

0001. 两数之和

👤 ITCharge ⌚ 大约 2 分钟

- 标签：数组、哈希表
- 难度：简单

题目链接

- [0001. 两数之和 - 力扣](#)

题目大意

描述： 给定一个整数数组 *nums* 和一个整数目标值 *target*。

要求： 在该数组中找出和为 *target* 的两个整数，并输出这两个整数的下标。可以按任意顺序返回答案。

说明：

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

示例：

- 示例 1：

输入：nums = [2,7,11,15], target = 9

输出：[0,1]

解释：因为 $\text{nums}[0] + \text{nums}[1] == 9$ ，返回 [0, 1]。

py

- 示例 2：

输入：nums = [3,2,4], target = 6

输出：[1,2]

py

解题思路

思路 1：枚举算法

1. 使用两重循环枚举数组中每一个数 $nums[i]$ 、 $nums[j]$ ，判断所有的 $nums[i] + nums[j]$ 是否等于 $target$ 。
2. 如果出现 $nums[i] + nums[j] == target$ ，则说明数组中存在和为 $target$ 的两个整数，将两个整数的下标 i 、 j 输出即可。

思路 1：代码

```
class Solution:
    def twoSum(self, nums: List[int], target: int) -> List[int]:
        for i in range(len(nums)):
            for j in range(i + 1, len(nums)):
                if i != j and nums[i] + nums[j] == target:
                    return [i, j]
        return []
```

py

思路 1：复杂度分析

- 时间复杂度： $O(n^2)$ ，其中 n 是数组 $nums$ 的元素数量。
- 空间复杂度： $O(1)$ 。

思路 2：哈希表

哈希表中键值对信息为 $target - nums[i] : i$ ，其中 i 为下标。

1. 遍历数组，对于每一个数 $nums[i]$ ：
 1. 先查找字典中是否存在 $target - nums[i]$ ，存在则输出 $target - nums[i]$ 对应的下标和当前数组的下标 i 。
 2. 不存在则在字典中存入 $target - nums[i]$ 的下标 i 。

思路 2：代码

py

```
def twoSum(self, nums: List[int], target: int) -> List[int]:
    numDict = dict()
    for i in range(len(nums)):
        if target-nums[i] in numDict:
            return numDict[target-nums[i]], i
        numDict[nums[i]] = i
    return [0]
```

思路 2：复杂度分析

- 时间复杂度： $O(n)$ ，其中 n 是数组 *nums* 的元素数量。
- 空间复杂度： $O(n)$ 。