41. First Missing Positive

Hard | Topics | Companies | Hint

Given an unsorted integer array `nums`, return the smallest missing positive integer.

You must implement an algorithm that runs in `O(n)` time and uses `O(1)` auxiliary space.

Example 1:

```
Input: nums = [1,2,0]
Output: 3
Explanation: The numbers in the range [1,2] are all in the array.
```

Example 2:

```
Input: nums = [3,4,-1,1]
Output: 2
Explanation: 1 is in the array but 2 is missing.
```

Example 3:

```
Input: nums = [7,8,9,11,12]
Output: 1
Explanation: The smallest positive integer 1 is missing.
```

Constraints:

- `1 <= nums.length <= 10^5`
- `-2^31 <= nums[i] <= 2^31 - 1`

Approach 1: using unordered map

→ insert all array elements into a set.
→ now start checking from i : 1 to n exists in array or not by using set

```
for( i : 1 to n)
    if( set doesn't has i)
        return i

return array size + 1
```

Q: How is it the loop is from 1 to n?

An array of size n can contain max n positives and as per our problem we need to give first missing positive. So in worst case our i/p array can contain values from 1 to n. and hence first missing positive is n+1.

T(n): O(n) + O(n)
S(n): O(n)

Approach 2:

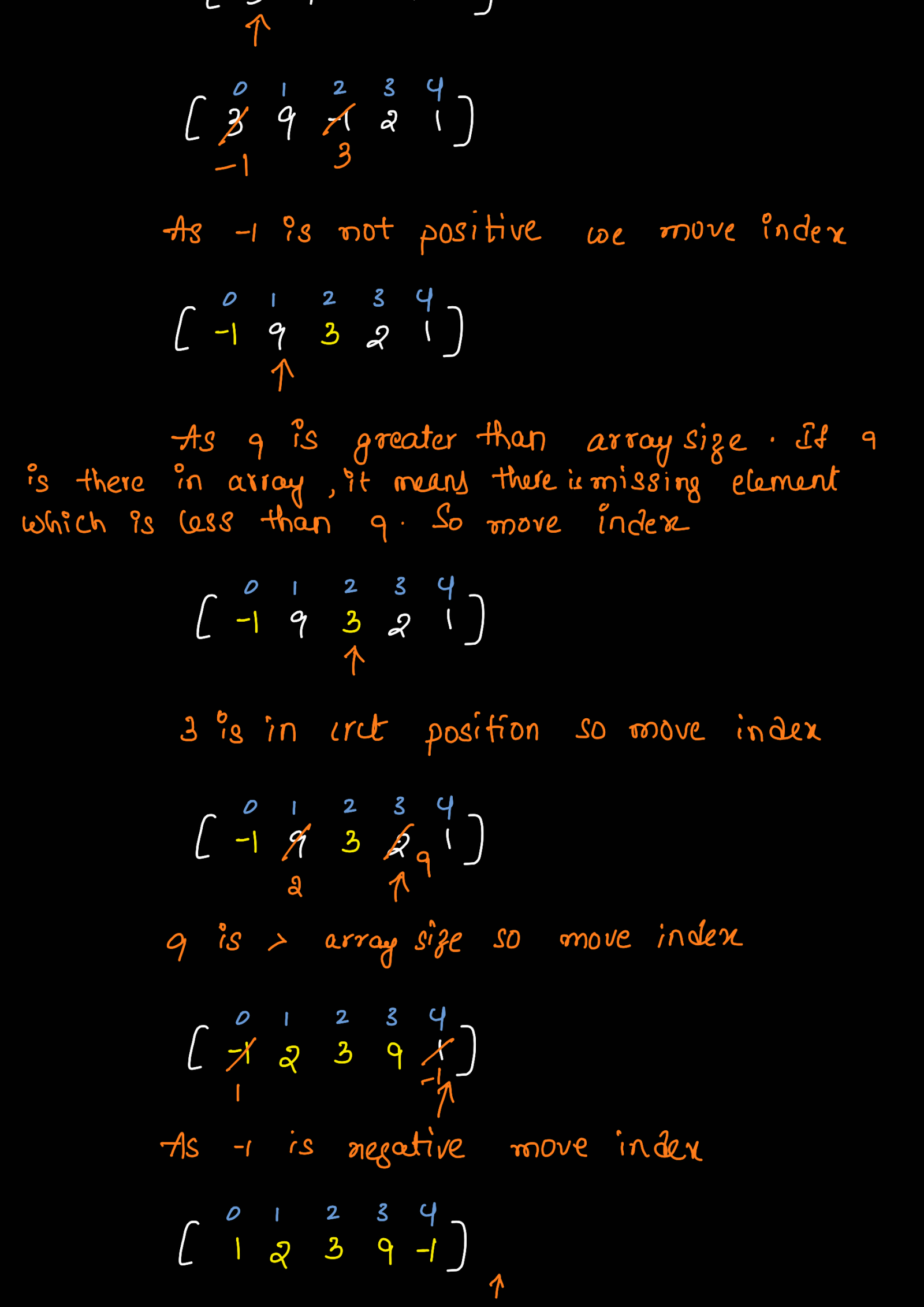
If we see the above approach, we are using an explicit map to check if a number is existing in array or not.

Here the hint we need to catch is " we are supposed to check if i: 1 to n exists in the array. so **why can't we use array indices to mark that element i: 1 to n is present in array or not.**"

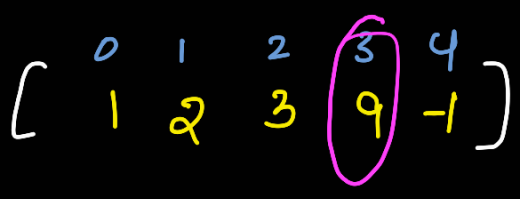
0	1	2	3	4
3	9	-1	2	1

keep an element i at index i-1.

0	1	2	3	4
3	9	-1	2	1


0 1 2 3 4
3 9 -1 2 1
-1 3
As -1 is not positive we move index
0 1 2 3 4
-1 9 3 2 1
As 9 is greater than array size. If 9 is there in array, it means there is missing element which is less than 9. So move index
0 1 2 3 4
-1 9 3 2 1
3 is in rct position so move index
0 1 2 3 4
-1 9 3 2 1
2 9
9 is > array size so move index
0 1 2 3 4
-1 2 3 9 1
1 -1
As -1 is negative move index
0 1 2 3 4
1 2 3 9 -1

now find first index where $i \neq a_i - 1$



So return 3+1 = 4

```
class Solution {
public:
    int firstMissingPositive(vector<int>& nums) {
        for(int i=0;i<nums.size();i++){
            while(nums[i]>0 && nums[i]<=nums.size() && nums[nums[i]-1] != nums[i])
                swap(nums[i],nums[nums[i]-1]);
        }

        for(int i=0;i<nums.size();i++){
            if(nums[i]!=i+1) return i+1;
        }

        return nums.size()+1;
    }
};
```

O(n)

O(n)

T(n) : O(n)
S(n) : O(1)