

## 0524

### 如何通过一个不均匀的硬币得到公平的结果

问题描述：有一枚不均匀的硬币，抛出此硬币后，可用foo()表示其结果。已知foo()能返回0和1两个值，其概率分别为0.6和0.4。问怎么利用foo()得到另一个函数，使得返回0和1的概率均为0.5。

(1) 两次均为正面： $0.6 \times 0.6 = 0.36$

(2) 第一次正面，第二次反面： $0.6 \times 0.4 = 0.24$

(3) 第一次反面，第二次正面： $0.4 \times 0.6 = 0.24$

(4) 两次均为反面： $0.4 \times 0.4 = 0.16$

用 coin()表示能产生不同概率的函数（即硬币） 设计算法

```
def makecoin():  
    while True:  
        a = coin() # 可能是正面或反面  
        if a != coin():  
            return a  
    }
```

## 232. 用栈实现队列

请你仅使用两个栈实现先入先出队列。队列应当支持一般队列支持的所有操作（push、pop、peek、empty）：

实现 MyQueue 类：

void push(int x) 将元素 x 推到队列的末尾 int pop() 从队列的开头移除并返回元素 int peek() 返回队列开头的元素 boolean empty() 如果队列为空，返回 true；否则，返回 false

说明：

你只能使用标准的栈操作 —— 也就是只有 push to top, peek/pop from top, size, 和 is empty 操作是合法的。你所使用的语言也许不支持栈。你可以使用 list 或者 deque（双端队列）来模拟一个栈，只要是标准的栈操作即可。

来源：[LeetCode](#)

- 实现

```

class MyQueue:

    def __init__(self):
        """
        Initialize your data structure here.
        """
        self.lista = []
        self.listb = []

    def push(self, x: int) -> None:
        """
        Push element x to the back of queue.
        """
        self.lista.append(x)

    def pop(self) -> int:
        """
        Removes the element from in front of queue and returns that element.
        """
        while len(self.lista) != 0:
            self.listb.append(self.lista.pop())
        cur = self.listb.pop()
        while len(self.listb) != 0:
            self.lista.append(self.listb.pop())
        return cur

    def peek(self) -> int:
        """
        Get the front element.
        """
        while len(self.lista) != 0:
            self.listb.append(self.lista.pop())
        cur = self.listb.pop()
        self.lista.append(cur)
        while len(self.listb) != 0:
            self.lista.append(self.listb.pop())
        return cur

    def empty(self) -> bool:
        """
        Returns whether the queue is empty.
        """
        if len(self.lista) == 0 and len(self.listb) == 0:
            return True
        else:
            return False

```

执行结果: 通过 [显示详情](#)

添加备注

执行用时: **52 ms** , 在所有 Python3 提交中击败了 **5.49%** 的用户

内存消耗: **15.1 MB** , 在所有 Python3 提交中击败了 **15.52%** 的用户

刷题一下:

📌 刷题解, 分享我的刷题思路

提交结果	执行用时	内存消耗	语言	提交时间	备注
通过	52 ms	15.1 MB	Python3	2021/05/24 23:06	<div>添加备注</div>
执行出错	N/A	N/A	Python3	2021/05/24 22:57	<div>添加备注</div>
解答错误	N/A	N/A	Python3	2021/05/24 22:13	<div>添加备注</div>

```
1 class MyQueue:
2
3     def __init__(self):
4         """
5         Initialize your data structure here.
6         """
7         self.lista = []
8         self.listb = []
9
10    def push(self, x: int) -> None:
11        """
12        Push element x to the back of queue.
13        """
14        self.lista.append(x)
15
16    def pop(self) -> int:
17        """
18        Removes the element from in front of queue and returns that element.
19        """
20        while len(self.lista) != 0:
21            self.listb.append(self.lista.pop())
22        cur = self.listb.pop()
23        while len(self.listb) != 0:
24            self.lista.append(self.listb.pop())
25        return cur
26
27    def peek(self) -> int:
28        """
29        Get the front element.
30        """
31        while len(self.lista) != 0:
32            self.listb.append(self.lista.pop())
33        cur = self.listb.pop()
34        self.lista.append(cur)
35        while len(self.listb) != 0:
36            self.lista.append(self.listb.pop())
37        return cur
38
```