Input: Array $A = [a_1, a_2, ..., a_n]$.
Result: Sorted array A

```
procedure MERGE_SORT($A$)
    if $|A| \leq 1$ then return
    $mid \leftarrow \lfloor |A|/2 \rfloor$
    $A_l \leftarrow [A_1, A_2, ..., A_{mid}]$
    $A_r \leftarrow [A_{mid+1}, A_{mid+2}, ..., A_{|A|}]$

    $Merge\_Sort(A_l)$
    $Merge\_Sort(A_r)$

    $Combine(A, A_l, A_r)$
    return
```

Input: Original array A, sorted arrays L and R of A corresponding to left and right subarrays of A
Result: L and R combined into A to form a sorted array A

```
procedure COMBINE($A, L, R$)
    $l \leftarrow 0$                                  ▷ Here, we use 0-based indexing
    $r \leftarrow 0$
    $i \leftarrow 0$                                  ▷ Index for array $A$

    while $l < |L|$ OR $r < |R|$ do
        if ($l < |L|$) AND ($r \geq |R|$ OR $L[l] \leq R[r]$) then
            $A[i] \leftarrow L[l]$
            $l \leftarrow l + 1$
        else
            $A[i] \leftarrow R[r]$
            $r \leftarrow r + 1$
        $i \leftarrow i + 1$
    return
```