

```
void merge (vector<int>& arr, str, mid, end)
```

```
    n1 ← mid - str + 1  
    n2 ← end - mid
```

```
    vector<int> leftsort(n1);  
    vector<int> rightsort(n2);
```

```
    for i = 0 to n1  
        leftsort[i] = arr[str + i]
```

```
    for j = 0 to n2  
        rightsort[j] = arr[mid + 1 + j]
```

```
    int i ← 0  
    int j ← 0  
    int k ← 0
```

```
    while (i < n1 && j < n2)  
        if (leftvector[i] < rightvector[j]) then  
            arr[k] = leftvector[i]  
            i++
```

```
        else  
            arr[k] = rightvector[j]  
            j++
```

```
    k++;
```

```
    while (i < n1)  
        arr[k] = leftvector[i]  
        i++  
        j++;
```

```
    while (j < n2)  
        arr[k] = rightvector[j]  
        j++  
        k++
```

```
void Mergesort (vector<int>& arr, str, end)
```

```
    if str > end then  
        return;
```

```
    mid ← (end + str) / 2
```

```
    Mergesort (arr, str, mid)
```

```
    Mergesort (arr, mid + 1, end)
```

```
    merge (arr, str, mid, end)
```

Divide: the problem into independent sub-problems that are similar to the original but smaller in size

Conquer: the sub-problems by solving them recursively. If they are small enough, just solve them in a straightforward manner.

$$\begin{aligned} T(n) &= 2T(n/2) + O(n) \\ &= 2[2T(n/4) + O(n/2)] \\ &= 4T(n/4) + O(n) \\ &= 8T(n/8) + O(n) \\ &\quad \vdots \\ &\quad \quad \quad kn \end{aligned}$$

Code:

Divide part

```
MergeSort(A, p, r)
if p < r
    then q ← (p+r)/2
    MergeSort(A, p, q)
    MergeSort(A, q+1, r)
    Merge(A, p, q, r)
```

So, $T(n) = 2T(n/2) + O(n)$

Conquer and Combine part (Merging)

```
Merge(A, p, q, r) // Let r-p+1 = n
1. n1 ← q - p + 1 // O(1)
2. n2 ← r - q // O(1)
for i ← 1 to n1 // O(n)
    do L[i] ← A[p + i - 1]
for j ← 1 to n2 // O(n)
    do R[j] ← A[q + j]
L[n1+1] ← ∞
R[n2+1] ← ∞
i ← 1
j ← 1
for k ← p to r // O(r-p+1) = O(n)
    do if L[i] ≤ R[j]
        then A[k] ← L[i]
        i ← i + 1
    else A[k] ← R[j]
        j ← j + 1
// Total time: O(n)
```

here the Recurrence tree will be merge basis on post order traversal

