

剑指Offer（五十）：数组中重复的数字

🕒 2018年1月21日 10:28:30 📄 9 🌡 6,616 °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

在一个长度为n的数组里的所有数字都在0到n-1的范围内。数组中某些数字是重复的，但不知道有几个数字是重复的。也不知道每个数字重复几次。请找出数组中任意一个重复的数字。例如，如果输入长度为7的数组{2,3,1,0,2,5,3}，那么对应的输出是第一个重复的数字2。

1、思路

还可以把当前序列当成是一个下标和下标对应值是相同的数组（时间复杂度为O(n),空间复杂度为O(1)）；遍历数组，判断当前位的值和下标是否相等

- 若相等，则遍历下一位；
- 若不等，则将当前位置i上的元素和a[i]位置上的元素比较：若它们相等，则找到了第一个相同的元素；若不等，则将它们两交换。换完之后a[i]位置上的值和它的下标是对应的，但i位置上的元素和下标并不一定对应；重复2的操作，直到当前位置i的值也为i，将i向后移一位，再重复2。

本文采用思路3，如果还是不懂，看下面的实例分析就懂了！

举例说明：{2,3,1,0,2,5,3}

- 0(索引值)和2(索引值位置的元素)不相等，并且2(索引值位置的元素)和1(以该索引值位置的元素2为索引值的位置的元素)不相等，则交换位置，数组变为：{1,3,2,0,2,5,3}；
- 0(索引值)和1(索引值位置的元素)仍然不相等，并且1(索引值位置的元素)和3(以该索引值位置的元素1为索引值的位置的元素)不相等，则交换位置，数组变为：{3,1,2,0,2,5,3}；
- 0(索引值)和3(索引值位置的元素)仍然不相等，并且3(索引值位置的元素)和0(以该索引值位置的元素3为索引值的位置的元素)不相等，则交换位置，数组变为：{0,1,2,3,2,5,3}；
- 0(索引值)和0(索引值位置的元素)相等，遍历下一个元素；
- 1(索引值)和1(索引值位置的元素)相等，遍历下一个元素；
- 2(索引值)和2(索引值位置的元素)相等，遍历下一个元素；
- 3(索引值)和3(索引值位置的元素)相等，遍历下一个元素；
- 4(索引值)和2(索引值位置的元素)不相等，但是2(索引值位置的元素)和2(以该索引值位置的元素2为索引值的位置的元素)相等，则找到了第一个重复的元素。

2、代码

C++：

```
1 class Solution {
2 public:
3     // Parameters:
4     //     numbers:    an array of integers
5 }
```


```
5 //      length:      the length of array numbers
6 //      duplication: (Output) the duplicated number in the array number
7 // Return value:      true if the input is valid, and there are some duplications in the array number
8 //      otherwise false
9 bool duplicate(int numbers[], int length, int* duplication) {
10     // 非法输入
11     if(numbers == NULL || length <= 0){
12         return false;
13     }
14     // 非法输入
15     for(int i = 0; i < length; i++){
16         if(numbers[i] < 0 || numbers[i] > length - 1){
17             return false;
18         }
19     }
20     // 遍历查找第一个重复的数
21     for(int i = 0; i < length; i++){
22         while(numbers[i] != i){
23             if(numbers[i] == numbers[numbers[i]]){
24                 *duplication = numbers[i];
25                 return true;
26             }
27             swap(numbers[i], numbers[numbers[i]]);
28         }
29     }
30     return false;
31 }
32 ;;
```

Python：

	Pytho
<pre>1 # -*- coding:utf-8 -*- 2 class Solution: 3 # 这里要特别注意~找到任意重复的一个值并赋值到duplication[0] 4 # 函数返回True/False 5 def duplicate(self, numbers, duplication): 6 # write code here 7 n = len(numbers) 8 if n == 0: 9 return False 10 for i in range(n): 11 if numbers[i] < 0 or numbers[i] > n-1: 12 return False 13 for i in range(n): 14 while numbers[i] != i: 15 if numbers[i] == numbers[numbers[i]]: 16 duplication[0] = numbers[i] 17 return True 18 numbers[numbers[i]], numbers[i] = numbers[i], numbers[numbers[i]] 19 return False</pre>	

个人感觉最优的方法：题目里写了数组里数字的范围保证在0 ~ n-1 之间，所以可以利用现有数组设置标志，当一个数字被访问过后，可以设置对应位的数 + n，之后再遇到相同的数时，会发现对应位上的数已经大于等于n了，那么直接返回这个数即可。

	Pytho
<pre>1 # -*- coding:utf-8 -*- 2 class Solution: 3 # 这里要特别注意~找到任意重复的一个值并赋值到duplication[0] 4 # 函数返回True/False 5 def duplicate(self, numbers, duplication): 6 # write code here 7 n = len(numbers) 8 if n == 0: 9 return False 10 for i in range(n): 11 index = numbers[i] 12 if index >= n: 13 index -= n 14 if numbers[index] >= n: 15 duplication[0] = index 16 return True 17 numbers[index] += n 18 return False</pre>	



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。

芝兰生于幽林，不以无人而不芳；君子修道立德，不为穷困而改节。--- 孔子