# 题目

给你一个正整数n，请你找出符合条件的最小整数，其由重新排列n中存在的每位数字组成，并且其值大于n。如果不存在这样的正整数，则返回-1。

注意，返回的整数应当是一个32位整数 ，如果存在满足题意的答案，但不是32位整数 ，同样返回-1。

示例 1：

输入：n = 12

输出：21

示例 2：

输入：n = 21

输出：-1

提示：

1 <= n <= 231 – 1

类似题目：

31. 下一个排列的官方题解

# 分析

## 方法一：两次遍历

思路：

把n转换成字符串（字符数组），那么本题实际上是在求字符数组的31.下一个排列，当不存在下一个排列时返回−1。

代码：

class Solution {

public:

int nextGreaterElement(int n) {

// 转换为数组或字符串处理

auto nums = to\_string(n);

int i = nums.size() - 2;

while (i >= 0 && nums[i] >= nums[i + 1]) {

i--;

}

if (i < 0) return -1;

int j = nums.size() - 1;

while (j >= 0 && nums[i] >= nums[j]) {

j--;

}

swap(nums[i], nums[j]);

reverse(nums.begin() + i + 1, nums.end());

long ans = stol(nums);

return ans > INT\_MAX ? -1 : ans;

}

};

复杂度分析

时间复杂度：O(logn)。

空间复杂度：O(logn)。