1. Next Permutation

Medium

Implement **next permutation**, which rearranges numbers into the lexicographically next greater permutation of numbers.

If such arrangement is not possible, it must rearrange it as the lowest possible order (ie, sorted in ascending order).

The replacement must be **in-place** and use only constant extra memory.

Here are some examples. Inputs are in the left-hand column and its corresponding outputs are in the right-hand column.

1,2,3` → `1,3,2`  
`3,2,1` → `1,2,3`  
`1,1,5` → `1,5,1

**解**

对于数组，将从后往前枚举，在中找出最小的并且大于的数，交换，然后将排序

class Solution {  
public:  
 void nextPermutation(vector<int>& nums) {  
 for(int i = nums.size() - 1; i >= 0; --i){  
 int minNum = INT\_MAX, index;  
 for(int j = i + 1; j < nums.size(); ++j){  
 if(nums[j] > nums[i] && nums[j] < minNum){  
 minNum = nums[j];  
 index = j;  
 }  
 }  
 if(minNum != INT\_MAX){  
 swap(nums[index], nums[i]);  
 sort(nums.begin() + i + 1, nums.end());  
 return;  
 }  
 }  
 sort(nums.begin(), nums.end());  
 return;  
 }  
};