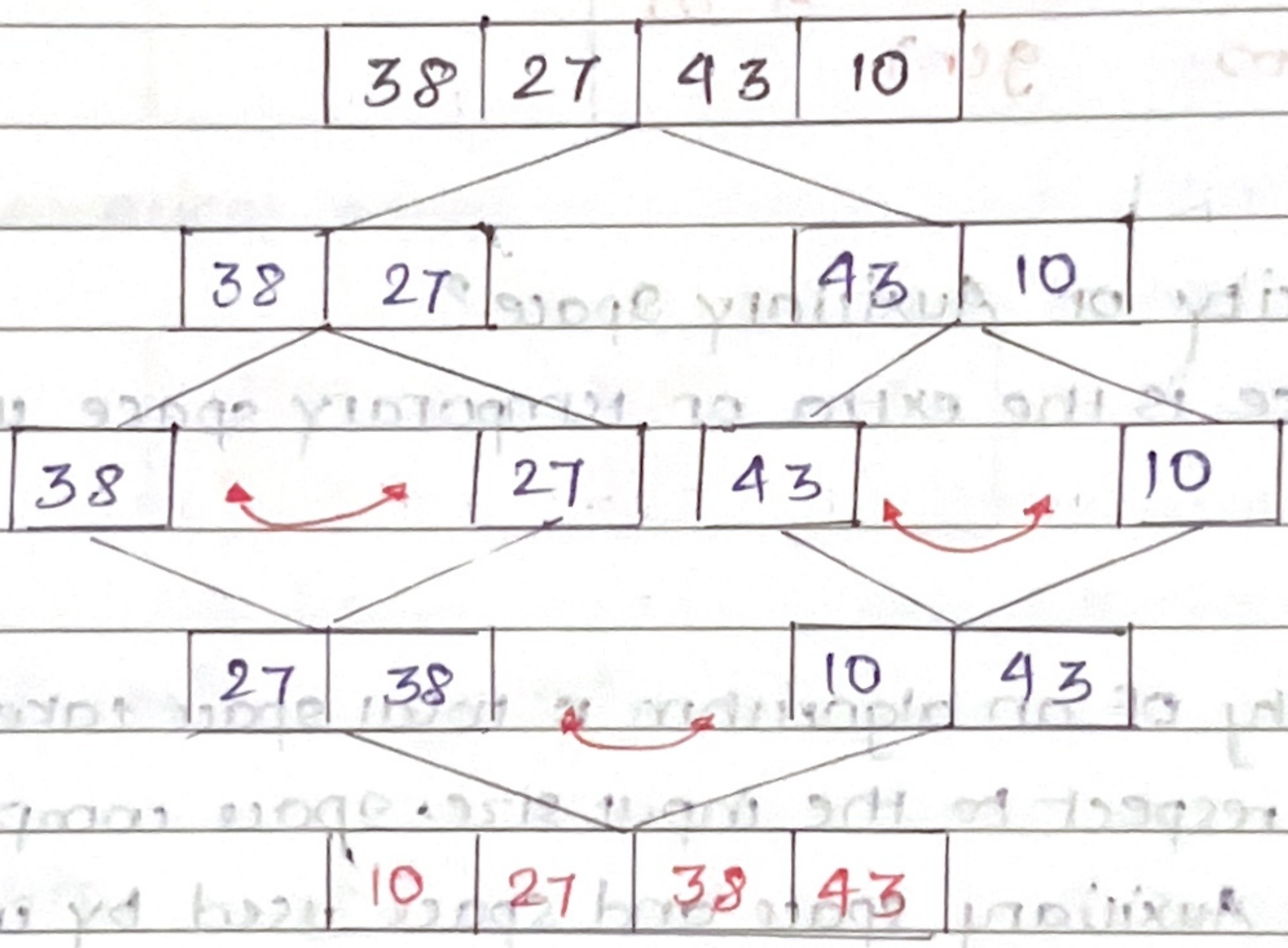
MERGE SORT

Merge sort is defined as sorting algorithm that works by dividing an array into smaller subarrays, sorting each sub array and then ~~sorting~~ merging the sorted subarrays back together to form final sorted array.

Working-

1. Divide array into two parts
2 Creth both parts sorted via recursion
3. Merge two sorted paths



Example -

```
public class Mergesort {
        public  static void main (String[] args) {
                int[] arr = {5, 4, 3, 2, 1};
                System.out.println(mergeSort(arr));
        }
        static int[] mergeSort (int[] arr) {
                if (arr.length == 1)
                        return arr;
                int mid = arr.length / 2;
                int[] left = mergeSort (Arrays.copyOfRange(arr, 0,
                                                      mid));
                int[] right = mergeSort(Arrays.copyOfRange(arr, mid,
                                                      arr.length));
                return merge(left, right);
        }
        static int[] merge (int[] left, int[] right){
              int[] mix = new int[left.length + right.length];
              int i = 0;
              int j = 0;
              int k = 0;
              while (i < left.length && j < right.length){
                   if (arrleft[i] < right[j]){
                             mix[K] = left[i];
                   }else {
                             mix[K] = right[j];
                   }
                   K++
              }
            while (i < left.length){
              mix[K] = left[i];
              k++;
              i++;
            }
        while (j < right.length){
            mix[k] = right[j];
            k++;
            j++;
        }
        return mix;
    }
```

- Merge Sort In Place

```
public class Mergesort{
    public static void main (String[] args){
        int[] arr = {5, 4, 3, 2, 1};
        mergeSortInPlace(arr, 0, arr.length);
        System.out.println(Arrays.toString(arr));
    }
    static void mergeSortInPlace(int[] arr, int s, int e){
        if (e-s == 1)
            return;
        int mid = (s+e)/2;
        mergeSortInPlace(arr, s, mid);
        mergeSortInPlace(arr, mid, e);
        mergeInPlace(arr, s, mid, e);
    }
    public static void mergeInPlace(int[] arr, int s, int m,
    int e){
        int[] mix = new int[e-s];
        int i = s;
        int j = m;
        int k = 0;
```

```
            while (i<m && j<e){
                if (arr[i] < arr[j] {
                    mix[k] = arr[i];
                } else {
                    mix[k] = arr[j];
                }
                k++
            }
            mewhile (i<m){
                mix[k] = arr[i];
                i++;
                k++;
            }
            while (j<e){
                mix[k] = arr[j];
                j++;
                k++;
            }
            For(int l=0 ; i< k mix.length ; i++){
                arr[s+l] = mix[l];
            }
        }
    }
```