Given an array containing *n* distinct numbers taken from0, 1, 2, ..., n, find the one that is missing from the array.

For example,  
Given *nums* = [0, 1, 3] return 2.

**Note**:  
Your algorithm should run in linear runtime complexity. Could you implement it using only constant extra space complexity?

**Credits:**  
Special thanks to [@jianchao.li.fighter](https://leetcode.com/discuss/user/jianchao.li.fighter) for adding this problem and creating all test cases.

<https://leetcode.com/problems/missing-number/>

**public** **static** **int** MissingNumber(**int**[] nums)

        {

*/\*como el conjunto va de 0...n,*

*declaro un booleano de tamaño n + 1*

*y marco a true los que estan \*/*

**bool**[] marcas = new **bool**[nums.Length+1];

**for** (**int** i = 0; i < nums.Length; i++)

            {

                marcas[nums[i]] = **true**;

            }

*/\* luego busco al que quedo en false \*/*

**for** (**int** i = 0; i < marcas.Length; i++)

            {

**if** (!marcas[i])

                {

**return** i;

                }

            }

**return** 0;

        }