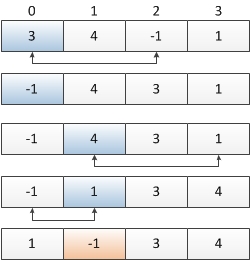
思路：

虽然不能再另外开辟非常数级的额外空间，但是可以在输入数组上就地进行swap操作。

思路：交换数组元素，使得数组中第i位存放数值(i+1)。最后遍历数组，寻找第一个不符合此要求的元素，返回其下标。整个过程需要遍历两次数组，复杂度为O(n)。

下图以题目中给出的第二个例子为例，讲解操作过程。



首先思路上，其次临界条件，这题和下面题异曲同工：

n个元素的数组，里面的数都是0~n-1范围内的，求数组中重复的某一个元素，没有返回-1, 要求时间性能O(n) 空间性能O(1)。

/\*\*\*

\* 思路是:追赶法

\* \*/

class Solution {

public:

int firstMissingPositive(vector<int>& nums) {

int lent = nums.size();

if(lent==0){return 1;}

int pos=0;

while(pos<lent){

if(nums[pos]>0 && nums[pos]!=pos+1 && nums[pos]-1<lent && nums[pos]!=nums[nums[pos]-1]){

swap(nums[pos],nums[nums[pos]-1]);

}

else{ pos++;}

}

for(int i=0;i<lent;i++){

if(nums[i]!=i+1)

return i+1;

}

return lent+1;

}

};