

# 面试题30：包含min函数的栈

## 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中所含最小元素的min函数（时间复杂度应为 $O(1)$ ）。

## 解题思路

除了原来的栈s。增加一个辅助栈s\_min，用来保存每次进栈时的最小元素。

## C++

```
1 class Solution {
2 public:
3     //push操作
4     void push(int value) {
5         //栈s：元素value直接进栈s
6         s.push(value);
7         //栈s_min：判断s_min是否为空或者value是否小于s_min的栈顶元素
8         if(s_min.empty() || value < s_min.top())
9             //如果是，将value压入栈s_min
10            s_min.push(value);
11     }
12 }
```

```

12         //否则将s_min.top()压入栈s_min (数字相
    同, 重复操作, 浪费空间)
13         //否则不作为
14         s_min.push(s_min.top());
15     }
16     //pop操作
17     void pop() {
18         //栈s: s.pop()
19         s.pop();
20         //栈s_min: 如果s.top()等于s_min.pop(), 则
    s_min.pop(), 否则不作为
21         s_min.pop();
22     }
23     int top() {
24         return s.top();
25     }
26     //min操作
27     int min() {
28         return s_min.top();
29     }
30 private:
31     stack<int> s;
32     stack<int> s_min;
33 };

```

# Java

```
1 import java.util.Stack;
2
3 public class Solution {
4     //定义两个栈, 一个辅助栈
5     Stack<Integer> dataStack = new
Stack<Integer>();
6     Stack<Integer> minStack = new Stack<Integer>
();
7     public void push(int node) {
8         //压入
9         dataStack.push(node);
10        //Java里Stack的peek方法是返回栈顶的元素但不
移除它。但Stack的pop方法是会移除的。
11        if(minStack.isEmpty() || node <
minStack.peek())
12            minStack.push(node);
13        else
14            minStack.push(minStack.peek());
15    }
16
17    public void pop() {
18        dataStack.pop();
19        minStack.pop();
20    }
21
22    public int top() {
23        return dataStack.peek();
24    }
```

```
25
26     public int min() {
27         return minStack.peek();
28     }
29 }
```

## Python

```
1  # -*- coding:utf-8 -*-
2  class Solution:
3      def __init__(self):
4          self.arr=[]
5      def push(self, node):
6          # write code here
7          self.arr.append(node)
8      def pop(self):
9          # write code here
10         return self.arr.pop()
11     def top(self):
12         # write code here
13         return self.arr[-1]
14     def min(self):
15         # write code here
16         return min(self.arr)
```