

LeetCode_两数之和

原创

xyzso1z

2020-05-25 00:42:47

139

★ 收藏

编辑 版权

分类专栏: # LeetCode

文章标签: LeetCode第一题

两数之和

两数之和最优解

LeetCode

LeetCode两数之和

题目

来源：力扣（LeetCode）

给定一个整数数组 `nums` 和一个目标值 `target`，请你在该数组中找出和为目标值的那两个整数，并返回它们的数组下标。

你可以假设每种输入只会对应一个答案。但是，数组中同一个元素不能使用两遍。
示例:

```
1 给定 nums = [2, 7, 11, 15], target = 9
2
3 因为 nums[0] + nums[1] = 2 + 7 = 9
4 所以返回 [0, 1]
```

第一版(简单、直接):

两层循环(与上学时输出乘法口诀算法类似)

```
1  class Solution {
2      public int[] twoSum(int[] nums, int target) {
3          int[] result = new int[2];
4          for (int i = 0; i < nums.length - 1; i++) {
5              for (int j = i + 1; j < nums.length; j++) {
6                  if (nums[i] + nums[j] == target) {
7                      result[0] = i;
8                      result[1] = j;
9                      break;
10             }
11         }
12     }
13     return result;
14 }
15 }
```

结果：

执行用时：85 ms (战胜20%多，作为有追求的人肯定不满足)

内存消耗：39.7 MB

第二版(想办法减少循环层数):

一次循环+用map记录已循环的值及下标

```
1  class Solution {
2      public int[] twoSum(int[] nums, int target) {
3          int[] result = new int[2];
4          Map<Integer, Integer> map = new HashMap<Integer, Integer>(nums.le
5              int length = nums.length;
6              Integer index;
7              for (int i = 0; i < length; i++) {
8                  index = map.get(Integer.valueOf(target - nums[i]));
9                  if (index != null) {
10                     result[0] = index;
11                     result[1] = i;
12                     break;
13                 } else {
14                     map.put(nums[i], i);
15                 }
16             }
17             return result;
18         }
19     }
```

结果：

执行用时：2 ms(战胜99.65%，很好奇剩下0.35%的人怎么实现的)

内存消耗：40 MB

有更好解决方案的朋友可以评论交流学习。