

31 下一个排列

Label: 数组、双指针

实现获取 下一个排列 的函数，算法需要将给定数字序列重新排列成字典序中下一个更大的排列。

如果不存在下一个更大的排列，则将数字重新排列成最小的排列（即升序排列）。

必须 原地 修改，只允许使用额外常数空间。

输入: `nums = [1,2,3]`

输出: `[1,3,2]`

- 转为字符数组，遍历

```
class Solution {
    public void nextPermutation(int[] nums) {
        int i = nums.length - 2;

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

        if (i >= 0) {
            int j = nums.length - 1;
            while (j >= 0 && nums[i] >= nums[j]) { // 找到右侧大于i的第一个数
                j--;
            }
            swap(nums, i, j);
        }
        reverse(nums, i + 1); // 从 i+1开始反转
    }

    public void swap(int[] nums, int i, int j) {
        int temp = nums[i];
        nums[i] = nums[j];
        nums[j] = temp;
    }

    public void reverse(int[] nums, int start) {
        int left = start, right = nums.length - 1;
        while (left < right) {
            swap(nums, left, right);
            left++;
            right--;
        }
    }
}
```