**难度: 简单**

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"]

提示：

0 <= nums.length <= 20

-231 <= nums[i] <= 231 - 1

nums 中的所有值都 互不相同

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/summary-ranges

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:**

代码有详细注释

|  |
| --- |
| public List<String> summaryRanges(int[] nums) {  List<String> list = new ArrayList<>();  for(int start = 0, end = 0; end < nums.length; end++) {  //避免越界,并且判断当前的数组+1,是否等于下一个数组的值  if(end + 1 < nums.length && nums[end] + 1 == nums[end+1]) {  continue;  }  //说明上面的不成立,也就是不连续  if (start == end) {  list.add(nums[start]+"");  }else {  //说明有一段连续的,但是下一个不连续  list.add(nums[start]+"->"+nums[end]);  }  //把那个下一个不连续的过度一下  start = end + 1;  }  return list;  } |