

155 最小栈

题目描述

设计一个支持 push, pop, top 操作，并能在常数时间内检索到最小元素的栈。

- push(x) -- 将元素 x 推入栈中。
- pop() -- 删除栈顶的元素。
- top() -- 获取栈顶元素。
- getMin() -- 检索栈中的最小元素。

示例:

```
1  MinStack minStack = new MinStack();
2  minStack.push(-2);
3  minStack.push(0);
4  minStack.push(-3);
5  minStack.getMin();   --> 返回 -3.
6  minStack.pop();
7  minStack.top();      --> 返回 0.
8  minStack.getMin();   --> 返回 -2.
```

代码

```
1  class MinStack:
2
3      def __init__(self):
4          """
5          initialize your data structure here.
6          """
7          self.stack = []
8          self.top_pointer = -1
9          self.min = float('inf')
10
11      def push(self, x: int) -> None:
12          self.stack.append(x)
```

```

13         self.top_pointer += 1
14         if x < self.min:
15             self.min = x
16
17     def pop(self) -> None:
18         self.stack.pop()
19         self.top_pointer -= 1
20         self.min = float('inf')
21         for i in self.stack:
22             if i < self.min:
23                 self.min = i
24
25     def top(self) -> int:
26         return self.stack[self.top_pointer]
27
28     def getMin(self) -> int:
29         return self.min
30
31
32 # Your MinStack object will be instantiated and called as such:
33 # obj = MinStack()
34 # obj.push(x)
35 # obj.pop()
36 # param_3 = obj.top()
37 # param_4 = obj.getMin()

```

成功 显示详情 >

执行用时：128 ms, 在Min Stack的Python3提交中击败了49.57%的用户

内存消耗：15.9 MB, 在Min Stack的Python3提交中击败了99.88%的用户

进行下一个挑战：

滑动窗口最大值

最大栈

炫耀一下:    

提交时间	状态	执行用时	内存消耗	语言
几秒前	通过	128 ms	15.9 MB	python3