```
/*
给定一个已按照升序排列 的有序数组，找到两个数使得它们相加之和等于目标数。
函数应该返回这两个下标值 index1 和 index2，其中 index1 必须小于 index2。
说明:
    返回的下标值（index1 和 index2）不是从零开始的。
    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:
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 。
来源：力扣（LeetCode）
链接：https://leetcode-cn.com/problems/two-sum-ii-input-array-is-sorted
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。
*/
```

分析:
看着标签是二分,结果用的双指针索引.

- 方法一:双指针索引分别从头(最小)和尾向(最大)中央逼近,直到找到符合条件的就返回这两个索引.
- 方法二:与001_两数之和一样,利用map数据结构简化查找,但需要$O(n)$的额外空间开销,其中n是$numbers$的长度.

方法一:C++_双指针索引法

```
class Solution
{
    public:
        vector<int> twoSum(vector<int>& numbers, int target)
        {
            vector<int> ret_val;
            int left    =   1               ;
            int right   =   numbers.size()  ;
            int mid     =   0               ;
            while(left < right)
            {
                if(numbers[left-1] + numbers[right-1] == target)
                {
                    ret_val.push_back(left);
                    ret_val.push_back(right);
                    break;
                }
                else if(numbers[left-1] + numbers[right-1] < target)
                {
                    left++;
                }
                else
                {
                    right--;
                }
            }
```

```
            return ret_val;
        }
};

/*
执行结果：
通过
显示详情
执行用时 :4 ms, 在所有 cpp 提交中击败了97.11% 的用户
内存消耗 :9.4 MB, 在所有 cpp 提交中击败了79.15%的用户
*/
```

方法二:C++_map查找法

```
class Solution
{
    public:
        vector<int> twoSum(vector<int>& numbers, int target)
        {
            vector<int>     ret_val             ;
            map<int,int>    mii                 ;
            int             i           =   0   ;

            for(i = 0 ; i < numbers.size() ; i++)
            {
                if(mii.count(numbers[i]))
                {
                    ret_val.push_back(mii[numbers[i]]+1);
                    ret_val.push_back(i+1);
                    break;
                }
                else
                {
                    mii[target-numbers[i]] = i;
                }

            }
            return ret_val;
        }
};

/*
执行结果：
通过
显示详情
执行用时 :4 ms, 在所有 cpp 提交中击败了97.11% 的用户
内存消耗 :9.6 MB, 在所有 cpp 提交中击败了12.46%的用户
*/

```

AlimyBreak
2019.11.10