Tutorial 8

Q1. Reverse Pairs

https://leetcode.com/problems/reverse-pairs/

1. Solve Reverse Pairs using Divide and Conquer on LeetCode and Write the Code on the Whiteboard.
2. Analyze the Time Complexity
   a) Each step (divide, conquer, combine)'s complexity
   b) Overall complexity
3. The example in the slides uses the approach: "(Given A is on the left of B and both are sorted) Increment count by the number of elements remaining in A." Will another approach work: "Increment count by the number of elements in B."?
4. Discuss your implementation:
   a) How to handle "Signed Integer Overflow"? Any pitfalls with finding the middle "(l + r) / 2", how to improve it?
   b) Is creating and releasing a helper array every time you call mergesort expensive, can it be improved?

Q2. Kth Largest Element in an Array

https://leetcode.com/problems/kth-largest-element-in-an-array/

1. Implement Kth Largest Element in an Array using Divide and Conquer on LeetCode and Write the Code on the Whiteboard:
2. Analyze the Time Complexity
   a) Each step (divide, conquer, combine)'s complexity
   b) Overall complexity
3. Implement the straightforward **sort and selection** approach and **heap-based** approach (e.g., using **priority_queue**). If implemented correctly, will any of these solutions achieve an "Accepted"? Compare and report the performance differences based on LeetCode results (e.g., 95 ms, beating 57.44%)
4. Implement another partition method. Compare and report the performance difference from LeetCode.

Q3. Sort Colors

https://leetcode.com/problems/sort-colors/

1. Which algorithm discussed in class can be used to solve this

problem?

2. Implement Sort Colors with the algorithm discussed in class on LeetCode and Write the Code on the Whiteboard.
3. Analyze the Time Complexity
    a) Each step (divide, conquer, combine)’s complexity
    b) Overall complexity
4. Implement another approach, e.g., counting sort/two-pass method (one pass to move 0s, another to move 1s). Compare and report the performance difference from LeetCode.