

15. 3Sum

Medium,
Array, Two Pointers.

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.

Note:

The solution set must not contain duplicate triplets.

Example:

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

```
A solution set is:
```

```
[  
  [-1, 0, 1],  
  [-1, -1, 2]  
]
```

解法一：

该方法为two pointer。

首先需要对数组进行排序。

分左基准点和右边的双指针。左基准点如果是大于0则和不可能等于0。

如果左基准点小于等于零，右侧可以用双指针算法。

时间复杂度 $O(n\log n)$ 。

java

```
class Solution {  
    public List<List<Integer>> threeSum(int[] nums) {  
        int len = nums.length;  
        Set<List<Integer>> res = new HashSet<>();  
        Arrays.sort(nums);
```

```

if(len==0 || nums[0]>0) return new ArrayList(res);
for(int i=0; i<len-2;i++){
    if(nums[i]<=0){
        int left = i+1;
        int right = len-1;
        while(left<right){
            int target = 0-nums[i];
            int sum = nums[left]+nums[right];
            if(sum==target){
                while(left<right && nums[left]==nums[left+1])left++;
                while(left<right && nums[right]==nums[right-1]) right--;
                List<Integer> tmp = Arrays.asList(nums[i],nums[left],nums[right]);
                res.add(tmp);
                left++;
                right--;
            }
            else if(sum>target) right--;
            else left++;
        }
    }
    else break;
}
return new ArrayList(res);
}
}

```