**难度: 简单**

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

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/he-wei-sde-liang-ge-shu-zi-lcof

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

思路(双指针):

1). 定义一个0下标和一个数组最后长度的下标

2). 循环判断两个下标的值的和是否等于要找的值

3). 大于的话,说明将右边的-1,小于的话将左边+1,相等直接返回这两个下标的值

|  |
| --- |
| public int[] twoSum(int[] nums, int target) {  int left = 0;  int right = nums.length -1;  while(left < right) {  if (nums[left] + nums[right] < target) {  left++;  }else if (nums[left] + nums[right] > target) {  right--;  }else {  return new int[]{nums[left],nums[right]};  }  }  return new int[]{};  } |