Algorithm 1: Merge

Input: Input
Output: Output
Procedure `Merge`(A1, A2, A)
    `i` $\leftarrow 0$
    `j` $\leftarrow 0$
    while $i <$|A1| $\wedge\ j <$|A2| **do**
        if A1$i <$ A2j **then**
            A[$i + j$] $\leftarrow$ A1i
            `i` $\leftarrow i + 1$
        else
            A[$i + j$] $\leftarrow$ A2j
            `j` $\leftarrow j + 1$
    end
    while $i <$|A1| **do**
        A[$i + j$] $\leftarrow$ A1i
        `i` $\leftarrow i + 1$
    end
    while $j <$|A2| **do**
        A[$i + j$] $\leftarrow$ A2j
        `j` $\leftarrow j + 1$
    end
    return A

Algorithm 2: SplitArray

Input: Input
Output: Output
Procedure `SplitArray`(A, $from, to$)
    `k` $\leftarrow$ *Array*
    for $i \leftarrow from$ **to** to **do**
        `append` (k, A[i])
    end
    return k

Algorithm 3: MergeSort

Input: Input
Output: Output
Procedure `MergeSort`(A)
    if $n \leq 1$ **then**
        return A
    `i` $\leftarrow \lfloor \frac{n}{2} \rfloor$
    A1 = MergeSort(A[0..i-1])
    A2 = MergeSort(A[i..n-1])
    return `Merge`($A1, A2, A$)