/\*

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, do not allocate 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

思路：//思路，一位一位往前换，找到比前面大的就交换，然后后面的排个序！

\*/

class Solution {

public:

void nextPermutation(vector<int>& nums)

{

if(nums.size()<2)

return;

int kk=nums.size()-2;

while(kk>=0)

{

for(int i=nums.size()-1;i>kk;i--)

{

if(nums[i]>nums[kk])

{

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

sort(nums.begin()+kk+1,nums.end());

return;

}

}

kk--;

}

if(kk=-1)

sort(nums.begin(),nums.end());

return;

}

};