* Mergesort (a, low, high)

i/p: unsorted Array A[0..n-1]

o/p: Sorted Array A[0...n-1]

```
if (low > high) return
mid ← (low+high)/2
Mergesort (a, low, mid)
Mergesort (a, mid+1, high)
SimpleMerge (a, low, mid, high)
```

* Algorithm SimpleMerge (A, B, C, m, n)

I/p: A is a sorted Array with m elements A[0...m]
B is a sorted Array with n elements B[0...n]

o/p: C is a sorted Array obtained after merging A & B

```
i ← j ← k ← 0
while (i<m and j<n)
  if (A[i] < B[j]) then
        C[k] ← A[i]
          i ← i+1
          k ← k+1
  else
        C[k] ← B[j]
          j ← j+1
          k ← k+1
  end if
end while
```

```
while (i<m)
      C[k] ← A[i]
        i ← i+1
        k ← k+1
end while

while (j<n)
      C[k] ← B[j]
        j ← j+1
        k ← k+1
end while
```