class MinStack {

    constructor() {

        this.stack = [];    // 主栈

        this.minStack = []; // 辅助栈，存储当前最小值

    }

    push(val) {

        this.stack.push(val);

        // 如果辅助栈为空或新值<=当前最小值，压入辅助栈

        if (this.minStack.length === 0 || val <= this.getMin()) {

            this.minStack.push(val);

        }

    }

    pop() {

        const val = this.stack.pop();

        // 如果弹出的值是当前最小值，也从辅助栈弹出

        if (val === this.getMin()) {

            this.minStack.pop();

        }

        return val;

    }

    top() {

        return this.stack[this.stack.length - 1];

    }

    getMin() {

        return this.minStack[this.minStack.length - 1];

    }

}

