# 题目

给你一个包含 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 ？请你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。

**示例：**

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]

]

# 分析

## 方法一：排序+双指针

**思路：**

先排序，在排序的基础上，虽然也是O(n^2)复杂度，但可以利用双指针尽量提高效率。

**代码：**

class Solution {

public:

vector<vector<int>> threeSum1(vector<int>& nums){

vector<vector<int>>res;

sort(nums.begin(),nums.end());

if(nums.size()<3)return res;

int l,r;

for(int i=0;i<nums.size();i++){

if(nums[i]>0)break;

//如果当前数字大于0，则三数之和一定大于0，所以结束循环

if(i>0&&nums[i]==nums[i-1])continue;//去重

l=i+1;

r=nums.size()-1;

while(l<r){

int sum=nums[i]+nums[l]+nums[r];

if(sum==0){

res.push\_back(vector<int>{nums[i],nums[l],nums[r]});

while (l<r && nums[l] == nums[l+1]) l++;

// 去重

while (l<r && nums[r] == nums[r-1]) r--;

// 去重

l++;

r--;

}

else if(sum>0)r-=1;

else l+=1;

}

}

return res;

}

};

## 方法二：哈希表

vector<vector<int>> threeSum(vector<int>& nums) {

vector<vector<int>>result;

vector<int>num;

unordered\_map<int, int> M;

for (int i = 0; i < nums.size();i++){

if (M.find(nums[i]) == M.end())

{

M[nums[i]] = 1;

num.push\_back(nums[i]);

}

else

M[nums[i]]++;

}

sort(num.begin(), num.end());

int firstoutput = 1;

for (int i = 0; i < num.size(); i++)

{

for (int j = i; j < num.size();j++){

if(j==i&&M[num[j]]==1)

continue;

if(-num[i]-num[j]<num[j])

continue;

if (M.find(-num[i] - num[j]) != M.end())

{

if (-num[i] - num[j]==num[j]){

if(num[j]==num[i]){

if(M[num[j]]<=2)continue;

}

else{

if(M[num[j]]==1)continue;

}

}

result.push\_back(vector<int>{num[i],num[j],-num[i] - num[j]});

}

}

}

return result;

}