

Week-4 Assignments

Question 1

```
class MyQueue {
    private Stack<Integer> input;
    private Stack<Integer> output;

    public MyQueue() {
        input = new Stack<>();
        output = new Stack<>();
    }

    public void push(int x) {
        input.push(x);
    }

    public int pop() {
        if(output.empty()){
            while(!input.empty()){
                output.push(input.pop());
            }
        }
        return output.pop();
    }

    public int peek() {
        if(output.empty()){
            while(!input.empty()){
                output.push(input.pop());
            }
        }
        return output.peek();
    }

    public boolean empty() {
        return input.empty() && output.empty();
    }
}

/**
 * Your MyQueue object will be instantiated and called as such:
 * MyQueue obj = new MyQueue();
 * obj.push(x);
 * int param_2 = obj.pop();
 * int param_3 = obj.peek();
 * boolean param_4 = obj.empty();
 */
```

The screenshot shows a LeetCode problem page for "Implement Queue using Stacks". The solution is written in Java and is marked as "Accepted". The runtime is 0 ms, and the memory usage is 41.33 MB. A bar chart shows the solution's performance relative to other submissions, with a blue bar at 100% for the first category. The code is as follows:

```

24  if (output.empty()) {
25      while (!input.empty()) {
26          output.push(input.pop());
27      }
28      return output.peek();
29  }
30  }
31
32  public boolean empty() {
33      return input.empty() && output.empty();
34  }
35  }
36
37  /**
38   * Your MyQueue object will be instantiated and called as such:
39   * MyQueue obj = new MyQueue();
40   * obj.push(x);

```

The test result shows the solution is "Accepted" with a runtime of 0 ms. The input is ["MyQueue", "push", "push", "peek", "pop", "empty"] and the output is [[], [1], [2], [], [], []].

Question 2:

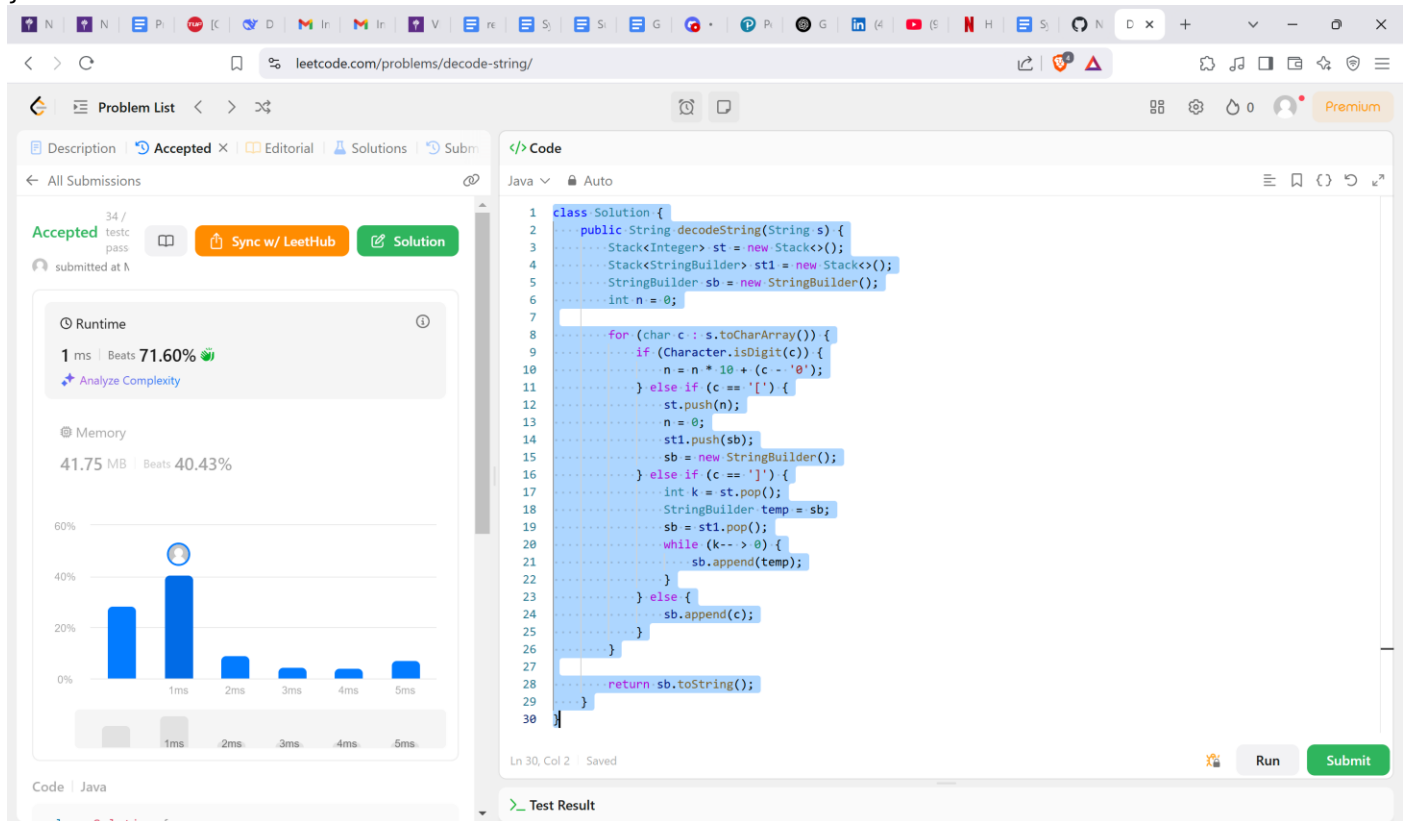
```

class Solution {
    public String decodeString(String s) {
        Stack<Integer> st = new Stack<>();
        Stack<StringBuilder> st1 = new Stack<>();
        StringBuilder sb = new StringBuilder();
        int n = 0;

        for (char c : s.toCharArray()) {
            if (Character.isDigit(c)) {
                n = n * 10 + (c - '0');
            } else if (c == '[') {
                st.push(n);
                n = 0;
                st1.push(sb);
                sb = new StringBuilder();
            } else if (c == ']') {
                int k = st.pop();
                StringBuilder temp = sb;
                sb = st1.pop();
                while (k-- > 0) {
                    sb.append(temp);
                }
            } else {
                sb.append(c);
            }
        }

        return sb.toString();
    }
}

```



Q3:

```
class Solution {
    public int peopleAwareOfSecret(int n, int delay, int forget) {
        long dp[] = new long[n + 1], mod = (long)1e9 + 7, share = 0, res = 0;
        dp[1] = 1;
        for (int i = 2; i <= n; ++i)
            dp[i] = share = (share + dp[Math.max(i - delay, 0)] - dp[Math.max(i - forget, 0)] +
mod) % mod;
        for (int i = n - forget + 1; i <= n; ++i)
            res = (res + dp[i]) % mod;
        return (int)res;
    }
}
```

Problem List

Description | Accepted x | Editorial | Solutions | Subm

All Submissions

Accepted 82 / 82
Testcases passed
submitted at Mar 29, 20.


Runtime

3 ms | Beats 83.13%

Analyze Complexity

Memory

41.43 MB | Beats 14.46%



Code | Java

Code

Java | Auto

```
1 class Solution {
2     public int peopleAwareOfSecret(int n, int delay, int forget) {
3         long dp[] = new long[n + 1], mod = (long)1e9 + 7, share = 0, res = 0;
4         dp[1] = 1;
5         for (int i = 2; i <= n; ++i)
6             dp[i] = share = (share + dp[Math.max(i - delay, 0)] - dp[Math.max(i - forget, 0)] + mod) % mod;
7         for (int i = n - forget + 1; i <= n; ++i)
8             res = (res + dp[i]) % mod;
9         return (int)res;
10    }
11 }
```

Ln 11, Col 2 | Saved

Run Submit

Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

n = 6

delay = 2