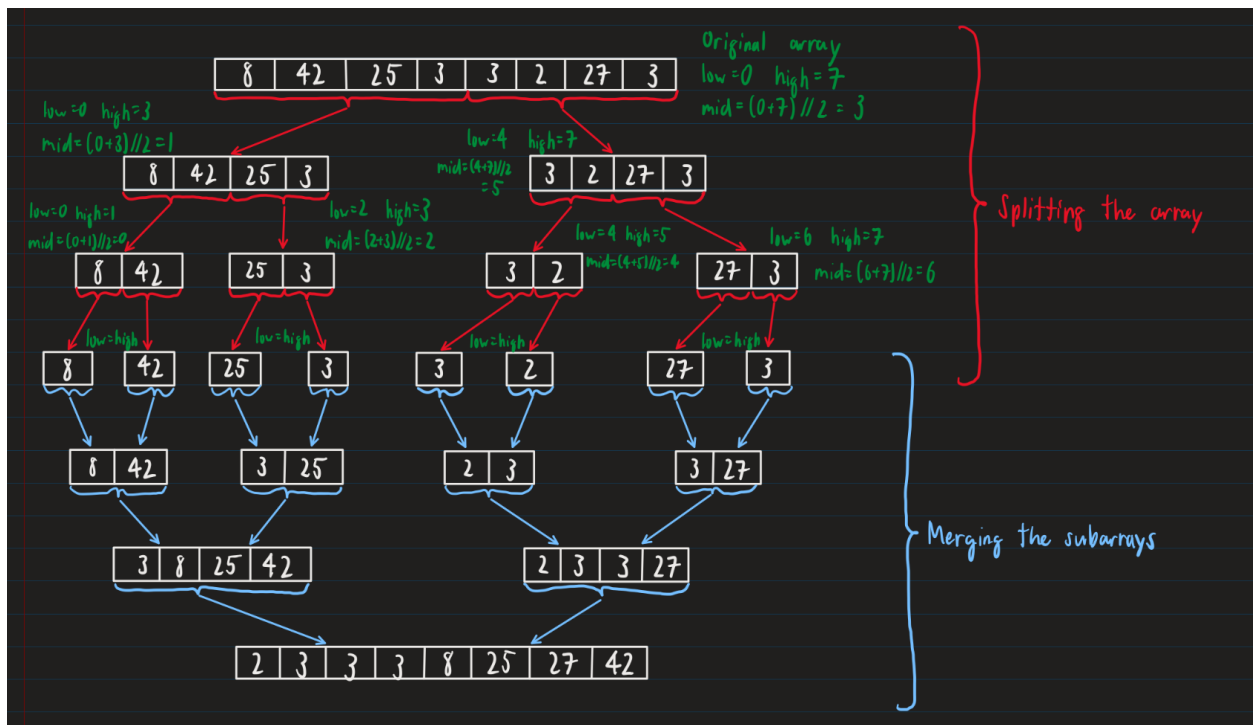


# Lab 3 Exercise 1

2. Our algorithm has a worst-case complexity of  $O(n \log n)$ . Splitting the array into each subarray is a logarithmic process that takes  $\log n$  steps since for each array element, it takes  $\log n$  steps to split the original array into a subarray of length 1 containing that element.

Merging the subarrays back together is a linear process that takes  $n$  steps because the sorted elements are in a temporary array in memory and need to be moved back into the original array one at a time. Therefore, the worst-case complexity is  $O(n \log n)$ .

3.



4. Yes because there are 3 steps ( $\log_2 8$ ) to split the array into 1-length subarrays, then putting each sorted element back into the original array takes 8 steps.