

剑指Offer（二十）：包含min函数的栈

🕒 2017年12月9日 15:59:54 🗨 24 🌡 3,245 °C 🖨 编辑



一、前言

本系列文章为《剑指Offer》刷题笔记。

刷题平台：[牛客网](#)

书籍下载：[共享资源](#)

二、题目

定义栈的数据结构，请在类型中实现一个能够得到栈最小元素的min函数。

1、思路

使用两个stack，一个为数据栈，另一个为辅助栈。数据栈用于存储所有数据，辅助栈用于存储最小值。

举个例子：

入栈的时候：首先往空的数据栈里压入数字3，显然现在3是最小值，我们也把最小值压入辅助栈。接下来往数据栈里压入数字4。由于4大于之前的最小值，因此我们只要入数据栈，不压入辅助栈。

出栈的时候：当数据栈和辅助栈的栈顶元素相同的时候，辅助栈的栈顶元素出栈。否则，数据栈的栈顶元素出栈。

获得栈顶元素的时候：直接返回数据栈的栈顶元素。

栈最小元素：直接返回辅助栈的栈顶元素。

2、代码

C++:

```
1 class Solution {
2 public:
3     void push(int value) {
4         Data.push(value);
5         if(Min.empty()){
6             Min.push(value);
7         }
8         if(Min.top() > value){
9             Min.push(value);
10        }
11    }
12    void pop() {
13        if(Data.top() == Min.top()){
14            Min.pop();
15        }
16        Data.pop();
17    }
18    int top() {
19        return Data.top();
20    }
21    int min() {
22        return Min.top();
23    }
24 private:
25     stack<int> Data;           //数据栈
26     stack<int> Min;           //最小栈
```

Python:

Pytho

```
1 class Solution:
2     def __init__(self):
3         self.Data = []
4         self.Min = []
5     def push(self, node):
6         # write code here
7         self.Data.append(node)
8         if self.Min:
9             if self.Min[-1] > node:
10                self.Min.append(node)
11            else:
12                self.Min.append(self.Min[-1]) #保证Min栈的栈顶一直指向最小的数字，即使出现重复输入也不会出错
13        else:
14            self.Min.append(node)
15    def pop(self):
16        # write code here
17        if self.Data == []:
18            return None
19        self.Min.pop()
20        return self.Data.pop()
21    def top(self):
22        # write code here
23        if self.Data == []:
24            return None
25        return self.Data[-1]
26    def min(self):
27        # write code here
28        if self.Min == []:
29            return None
30        return self.Min[-1]
```



微信公众号

分享技术，乐享生活：微信公众号搜索「JackCui-AI」关注一个在互联网摸爬滚打的潜行者。

不相信自己的人连努力的价值都没有。--- 凯《火影忍者》