汇总区间

给定一个无重复元素的有序整数数组 nums 。

返回 **恰好覆盖数组中所有数字** 的 **最小有序** 区间范围列表。也就是说，nums 的每个元素都恰好被某个区间范围所覆盖，并且不存在属于某个范围但不属于 nums 的数字 x 。

列表中的每个区间范围 [a,b] 应该按如下格式输出：

- "a->b" ，如果 a != b
- "a" ，如果 a == b

示例 1：

输入：nums = [0,1,2,4,5,7]

输出：["0->2","4->5","7"]

解释：区间范围是：

[0,2] --> "0->2"

[4,5] --> "4->5"

[7,7] --> "7"

示例 2：

输入：nums = [0,2,3,4,6,8,9]

输出：["0","2->4","6","8->9"]

解释：区间范围是：

[0,0] --> "0"

[2,4] --> "2->4"

[6,6] --> "6"

[8,9] --> "8->9"

示例 3：

输入：nums = []

输出：[]

示例 4：

输入： nums = [-1]

输出： ["-1"]

示例 5：

输入： nums = [0]

输出： ["0"]

```
class Solution {
public:
/*
使用 双指针，i 指向每个区间的起始位置，j 从 i 开始向后遍历直到不满足连续递增（或 j 达到数组边界），则当前区间结束；然后将 i 指向更新为 j + 1，作为下一个区间的开始位置，j 继续向后遍历找下一个区间的结束位置，如此循环，直到输入数组遍历完毕。
*/
    vector<string> summaryRanges(vector<int>& nums) {
        vector<string> res;
        int i=0;
        for(int j=0;j<nums.size();j++)
        {
            if(j+1==nums.size()||nums[j]+1!=nums[j+1])//当不满足 1 递增时结束记录范围  打印范围区间
            {
                string temp;
                temp.append(to_string(nums[i]));
                if(i!=j)
                {
                    temp.append("->");
                    temp.append(to_string(nums[j]));
                }
                res.push_back(temp);
                i=j+1;//更新下一次区间开始位置
            }
        }
        return res;
    }
};
```