- Given an array of distinct numbers in range [0, n]
  return only number missing in array.

Since array can contain values [0 ... n] inclusive, since the array only has 'n' items, we know there is exaclty one value missing within this range.

Ex.) Input: [3, 0, 2]
Output: 1

Approach # 1

1.) Sort array

2.) Iterating for loop starting @ 0 → n inclusive
When Iter does not match value,
return Iter

3.) If all values match, it is n.

Time: $O(n \log n)$ (Debatable) Space: $O(n)$

Approach # 2

1.) Create a set, insert all items in input. $O(n)$

2.) Iterating for loop, check if iter exists in set $O(n)$
If not, iter is answer

3.) There will be a value not existing in set.

Time: $O(n)$ Space: $O(n)$

Can we improve space?.

To improve space, we have to utilize input

Input: [3, 0, 2]

[3, 0, 2]
[-3, 0, 2]
[-3, 0, -2]

[4, 2, 1, 0]
[4, 2, 1, 0]
[4, 2, -1, 0]
[4 -2 -1 0]
[-4 -2 -1 0]

[3 4 1 0]
[3 4 1 0]
[3 4 1 0]
[3 -4 1 0]
[-3 -4 1 0]

[4 0 1 3]
[4 0 1 3]
[-4 0 1 3]
[-4 0 1 -3]

1.) Iterate over each item in input
If input[i] < n then make input[input[i]] negative
This marks it as visited

2.) Iterate again. Find first value > 0. The index is answer.

3.) If none found, n is answer.

Time: $O(n)$ Space: $O(1)$

Input: [9 6 4 2 3 5 7 0 1]   Output: 8

✗ ✗ ✗ ✗ ✗ ✗ ✗ ✗

Another solution:

		[0, 2]	[0, 1]
Pot. Sum:	$\frac{n(n+1)}{2}$	3	3
Act. Sum:	accumulate(input, 0)	2	1

1.) Sum all numbers in input

2.) Get potential sum if all values are present

$$\frac{n(n+1)}{2}$$

3.) Potential Sum - Actual Sum = Answer