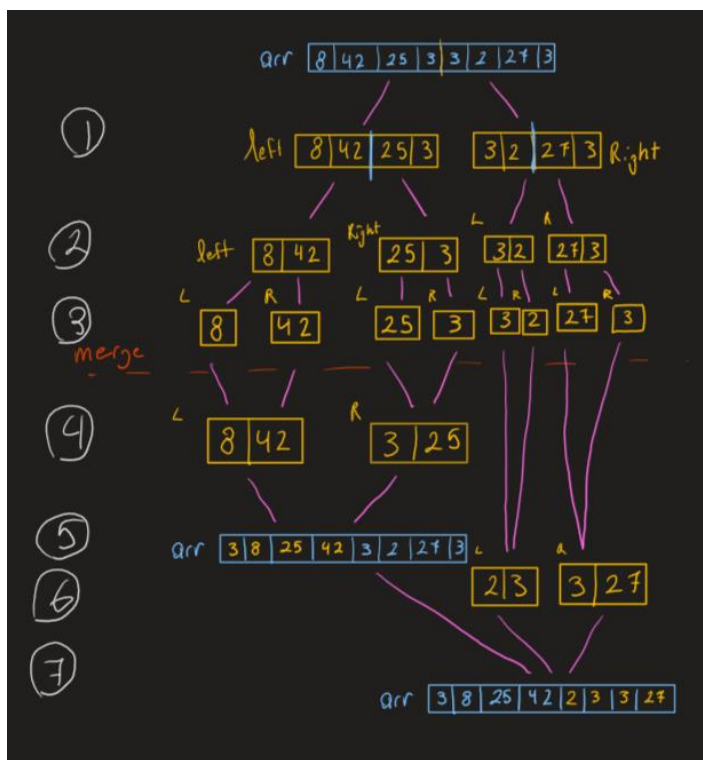


1.2

The **overall algorithm** is **$O(n \log n)$** . This is because the algorithm uses divide and conquer such that the array is divided in halves. Thus the depth of the tree is **$\log * n$** (Done in line 43 & 44). During the merge function, we iterate through the arrays which has a complexity of **n** . (Done in line 19-26) Also done in line 28 - 37 if the left and right array weren't symmetrical in length. All together merge_sort has a complexity of **$\log n * n = O(n \log n)$** .

1.3



White Numbers is the number of steps

Blue is the arr variable

Yellow is labeled **L** for left array **R** for right array.

1.4

$$O(n \log n) = 8 * \log(8) = 7.22 \approx 7.$$

As indicated with the white numbers in 1.3. It took 7 steps to complete the merge_sort algorithm. Thus the number of steps was consistent with the complexity analysis.