

剑指03 数组中重复的数字

评论：

这道题在原书上绝对不是简单级别啊！

它考察的是程序员的沟通能力，先问面试官要时间/空间需求！！

只是时间优先就用字典，

还有空间要求，就用指针+原地排序数组，

如果面试官要求空间O(1)并且不能修改原数组，还得写成二分法！！

几种方法：

- 1.先排序，然后检查相邻元素是否相同。很慢。
- 2.哈希表
- 3.原地哈希

```
1 // 我的第一次题解
2 // 时间和内存都只击败了17%的人
3
4 class Solution {
5 public:
6     int findRepeatNumber(vector<int>& nums) {
7         map<int, int> nmap;
8         int index = 0;
9         for (; index < nums.size(); index++) {
10             if (nmap.count(nums[index]) != 0) break;
11             nmap[nums[index]] = index;
12         }
13         return nums[index];
14     }
15 };
16
17
```

```
1 // K神 方法一：简单的哈希表
2 // 改为了unordered_map 和 bool，时间空间也只进步了一点点
3 class Solution {
4 public:
5     int findRepeatNumber(vector<int>& nums) {
6         unordered_map<int, bool> map;
7         for(int num : nums) {
8             if(map[num]) return num;
9             map[num] = true;
10        }
11        return -1;
12    }
13 };
14
15 // K神 方法二：原地交换
16 // 首先需要想到可以利用原数组空间来构建原地hash。
17 //
18 class Solution {
```

```
19 public:
20     int findRepeatNumber(vector<int>& nums) {
21         int i = 0;
22         while (i < nums.size()) {
23             if (nums[i] == i) {
24                 i++;
25                 continue;
26             }
27             else if (nums[nums[i]] == nums[i]) {
28                 return nums[i];
29             }
30             else {
31                 swap(nums[i], nums[nums[i]]);
32             }
33         }
34         return 0;
35     }
36 };
37
38
```