1. First Missing Positive

Given an unsorted integer array, find the smallest missing positive integer.

**Example 1:**

Input: [1,2,0]  
Output: 3

**Example 2:**

Input: [3,4,-1,1]  
Output: 2

**Example 3:**

Input: [7,8,9,11,12]  
Output: 1

**Note:**

Your algorithm should run in *O*(*n*) time and uses constant extra space.

**解**

缺失的正数的范围在1 ~ nums.size() + 1之间

解法1 开一个bool数组，记录1 ~ nums.size() + 1是否出现

解法2 利用nums数组本身，将正数调整至正确的位置，比如nums[3] = 4（如果4在nums中），然后寻找最小的不满足nums[i] = i + 1的i

class Solution {  
public:  
 int firstMissingPositive(vector<int>& nums) {  
 int n = nums.size();  
 for(int i = 0; i < n; ++i){  
 while(nums[i] != i + 1){  
 //不在合法区间或者已经调整好的数字不用调整  
 if(nums[i] <= 0 || nums[i] > n || nums[i] == nums[nums[i] - 1])break;  
 int temp = nums[i];  
 nums[i] = nums[temp - 1];  
 nums[temp - 1] = temp;  
 }  
   
 }  
 int ans = n + 1;  
 for(int i = 0; i < n; ++i){  
 if(nums[i] != i + 1){  
 ans = i + 1;  
 break;  
 }  
 }  
 return ans;  
 }  
};