丢失的数字

给定一个包含 [0, n] 中 n 个数的数组 nums ，找出 [0, n] 这个范围内没有出现在数组中的那个数。

进阶：

- 你能否实现线性时间复杂度、仅使用额外常数空间的算法解决此问题?

示例 1：

输入：nums = [3,0,1]

输出：2

解释：n = 3，因为有 3 个数字，所以所有的数字都在范围 [0,3] 内。2 是丢失的数字，因为它没有出现在 nums 中。

示例 2：

输入：nums = [0,1]

输出：2

解释：n = 2，因为有 2 个数字，所以所有的数字都在范围 [0,2] 内。2 是丢失的数字，因为它没有出现在 nums 中。

示例 3：

输入：nums = [9,6,4,2,3,5,7,0,1]

输出：8

解释：n = 9，因为有 9 个数字，所以所有的数字都在范围 [0,9] 内。8 是丢失的数字，因为它没有出现在 nums 中。

示例 4：

输入：nums = [0]

输出：1

解释：n = 1，因为有 1 个数字，所以所有的数字都在范围 [0,1] 内。1 是丢失的数字，因为它没有出现在 nums 中。

```cpp
class Solution {
public:
    int missingNumber(vector<int>& nums) {
        int len=nums.size();
        sort(nums.begin(),nums.end());//先对其进行从小到大排序
        if(nums[0]!=0)
        {
            return 0;
        }
        else if(nums[len-1]!=len)
        {
            return len;
        }
        //对排序好的进行遍历查找缺失项
        for(int i=0;i<len-1;i++)
        {
            if(nums[i+1]-nums[i]>1)
            {
                return nums[i]+1;
            }
        }
        return -1;
    }
};
```