# 题目

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"

提示：

0 <= nums.length <= 20

-2^31 <= nums[i] <= 2^31 - 1

nums 中的所有值都 互不相同

nums 按升序排列

# 分析

## 方法一：双指针

思路：

可以使用双指针来遍历数组。初始时，左右指针都指向数组的第一个元素。然后，向右移动右指针，直到找到一个位置，使得当前位置的值与下一个位置的值不连续（即 nums[right] + 1 != nums[right + 1]）。此时，如果左指针等于右指针，则说明只有一个数字，将该数字的字符串形式添加到结果中；否则，将左指针和右指针之间的数字范围表示为字符串，并添加到结果中。

代码：

class Solution {

public:

vector<string> summaryRanges(vector<int>& nums) {

vector<string> result; // 用于存放结果的数组

int n = nums.size(); // 数组的长度

if (n == 0) { // 如果数组为空，则直接返回空结果数组

return result;

}

int left = 0, right = 0; // 定义左右指针，初始都指向数组的第一个元素

while (right < n) { // 右指针小于数组长度时循环

if (right + 1 < n && nums[right] + 1 == nums[right + 1]) { // 如果右指针的下一个元素与当前元素连续

right++; // 右指针右移

} else { // 如果不连续

if (left == right) { // 如果左右指针相等，说明只有一个数字

result.push\_back(to\_string(nums[left])); // 将该数字的字符串形式添加到结果中

} else { // 否则表示有一个范围

result.push\_back(to\_string(nums[left]) + "->" + to\_string(nums[right])); // 将范围表示为字符串添加到结果中

}

left = right = right + 1; // 更新左右指针到下一个范围的起始位置

}

}

return result; // 返回结果数组

}

};