

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 。

思路，和找数字一样，直接map标记位置查找

```
class Solution {
public:
    vector<int> twoSum(vector<int>& numbers, int target) {
        map<int,int> bj;
        int n=numbers.size();
        vector<int> result;
        for(int i=0;i<n;++i){
            if(bj[target-numbers[i]]>0){
                result.push_back(bj[target-numbers[i]]);
                result.push_back(i+1);
                return result;
            }
            bj[numbers[i]]=i+1;
        }
    }
};
```