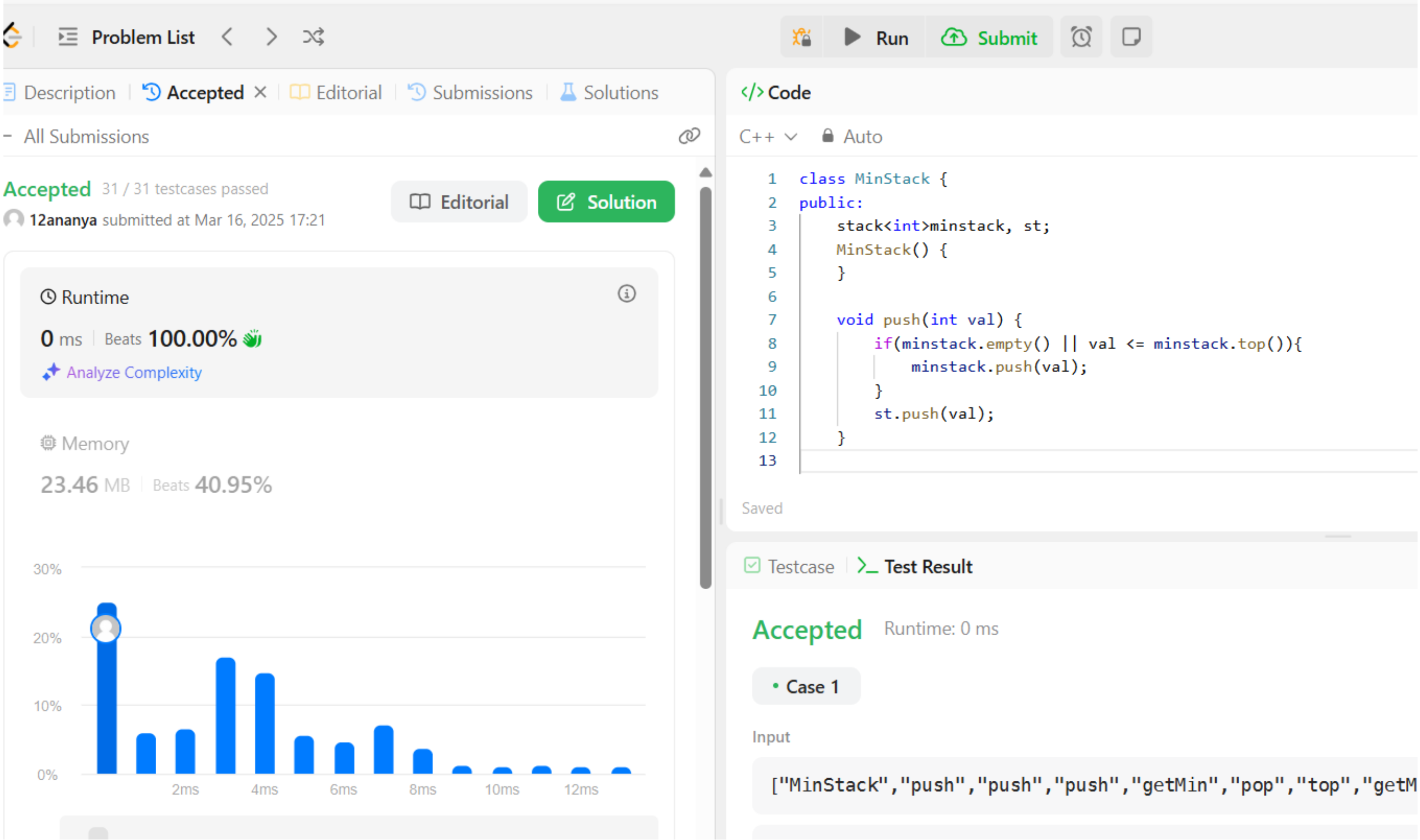
Implement Min Stack using Two Stacks

```
class MinStack {
public:
    stack<int>minstack, st;
    MinStack() {
    }

    void push(int val) {
        if(minstack.empty() || val <= minstack.top()){
            minstack.push(val);
        }
        st.push(val);
    }

    void pop() {
        if(minstack.top() == st.top()){
           minstack.pop();
        }

        st.pop();
    }

    int top() {
        return st.top();
    }

    int getMin() {
        return minstack.top();
    }
};

/**
 * Your MinStack object will be instantiated and called as such:
 * MinStack* obj = new MinStack();
 * obj->push(val);
 * obj->pop();
 * int param_3 = obj->top();
 * int param_4 = obj->getMin();
 */
```



Accepted 31 / 31 testcases passed

12ananya submitted at Mar 16, 2025 17:21

Runtime

0 ms | Beats 100.00%

Memory

23.46 MB | Beats 40.95%

```
class MinStack {
public:
    stack<int>minstack, st;
    MinStack() {
    }

    void push(int val) {
        if(minstack.empty() || val <= minstack.top()){
            minstack.push(val);
        }
        st.push(val);
    }

```

Accepted Runtime: 0 ms

Input

```
["MinStack","push","push","push","getMin","pop","top","getM
```