# 题目

输入一个递增排序的数组和一个数字s，在数组中查找两个数，使得它们的和正好是s。如果有多对数字的和等于s，则输出任意一对即可。

**示例 1：**

**输入：**nums = [2,7,11,15], target = 9

**输出：**[2,7] 或者 [7,2]

**示例 2：**

**输入：**nums = [10,26,30,31,47,60], target = 40

**输出：**[10,30] 或者 [30,10]

**限制：**

1 <= nums.length <= 10^5

1 <= nums[i] <= 10^6

**类似题目：Leetcode 1**

# 分析

## 方法一：双指针

**思路：**

从第一个和最后一个元素开始算。

如果算出来的结果比目标大，那就最后一个往前移，否则就第一个目标往后移。

**代码：**

class Solution {

public:

vector<int> twoSum(vector<int>& nums, int target) { // 双指针

vector<int> result;

int sum, i = 0, j = nums.size()-1;

while(i < j){

sum = nums[i]+nums[j];

if(sum==target){

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

result.push\_back(nums[j]);

break;

}

if(sum>target) j--;

else i++;

}

return result;

}

};