268. Missing Number

Easy 3458 2604 Add to List Share

Given an array `nums` containing `n` distinct numbers in the range `[0, n]`, return *the only number in the range that is missing from the array.*

Follow up: Could you implement a solution using only `O(1)` extra space complexity and `O(n)` runtime complexity?

Example 1:

```
Input: nums = [3,0,1]
Output: 2
Explanation: n = 3 since there are 3 numbers, so all numbers are in the range [0,3]. 2 is the missing number in the range since it does not appear in nums.
```

Example 2:

```
Input: nums = [0,1]
Output: 2
Explanation: n = 2 since there are 2 numbers, so all numbers are in the range [0,2]. 2 is the missing number in the range since it does not appear in nums.
```

Example 3:

```
Input: nums = [9,6,4,2,3,5,7,0,1]
Output: 8
Explanation: n = 9 since there are 9 numbers, so all numbers are in the range [0,9]. 8 is the missing number in the range since it does not appear in nums.
```

Example 4:

```
Input: nums = [0]
Output: 1
Explanation: n = 1 since there is 1 number, so all numbers are in the range [0,1]. 1 is the missing number in the range since it does not appear in nums.
```

Constraints:

- `n == nums.length`
- `1 <= n <= 10^4`
- `0 <= nums[i] <= n`
- All the numbers of `nums` are **unique**.

```python
class Solution:
    def missingNumber(self, nums: List[int]) -> int:

        n = len(nums)
        range_list = [x for x in range(0,n+1)]
        nums_set = set(nums)
        range_set = set(range_list)

        res = range_set.difference(nums_set)
        print(res)
        for i in res :
            return i

```

Testcase | Run Code Result | Debugger

Accepted Runtime: 37 ms

Your input [0,1]

stdout {2}

Output 2 Diff

Expected 2

Console Use Example Testcases Run Code Submit