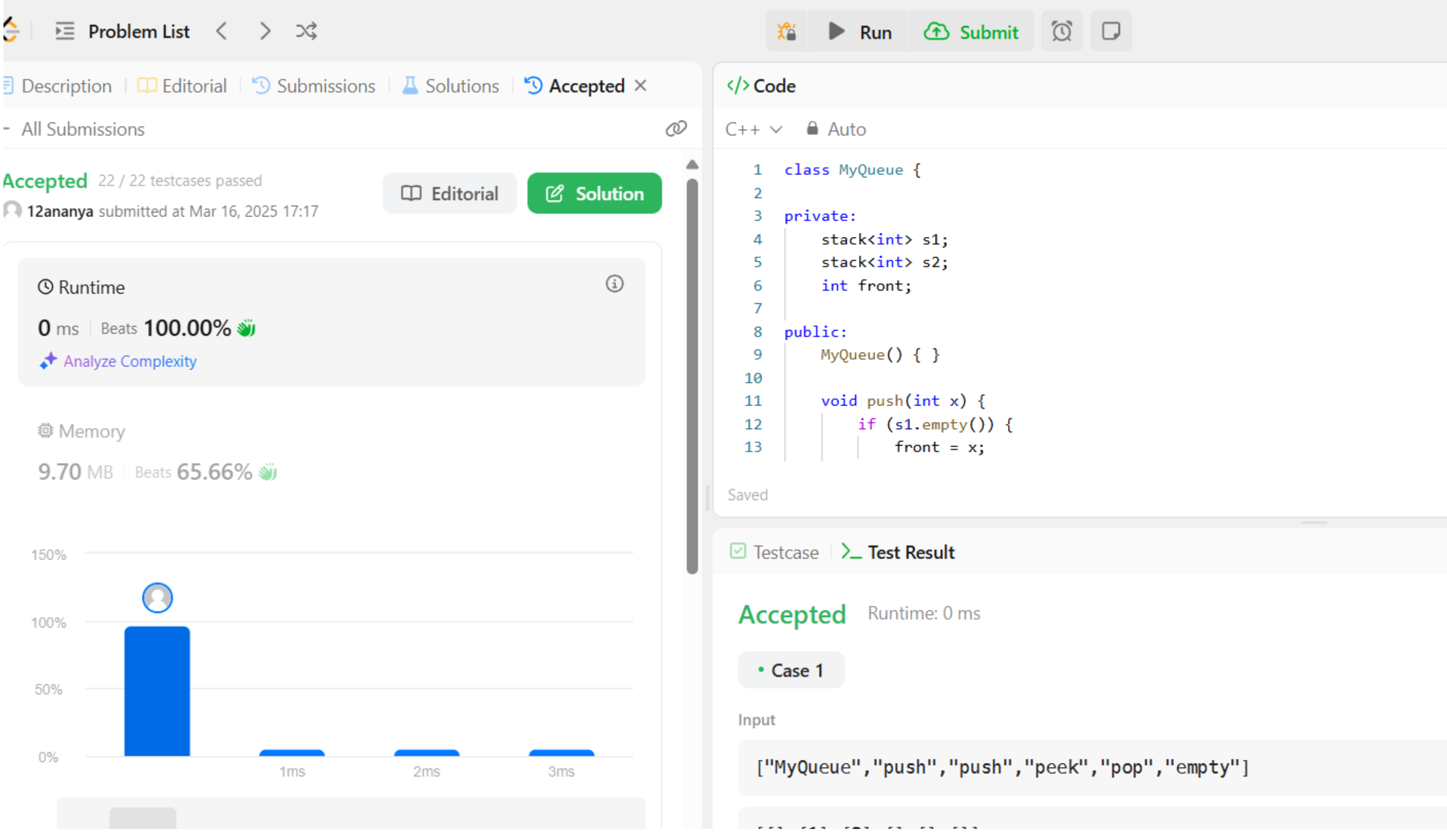
Implementing Queue using stack

```
class MyQueue {

private:
    stack<int> s1;
    stack<int> s2;
    int front;

public:
    MyQueue() { }

    void push(int x) {
        if (s1.empty()) {
            front = x;
        }
        s1.push(x);
    }

    int pop() {
        if (s2.empty()) {
            while (!s1.empty()) {
                s2.push(s1.top());
                s1.pop();
            }
        }
        int val = s2.top();
        s2.pop();
        return val;
    }

    int peek() {
        return !s2.empty() ? s2.top() : front;
    }

    bool empty() {
        return s1.empty() && s2.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();
 * bool param_4 = obj->empty();
 */
```



Accepted 22 / 22 testcases passed

12ananya submitted at Mar 16, 2025 17:17

Runtime

0 ms | Beats 100.00%

Memory

9.70 MB | Beats 65.66%

```
class MyQueue {

private:
    stack<int> s1;
    stack<int> s2;
    int front;

public:
    MyQueue() { }

    void push(int x) {
        if (s1.empty()) {
            front = x;
```

Saved

Accepted Runtime: 0 ms

Input

```
["MyQueue","push","push","peek","pop","empty"]
```