# [三数之和](https://leetcode-cn.com/explore/interview/card/bytedance/243/array-and-sorting/1020/)

**头条重点**

## 题目

给定一个包含 n 个整数的数组 nums ，判断 nums 中是否存在三个元素 a，b，c ，使得 a + b + c = 0 ？找出所有满足条件且不重复的三元组。

注意：答案中不可以包含重复的三元组。

例如, 给定数组 nums = [-1, 0, 1, 2, -1, -4]，

满足要求的三元组集合为：
[
 [-1, 0, 1],
 [-1, -1, 2]
]

## 解题思路

1. 将数组排序
2. 固定一位数，然后通过两个指针对撞，寻找总和为 0 的三个数

public static List<List<Integer>> threeSum(int[] nums) {
 if (nums.length < 3) {
 return Collections.emptyList();
 }

 Set<List<Integer>> res = new HashSet<>();
 Arrays.sort(nums);

 int zCount = 0;
 for (int num : nums) {
 if (num == 0) {
 zCount++;
 }
 }

 for (int i = 0; i < nums.length && nums[i] < 0; i++) {
 int first = nums[i];
 int j = i + 1;
 int k = nums.length - 1;
 while (j < k) {
 int t = nums[j] + nums[k] + first;
 if (t == 0) {
 List<Integer> list = new ArrayList<>();
 list.add(first);
 list.add(nums[j]);
 list.add(nums[k]);

 res.add(list);
 j++;
 k--;
 } else if (t > 0) {
 k--;
 } else {
 j++;
 }
 }

 }

 if (zCount >= 3) {
 List<Integer> list = new ArrayList<>();
 list.add(0);
 list.add(0);
 list.add(0);
 res.add(list);
 }
 return new ArrayList<>(res);
}