

## 两数之和 II - 输入有序数组

给定一个已按照 **非递减顺序排列** 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 1 开始计数**，所以答案数组应当满足  $1 \leq \text{answer}[0] < \text{answer}[1] \leq \text{numbers.length}$ 。

你可以假设每个输入 **只对应唯一的答案**，而且你 **不可以** 重复使用相同的元素。

**示例 1:**

输入: `numbers = [2,7,11,15]`, `target = 9`

输出: `[1,2]`

解释: 2 与 7 之和等于目标数 9。因此 `index1 = 1`, `index2 = 2`。

**示例 2:**

输入: `numbers = [2,3,4]`, `target = 6`

输出: `[1,3]`

**示例 3:**

输入: `numbers = [-1,0]`, `target = -1`

输出: `[1,2]`

```
class Solution:
    def twoSum(self, numbers: List[int], target: int) -> List[int]:
        #双指针
        left=0
        right=len(numbers)-1
        while left<right:
            sum_num=numbers[left]+numbers[right]
            if(sum_num==target):
                return [left+1,right+1]
            if(sum_num<target):
                left+=1
            else:
                right-=1
        return [-1,-1]
```