41. First Missing Positive

Hard

1308488FavoriteShare

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

class Solution {

public:

int firstMissingPositive(vector<int>& nums) {

int len=nums.size();

bool one\_present=false;

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

int curr=nums[i];

if(curr==1) one\_present=true;

if(curr<1||curr>len) nums[i]=1;

}

if(!one\_present) return 1;

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

int p=nums[i]<0?(-1\*nums[i]):nums[i];

nums[p-1]=nums[p-1]>0?(-1\*nums[p-1]):nums[p-1];

}

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

if(nums[i]>0) return i+1;

}

return len+1;

}

};

Success

[Details](https://leetcode.com/submissions/detail/205904953/)

Runtime: 0 ms, faster than 100.00% of C++ online submissions for First Missing Positive.

Memory Usage: 778.2 KB, less than 89.77% of C++ online submissions forFirst Missing Positive.