

```

public class Solution {
    public void nextPermutation(int[] nums) {
        if (nums == null || nums.length <= 1) return;

        int i;
        for (i = nums.length-1; i > 0; i--){
            if (nums[i] > nums[i-1]){
                break;
            }
        }
        if (i != 0) {
            swap(nums, i-1);
        }

        reverse(nums, i);
    }

    private void swap(int[] a, int index){
        for (int i = a.length-1; i > index; i--){
            if (a[i] > a[index]){
                int tmp = a[i];
                a[i] = a[index];
                a[index] = tmp;
                break;
            }
        }
    }

    private void reverse(int[] nums, int index){
        int left = index;
        int right = nums.length - 1;

        while(left < right) {
            int temp = nums[left];
            nums[left] = nums[right];
            nums[right] = temp;
            left++;
            right--;
        }
    }
}

```

/\*

在当前序列中，从尾端往前寻找两个相邻元素，前一个记为first，后一个记为second，并且满足first 小于 second。

然后再从尾端寻找另一个元素number，如果满足first 小于number，

即将第first个元素与number元素对调，并将second元素之后（包括second）的所有元素颠倒排序，即求出下一个序列

example:

6，3，4，9，8，7，1

此时 first = 4，second = 9

从尾巴到前找到第一个大于first的数字，就是7

交换4和7，即上面的swap函数，此时序列变成6，3，7，9，8，4，1

再将second = 9以及以后的序列重新排序，让其从小到大排序，使得整体最小，即reverse一下  
(因为此时肯定是递减序列)

得到最终的结果：6，3，7，1，4，8，9

\*/