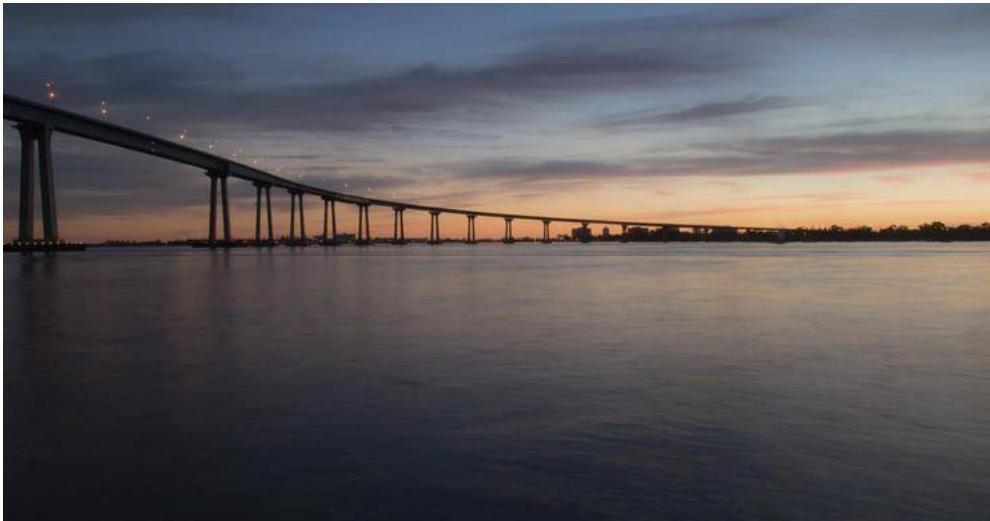


剑指Offer（五）：用两个栈实现队列

🕒 2017年11月23日 10:41:11 📄 3 🌡 9,042 °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

用两个栈来实现一个队列，完成队列的Push和Pop操作。队列中的元素为int类型。

1、思路

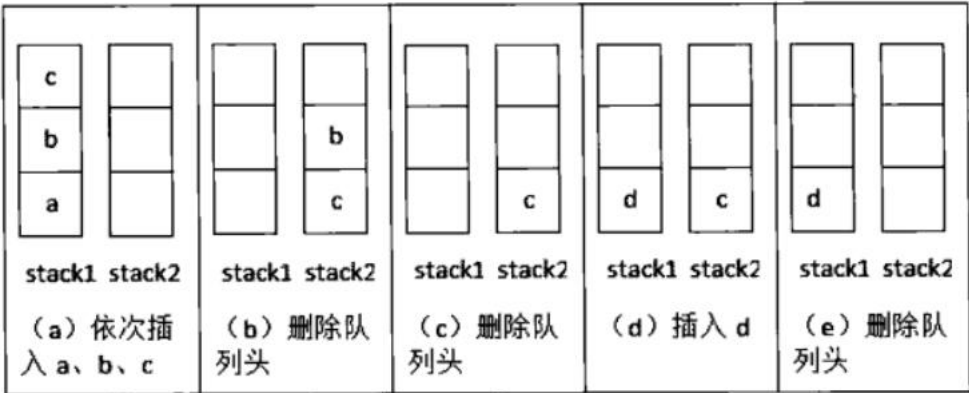
创建两个栈stack1和stack2，使用两个“先进后出”的栈实现一个“先进先出”的队列。

我们通过一个具体的例子分析往该队列插入和删除元素的过程。首先插入一个元素a，不妨先把它插入到stack1，此时stack1中的元素有{a}，stack2为空。再压入两个元素b和c，还是插入到stack1中，此时stack1的元素有{a,b,c}，其中c位于栈顶，而stack2仍然是空的。

这个时候我们试着从队列中删除一个元素。按照先进先出的规则，由于a比b、c先插入队列中，最先删除的元素应该是a。元素a存储在stack1中，但并不在栈顶，因此不能直接进行删除操作。注意stack2我们一直没有使用过，现在是让stack2发挥作用的时候了。如果我们把stack1中的元素逐个弹出压入stack2，元素在stack2中的顺序正好和原来在stack1中的顺序相反。因此经过3次弹出stack1和要入stack2操作之后，stack1为空，而stack2中的元素是{c,b,a}，这个时候就可以弹出stack2的栈顶a了。此时的stack1为空，而stack2的元素为{b,a}，其中b在栈顶。

因此我们的思路是：当stack2中不为空时，在stack2中的栈顶元素是最先进入队列的元素，可以弹出。如果stack2为空时，我们把stack1中的元素逐个出并压入stack2。由于先进入队列的元素被压倒stack1的栈底，经过弹出和压入之后就处于stack2的栈顶，有可以直接弹出。如果有新元素d插入，我们直接把它压入stack1即可。

流程示意图：



2、代码

C++:

```
1 class Solution
2 {
3 public:
4     void push(int node) {
5         stack1.push(node);
6     }
7
8     int pop() {
9         if(stack2.empty()){
10             while(stack1.size() > 0){
11                 int data = stack1.top();
12                 stack1.pop();
13                 stack2.push(data);
14             }
15         }
16         int pop_element = stack2.top();
17         stack2.pop();
18         return pop_element;
19     }
20
21 private:
22     stack<int> stack1;
23     stack<int> stack2;
24 };
```

C+

Python2.7:

对于python来讲，栈就是用list实现的。

```
1 # -*- coding:utf-8 -*-
2 class Solution:
3     def __init__(self):
4         self.stack1 = []
5         self.stack2 = []
6     def push(self, node):
7         # write code here
8         self.stack1.append(node)
9     def pop(self):
10        # return xx
11        if len(self.stack2) == 0:
12            while self.stack1:
13                self.stack2.append(self.stack1.pop())
14        return self.stack2.pop()
```

Pytho



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。

不在其位，不谋其政。--- 孔子