Sorting Algorithms

Merge Sort

Time Complexity

Best, Average and Worst Case:

O(N * LogN)

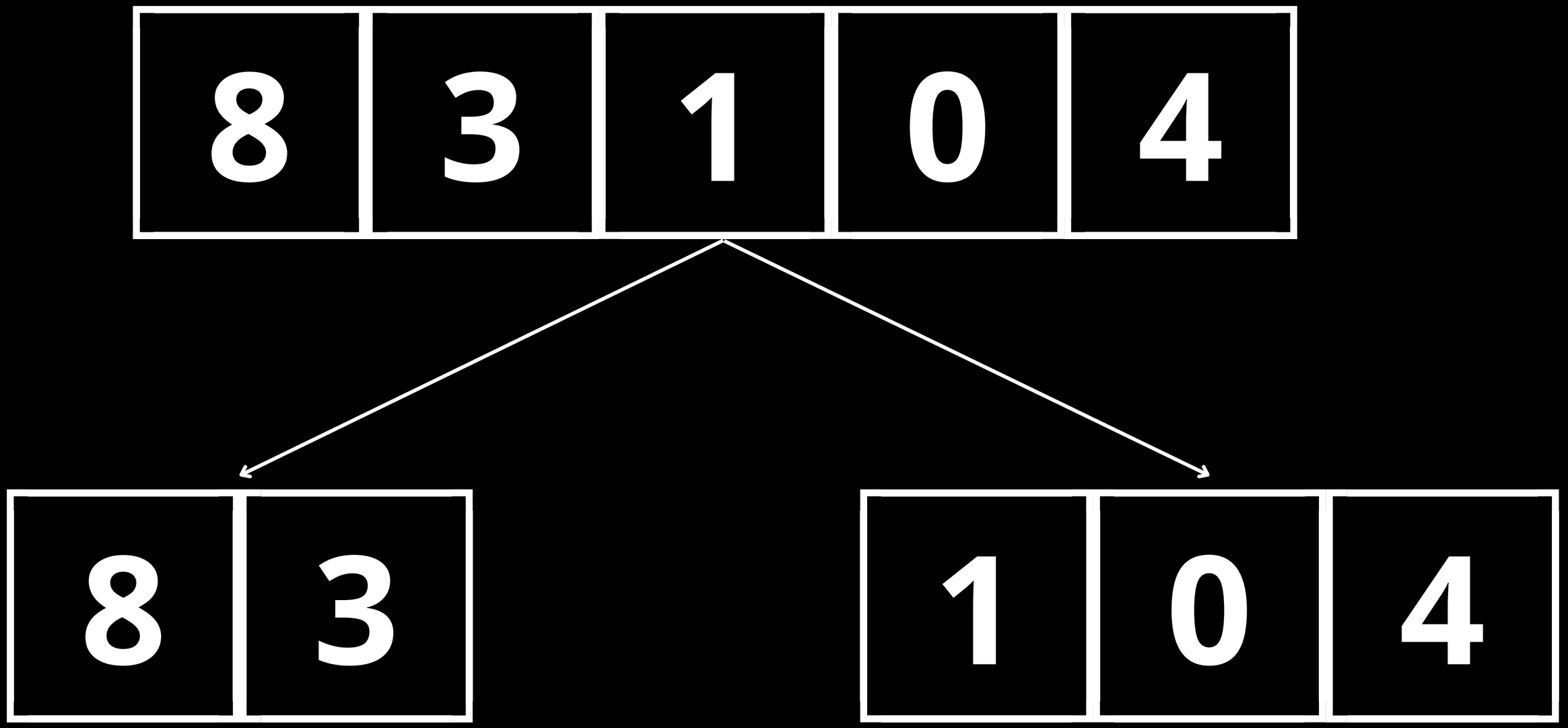
Space Complexity

Is the sum of the space complexities of the merging process and the recursion call stack, which is

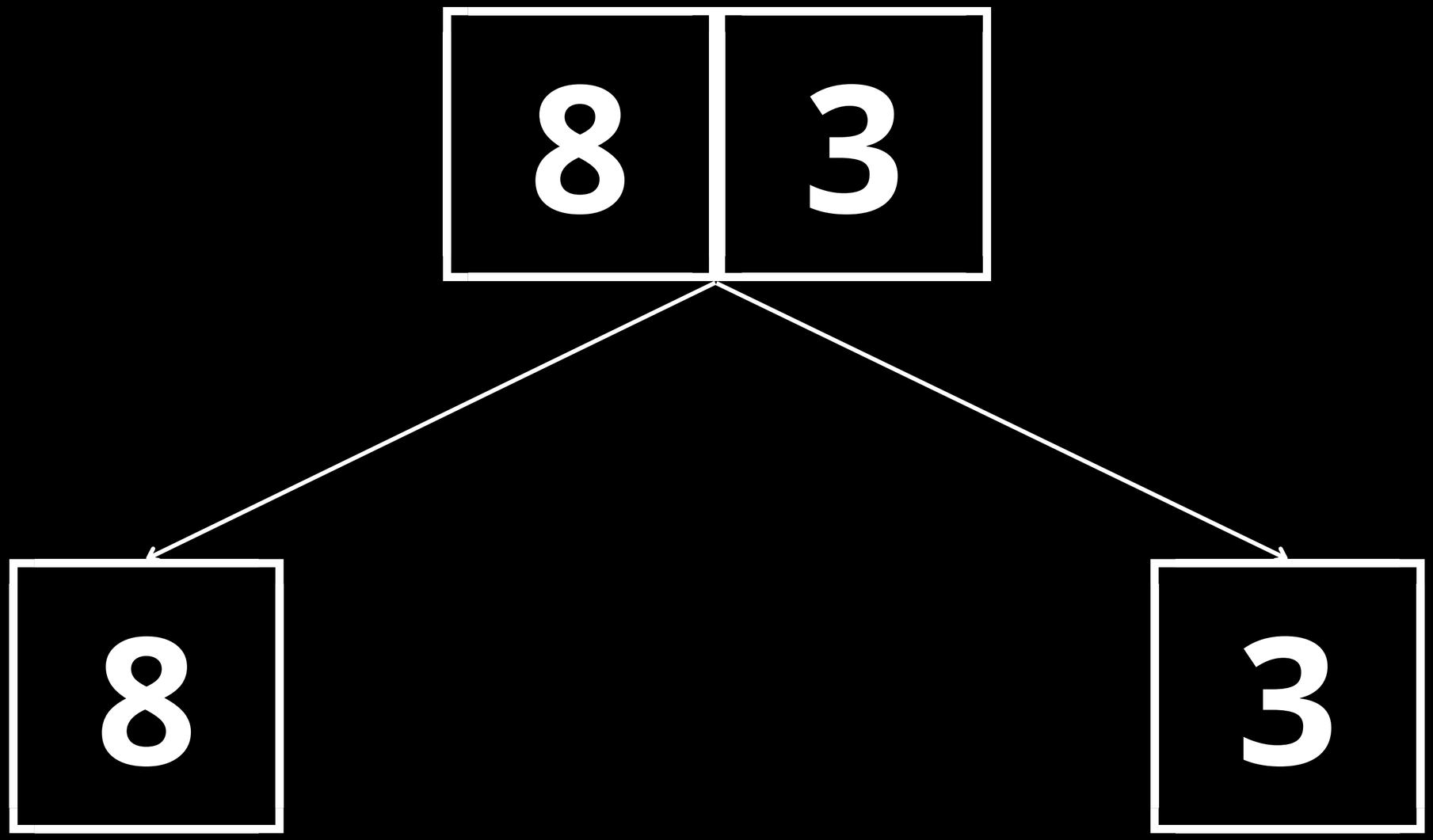
$$O(n) + O(\log n) = O(n).$$

Original Array

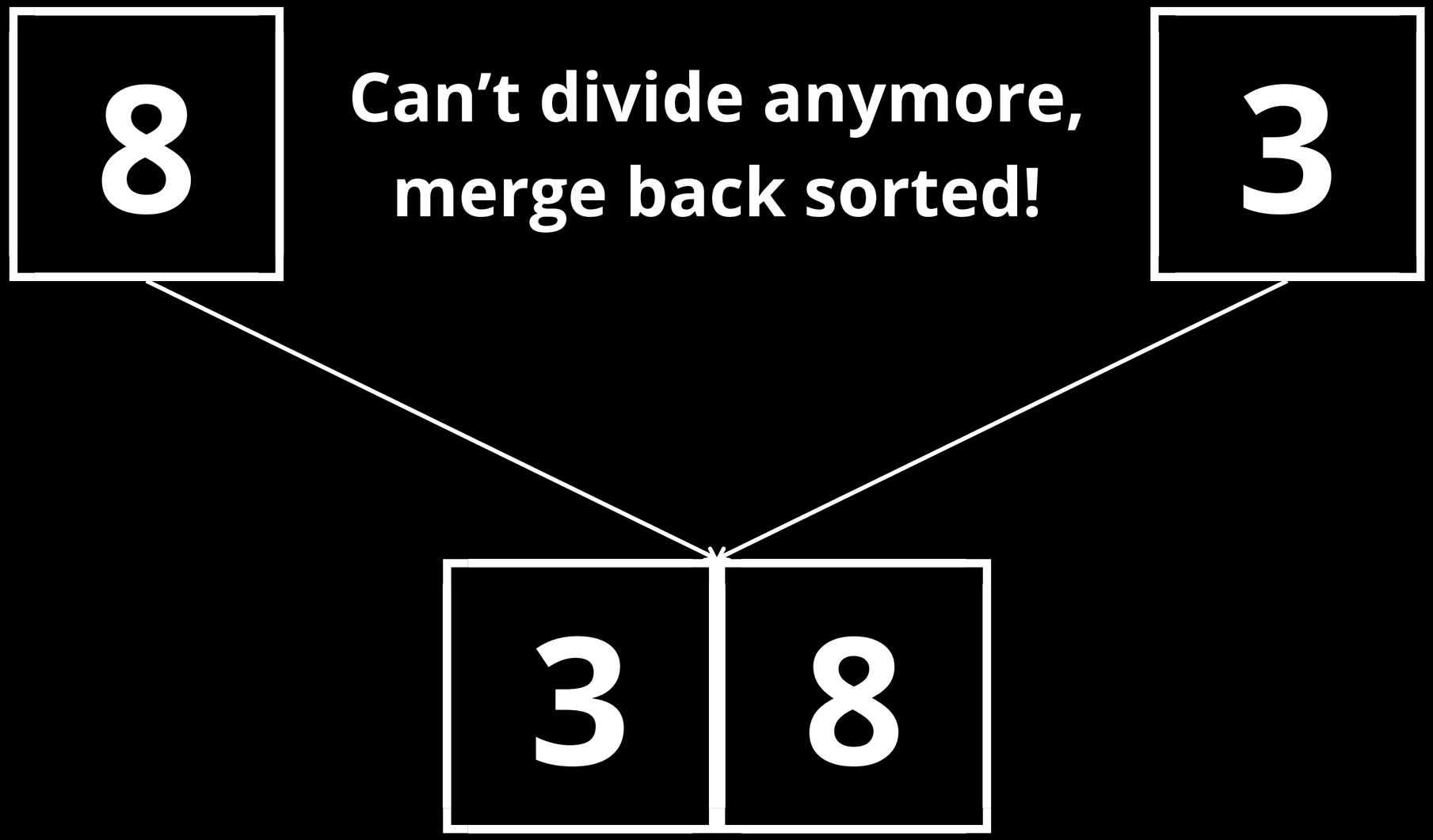
8	3	1	0	4

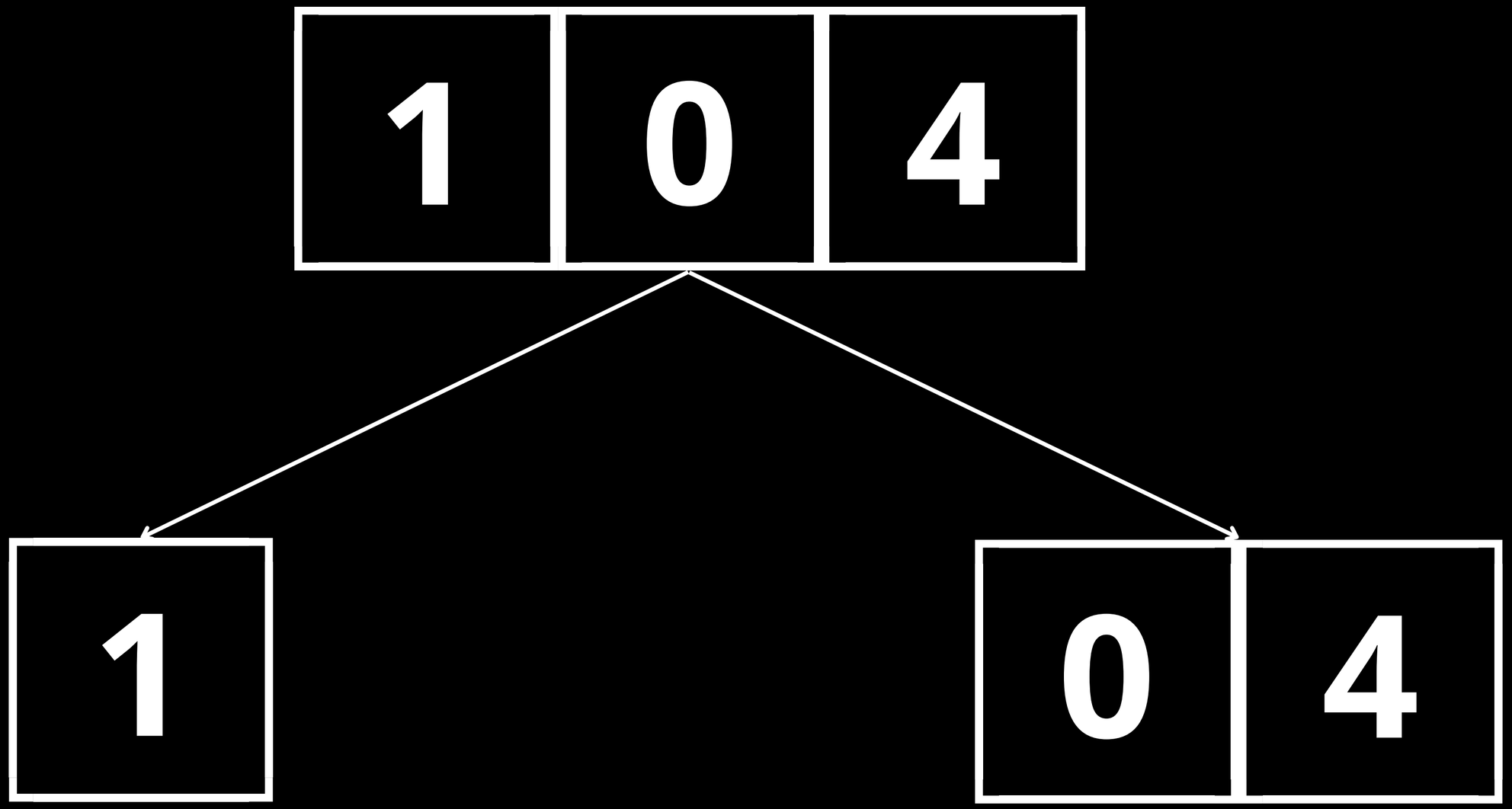


Divide the Array into 2 recursively

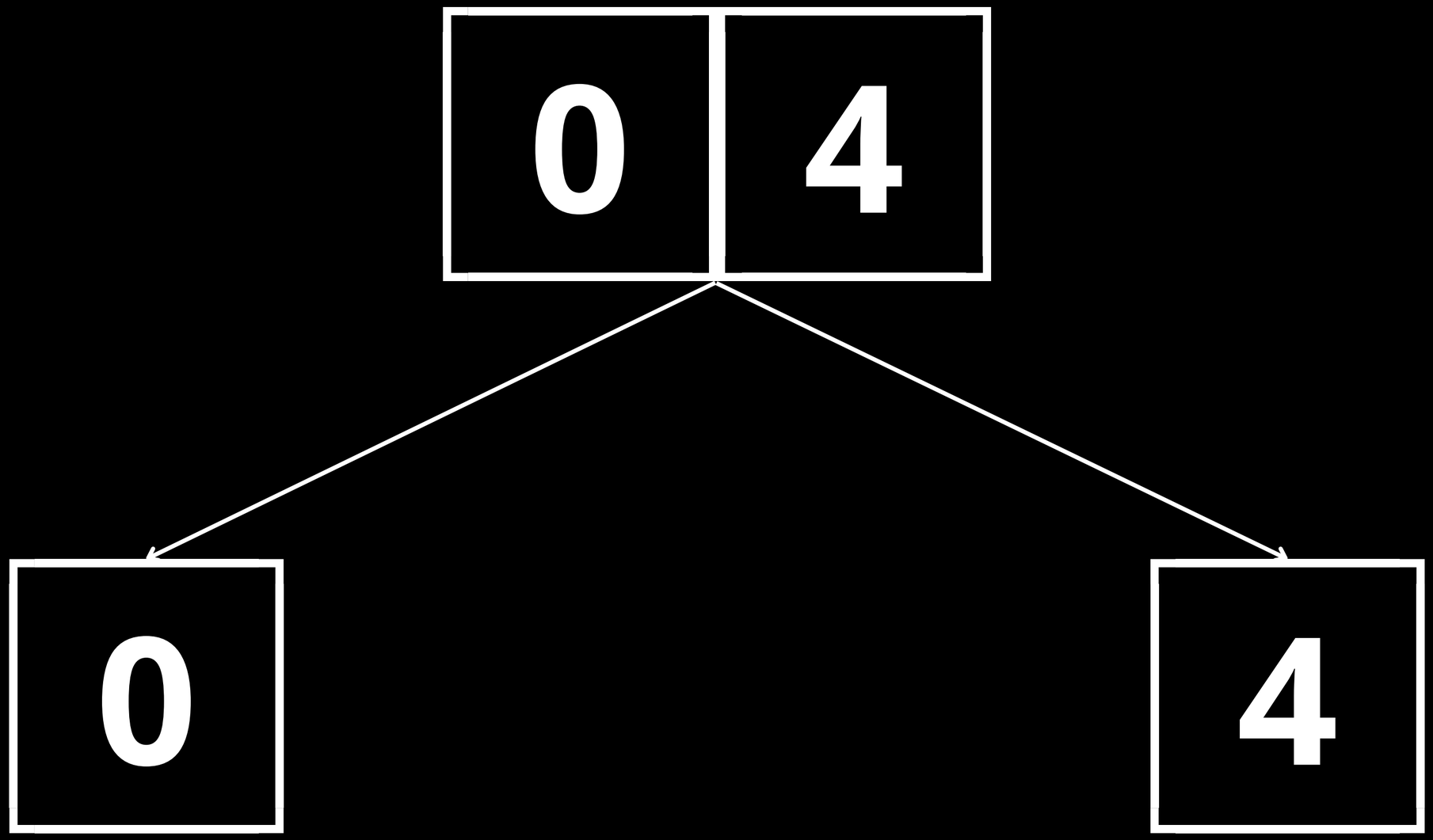


Same thing on the left sub-array

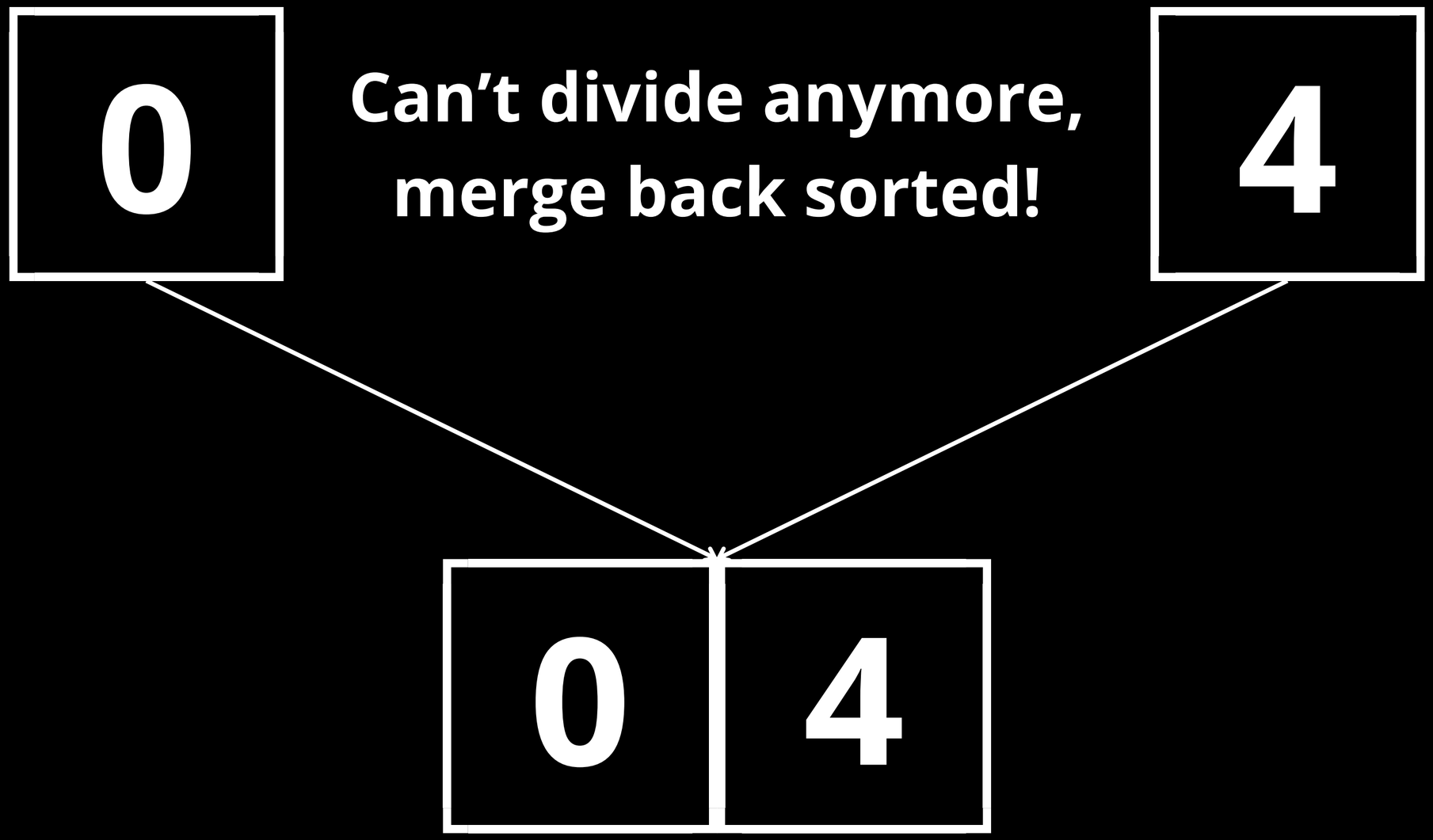

8
Can’t divide anymore,
merge back sorted!
3
3
8

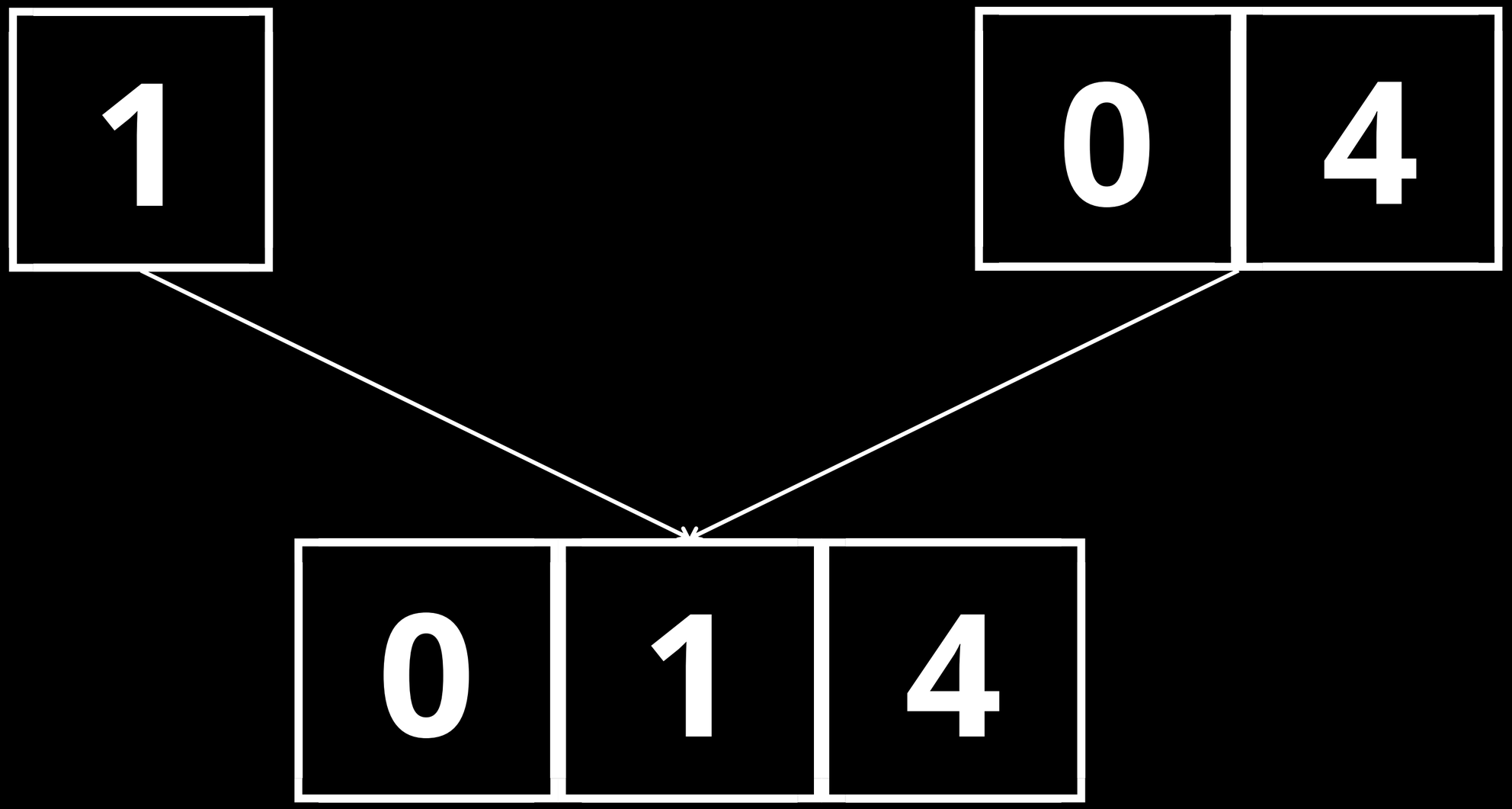


Same thing on the right sub-array

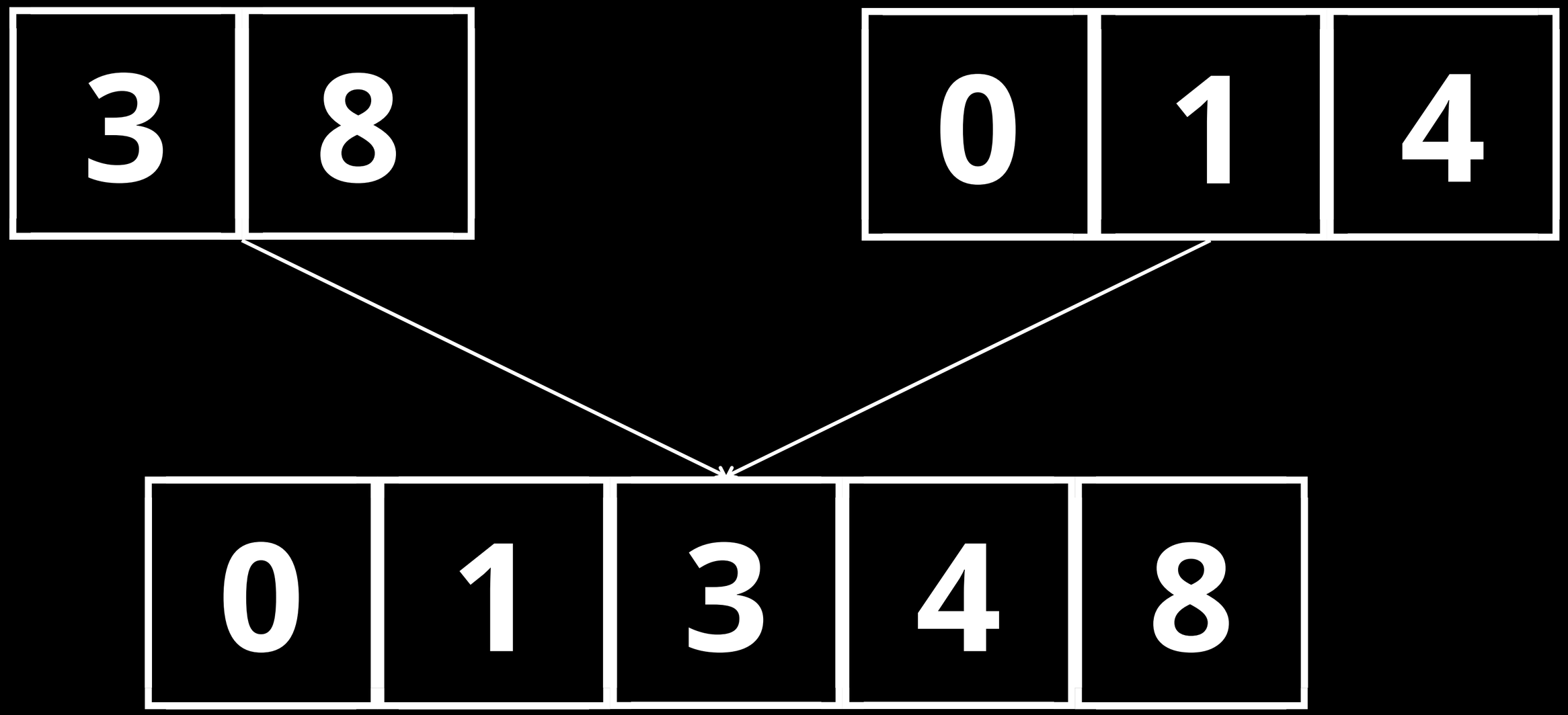


Do it again on the right subarray that still has at least 2 elements


0
Can’t divide anymore,
merge back sorted!
4
0
4



Merge it back with the previous iterarion



Merge it back with the previous iterarion

Ans this is the final sorted array after we've applied all iterations on all sub-arrays

0	1	3	4	8

The Merge Function

For merging, we use a merge function.
Here’s how it works:

1 - It will receive 3 arrays and their sizes

2 - We iterate over the 2 arrays comparing the elements

3 - We insert in the 3rd array the elements sorted

Example:

Array 1:

3	8

We compare the pink with yellow, and insert the greater one in the purple.
We than decrease the pointer on the selected array and the purple pointer.

Array 2:

0	1	4

Array 3:

Example:

Array 1:

3	8

We first insert 8, as it's greater than 4, we than decrement the yellow and the purple.

Array 2:

0	1	4

Array 3:

				8

Example:

Array 1:

3	8

We than insert 4, as it's greater than 3, and decrement the pink and the purple.

Array 2:

0	1	4

Array 3:

			4	8

Example:

Array 1:

3	8

Now we insert 3, as it's greater than one, decrement the yellow and the purple.

Array 2:

0	1	4

Array 3:

		3	4	8

Example:

Array 1:

3	8

As the yellow pointer is smaller than 0 now, our only option is to insert the remaining elements from the 2nd Array, and keep decrementing the pink and purple

Array 2:

0	1	4

Array 3:

		3	4	8

Example:

Array 1:

3	8

So we insert 1...

Array 2:

0	1	4

Array 3:

	1	3	4	8

Example:

Array 1:

3	8

Than we insert 0...

Array 2:

0	1	4

Array 3:

0	1	3	4	8

Example:

Array 1:

3	8

Array 2:

0	1	4

And we're done!

Array 3:

0	1	3	4	8

This sequence of merge events start when we only have one element in each sub-array, and they keep getting returned and merging the previous arrays until we finally merge sort the original array.