**15. 3Sum**

Given an array nums of *n* integers, are there elements *a*, *b*, *c* in nums such that *a* + *b* + *c* = 0? Find all unique triplets in the array which gives the sum of zero.

给定一个n位整型数的数组，是否存在三个数字a, b, c满足a+b+c=0？找出数组中所有满足条件的三元祖

**Note:**

The solution set must not contain duplicate triplets.

结果集中不能包含重复的三元组

**Example:**

Given array nums = [-1, 0, 1, 2, -1, -4],

A solution set is:

[

[-1, 0, 1],

[-1, -1, 2]

]

解题思路：

1. 首先，拿到这道题，我第一个想法是，遍历每一个数，取其相反数作为target。在剩下的数字当中，找出两个数之和为target的集合。时间复杂度为O(n)2就拿示例来说，这样做存在重复的组合，如：[-1, 0, 1], [0, 1, -1]。

因此，就必须考虑去从的问题。为了去除重复，考虑用一个Set集合来保存已经添加的三元组，但是在查重的时候，必须要遍历Set中每一个三元组的每一个元素，这样就造成时间复杂度为O(n)3。这样肯定会超时，因此这个思路不可行。

1. 于是尝试用网上的思路。

首先将数组进行排序（涉及到数组的问题都可以优先考虑排序），得到排序后的数组：{x0, x1, …, xn-1}

1. 同样的针对每一个数字取相反数为target，查找剩余数字中和为target的数字。这里我们不在逐个遍历每一个数，那样会造成TLE（Time Limited Exceeded）。
2. 首先我们令a = xi, t=-a, p、q两个指针分别指向xi+1，xn-1。要满足等式

a + b + c = 0 ⑴

那么就要找到满足等式[p] + [q] = t这样的两个数。在移动p和q两个指针的过程中，p递增，q递减。p和q之和存在三种可能：

1. [p] + [q] > t 此时q左移（减少总和）
2. [p] + [q] < t 此时p右移（增大总和）
3. [p] + [q] = t 此时满足等式(1)，即为所求解
4. p = q时结束查找过程
5. 注：[q]表示q指向的元素的值
6. 在遍历过程中，会遇到重复的三元组。
7. 对于第一个数a，假设存在下标i，满足xi=xi+1，那么包含xi的三元祖与包含xi+1的三元祖是等价的。此时需要跳过重复的数字，直到找到第一个下标j，满足。
8. 当确定了第一个下标i之后，指针p和q指向的数字中仍然包含重复的元素。此时，我们只需要简单的跳过重复数字即可。
9. 优化点：
10. 对于第一个数a，只需要遍历到倒数第3个数即可（因为是排好序的，a再往右走就找不到满足条件的数字了）。
11. 当第一个数字a > 0时，停止遍历。因为有序数组，a大于0时，右边的数字都大于0。

*p.s.：本文根据网上搜索的解题思路加上自己的理解写成，如有不当之处，欢迎指正。*