# 题目

给定一个已按照升序排列的整数数组numbers ，请你从数组中找出两个数满足相加之和等于目标数target 。

函数应该以长度为2的整数数组的形式返回这两个数的下标值。numbers 的下标从0开始计数 ，所以答案数组应当满足0 <= answer[0] < answer[1] < numbers.length 。

假设数组中存在且只存在一对符合条件的数字，同时一个数字不能使用两次。

示例 1：

输入：numbers = [1,2,4,6,10], target = 8

输出：[1,3]

解释：2 与 6 之和等于目标数 8 。因此 index1 = 1, index2 = 3 。

示例 2：

输入：numbers = [2,3,4], target = 6

输出：[0,2]

示例 3：

输入：numbers = [-1,0], target = -1

输出：[0,1]

提示：

2 <= numbers.length <= 3 \* 104

-1000 <= numbers[i] <= 1000

numbers 按 非递减顺序 排列

-1000 <= target <= 1000

仅存在一个有效答案

注意：本题与主站 167 题相似（下标起点不同）：https://leetcode.cn/problems/two-sum-ii-input-array-is-sorted/

# 分析

这类计算一组数组中的一些数相关的特性时，一般都可以采用暴力法、哈希表、双指针去实现。

代码：

class Solution {

public:

vector<int> twoSum(vector<int>& numbers, int target) {

int low = 0,high = numbers.size()-1;

vector<int> ret;

while(low < high)

{

int sum = numbers[low] + numbers[high];

if(sum == target)

{

ret.push\_back(low);

ret.push\_back(high);

return ret;

}

else if(sum < target)

{

low++;

}

else

{

high--;

}

}

return ret;

}

};