



链滴

LeetCode 刷题 (1)——栈的最小值

作者: [zyk](#)

原文链接: <https://ld246.com/article/1584800170321>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



题目

- 题目描述

请设计一个栈，除了常规栈支持的 pop 与 push 函数以外，还支持 min 函数，该函数返回栈元素中最小值。执行 push、pop 和 min 操作的时间复杂度必须为 $O(1)$ 。

- 示例

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

解题思路

可以定义两个栈，一个栈用于存放普通数据（数据栈），另一个栈用于记录当前栈中的最小值（最小栈）。

栈中最小值只有在入栈或出栈后才会发生变化。当入栈时，只需比较入栈元素是否小于原来栈中最小值，若小于将该元素值保存到最小值栈中；若大于等于，则将原来栈中最小值再次保存到最小值栈中。出栈时，依次将数据栈和最小值栈中元素出栈即可。

获取栈中最小值只需要返回最小值栈的栈顶元素即可。

具体代码

```
class MinStack:
    def __init__(self):
        """初始化"""
        self.items = [] # 数据栈
        self.min_items = [] # 最小值栈

    def push(self, x: int) -> None:
        """入栈"""
        self.items.append(x) # 将元素压入数据栈
        if (self.min_items and x < self.min_items[-1]) or not self.min_items: # 如果最小值栈中有
            素, 且入栈元素小于栈顶元素或最小值栈是空栈
                self.min_items.append(x) # 将元素压入最小值栈
            else: # 最小值未发生变化
                self.min_items.append(self.min_items[-1])

    def pop(self) -> None:
        """出栈"""
        self.items.pop()
        self.min_items.pop()

    def top(self) -> int:
        """获取栈顶元素"""
        return self.items[-1]

    def get_min(self) -> int:
        """获取栈中最小值"""
        return self.min_items[-1]
```