

题目描述

评论 (8.7k)

题解 (13.4k)

提交记录

Java

智能模式

1. 两数之和

难度 简单 11712 ☆

给定一个整数数组 `nums` 和一个整数目标值 `target`，请你在该数组中找出和为目标值 `target` 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，数组中同一个元素在答案里不能重复出现。

你可以按任意顺序返回答案。

示例 1:

输入: `nums = [2,7,11,15]`, `target = 9`

输出: `[0,1]`

解释: 因为 `nums[0] + nums[1] == 9`，返回 `[0, 1]`。

示例 2:

输入: `nums = [3,2,4]`, `target = 6`

输出: `[1,2]`

示例 3:

输入: `nums = [3,3]`, `target = 6`

输出: `[0,1]`

提示:

- $2 \leq \text{nums.length} \leq 10^4$
- $-10^9 \leq \text{nums}[i] \leq 10^9$
- $-10^9 \leq \text{target} \leq 10^9$
- 只会存在一个有效答案

进阶: 你可以想出一个时间复杂度小于 $O(n^2)$ 的算法吗?

通过次数 2,310,915 | 提交次数 4,471,060

请问您在哪类招聘中遇到此题?

```
1 class Solution {
2     public int[] twoSum(int[] nums, int target) {
3         int n=nums.length;
4         int[] temp=new int[2];
5         for(int i=0;i<n;i++){
6             for(int j=i+1;j<n;j++){
7                 if(nums[i]+nums[j]==target){
8                     temp[0]=i;
9                     temp[1]=j;
10                }
11            }
12        }
13        return temp;
14
15        int n=nums.length;
16        Map<Integer,Integer> map=new HashMap<>();
17        for(int i=0;i<n;i++){
18            map.put(nums[i],i);
19        }
20        for(int i=0;i<n;i++){
21            int temp=target-nums[i];
22            if(map.containsKey(temp)&&map.get(temp)!=i){
23                return new int[]{i,map.get(temp)};
24            }
25        }
26        return new int[]{-1,-1};
27    }
28 }
```

测试用例

代码执行结果

调试器 Beta

已完成 执行用时: 0 ms

输入

[2,7,11,15]
9

输出

[1,0]

预期结果

[0,1]