Crux Assignment (Stack and Queue)

1. Implement a Queue using two stacks
    a. Make Enqueue efficient
    b. Make Dequeue efficient
2. Implement a Stack using two queues
    a. Make push efficient
    b. Make pop efficient
3. Reverse a Queue using recursion
4. Check for duplicate parenthesis in an expression e.g. ((a + b) + ((c+d))) has duplicate parenthesis
5. Implement a class MinStack using the stack class we have already built. It should support O(1) push, O(1) pop and O(1) getMinimum() functions where getMinimum() returns the minimum element present in the stack. (Hint: You would need two stacks for doing this)
6. The span si of a stock's price on a certain day i is the maximum number of consecutive days (up to the current day) the price of the stock has been less than or equal to its price on day i. Given input array with all stock prices return the spans. We can do this using an array in O(n^2) time but stack can help us do it in O(n) time. Implement the array approach if you can't find a solution using stack.