两数之和

题目 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，并且你不能使用两次相同的元素。 你可以按任意顺序返回答案。

示例 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]

代码 class Solution { public: vector twoSum(vector& nums, int target) { // 创建哈希表：key存储数值，value存储对应的索引 unordered_map<int, int> m; vector res; int len = nums.size();

```
    for (int i = 0; i < len; i++) {
        // 检查 target - nums[i] 是否在哈希表中
        if (m.find(target - nums[i]) != m.end())
            // 如果找到，返回两个数的索引
            return {m[target - nums[i]], i};
        // 将当前数字和索引存入哈希表
        m[nums[i]] = i;
    }

    return {};
}
```

};

复杂度分析 时间复杂度：O(n)，其中 n 为 nums 的长度。 空间复杂度：O(n)，哈希表需要 O(n) 的空间。

解释 很多涉及到 **两个变量** 的题目，都可以枚举其中一个变量，把它当成常量看待，从而转化成 **一个变量** 的问题。 代码实现时，通常来说 **枚举右，寻找左** 是更加好写的。

知识点 数组、哈希表