Algorithm 1 Merge Sort

```
1: procedure MergeSort(A, l, r)          ▷ Where A - array, l - left index, r - right index
2:     if p < r then
3:         mid = (l + r)/2
4:
5:         MergeSort(A, l, mid)          ▷ Sort first half of array
6:         MergeSort(A, mid + 1, r)      ▷ Sort second half of array
7:         Merge(A, l, r)                ▷ Merge first and second sorted halves
8:     end if
9: end procedure
```

Algorithm 2 Merge

```
   function Merge(A, l, r)               ▷ Where A - array, l - left index, r - right index
2:     mid = (l + r)/2
       n_1 = mid − l + 1
4:     n_2 = r − mid

6:     Let L[1 . . . n_1] and R[1 . . . n_2 1] be new arrays   ▷ Make two new arrays for left half and right half each
       for i = 1 to n_1 do                                     ▷ Copy first half into array L
8:         L[i] = A[l + i − 1]
       end for
10:    for j = 1 to n_2 do                                     ▷ Copy first half into array R
           R[j] = A[mid + j]
12:    end for

14:    i = 1
       j = 1
16:    for k = l to r do                                       ▷ Loop through all elements total in L and R
           if L[i] < R[j] then                                 ▷ Use element of L if smaller
18:            A[k] = L[i]
               i = i + 1
20:        else                                                ▷ Use element of R if smaller
               A[k] = R[j]
22:            j = j + 1
           end if
24:    end for
   end function
```