

# 画解剑指 Offer - LeetBook - 力扣 (LeetCode) 全球极客挚爱的技术成长平台

 [leetcode.cn/leetbook/read/illustrate-lcof/xz6qdm](https://leetcode.cn/leetbook/read/illustrate-lcof/xz6qdm)

## A 剑指 Offer 30. 包含 min 函数的栈 - 解决方案

### 题目描述

定义栈的数据结构，请在该类型中实现一个能够得到栈的最小元素的 `min` 函数在该栈中，调用 `min`、`push` 及 `pop` 的时间复杂度都是  $O(1)$ 。

示例:

```
MinStack minStack = new MinStack();
minStack.push(-2);
minStack.push(0);
minStack.push(-3);
minStack.min();    --> 返回 -3.
minStack.pop();
minStack.top();     --> 返回 0.
minStack.min();     --> 返回 -2.
```

提示：

1. 各函数的调用总次数不超过 20000 次

### 解题方案

#### 思路

- 标签：辅助栈
- 整体思路：`push`、`pop`、`top` 操作可以通过建立普通的栈结构完成操作，对于取最小值 `min` 函数则需要建立辅助栈，辅助栈中降序存储 `push` 过程中的值
- 时间复杂度： $O(1)$ ，空间复杂度： $O(n)$

#### 算法流程

1. `MinStack` 构造函数中初始化数据栈 `stack1` 和辅助栈 `stack2`
2. `push` 函数中将 `x` 正常添加到 `stack1` 中，如果 `stack2` 为空或者 `stack2` 栈顶值大于等于 `x` 时，则将 `x` 加入 `stack2` 中，这样保证了 `stack2` 中的值一定是降序的，存储的数量会小于等于 `stack1`
3. `pop` 函数中首先 `stack1` 需要将值 `pop` 出去，如果 `stack2` 栈顶数据与 `stack1` 栈顶数据相等，则将 `stack2` 的值也 `pop` 出去，保证数据栈和辅助栈的数据一致性
4. `top` 函数则直接取 `stack1` 栈顶值即可

5. `min` 函数则直接取 `stack2` 栈顶值即可

## 代码

---

```
class MinStack {
    Stack<Integer> stack1;
    Stack<Integer> stack2;

    /** initialize your data structure here. */
    public MinStack() {
        stack1 = new Stack<>();
        stack2 = new Stack<>();
    }

    public void push(int x) {
        stack1.add(x);
        if(stack2.empty() || stack2.peek() >= x) {
            stack2.add(x);
        }
    }

    public void pop() {
        if(stack1.pop().equals(stack2.peek())) {
            stack2.pop();
        }
    }

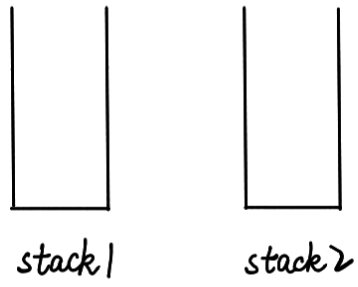
    public int top() {
        return stack1.peek();
    }

    public int min() {
        return stack2.peek();
    }
}

/**
 * Your MinStack object will be instantiated and called as such:
 * MinStack obj = new MinStack();
 * obj.push(x);
 * obj.pop();
 * int param_3 = obj.top();
 * int param_4 = obj.min();
 */
```

## 画解

---



初始化

1 / 9

© 本 LeetBook 由「力扣」和作者共同制作和发行，版权所有侵权必究。