

算法笔记

麦业民

1 简介

这是本人刷[LeetCode](#)所留笔记。其中涵盖了各种算法设计思想：

- 动态规划
- 双指针
- 二分法

以及各种数据结构：

- 数组（字符串）
- 散列表
- 栈

算法皆以 Python3 实现。简单题只做简略描述。

2 题目

2.1 三数之和

此题官方难度为中等，实际中等偏难，通过率为 38.65%，运用了双指针、散列表。

问题描述

给你一个整数数组 `nums`，判断是否存在三元组 `[nums[i], nums[j], nums[k]]` 满足 $i \neq j$ 、 $i \neq k$ 且 $j \neq k$ ，同时还满足 $nums[i] + nums[j] + nums[k] == 0$ 。请你返回所有和为 0 且不重复的三元组。

注意：答案中不可以包含重复的三元组。

示例

示例 1:

```
输入: nums = [-1,0,1,2,-1,-4]
输出: [[-1,-1,2],[-1,0,1]]
解释:
nums[0] + nums[1] + nums[2] = (-1) + 0 + 1 = 0 。
nums[1] + nums[2] + nums[4] = 0 + 1 + (-1) = 0 。
nums[0] + nums[3] + nums[4] = (-1) + 2 + (-1) = 0 。
不同的三元组是 [-1,0,1] 和 [-1,-1,2] 。
注意，输出的顺序和三元组的顺序并不重要。
```

示例 2:

```
输入: nums = [0,1,1]
输出: []
解释: 唯一可能的三元组和不为 0 。
```

示例 3:

```
输入: nums = [0,0,0]
输出: [[0,0,0]]
解释: 唯一可能的三元组和为 0 。
```

思路

固定其中一个数，将其转换为两数之和问题（参见[2.2](#)）。由于两数之和问题要求输入有序且只有唯一解，所以我们需要对上述输入进行排序。

2.2 两数之和

abc