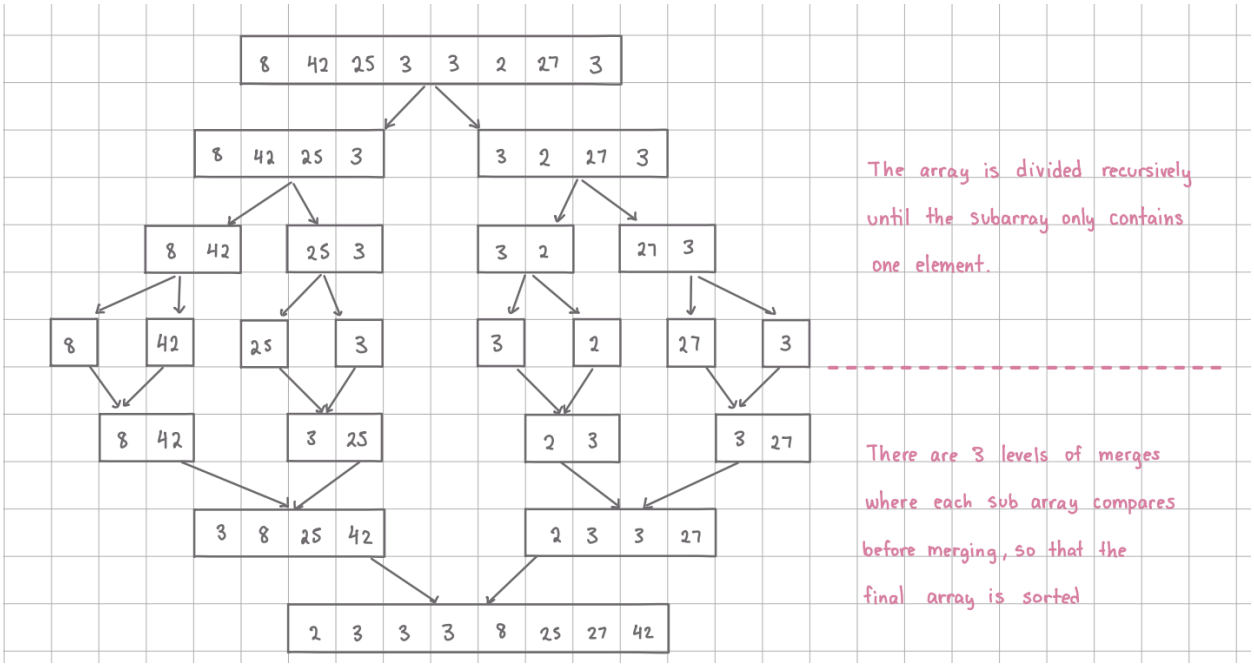
2. The Merge Sort function consistently divides the array into halves until each subarray contains only one element. This division process takes O(log(n)) time since the array is repeatedly halved. Each merge operation takes linear time proportional to the size of the subarrays being merged. Since the size of the subarrays doubles with each level of merging, the total number of merge operations is O(n). Combining the complexity of the division and merging, we get O(nlog(n)).

3.



4. Yes, it is consistent because the array is divided into 2 subarrays in three levels, which equals log (8) = 3. And the merging process is 8 merges which matches O(n).