Given an unsorted integer array nums. Return the smallest positive integer that is not present in nums.

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.

Constraints:

1 <= nums.length <= 105

-231 <= nums[i] <= 231 - 1

Solution :

class Solution {

public:

    int firstMissingPositive(vector<int>& nums) {

        sort(nums.begin(),nums.end());

        int a=1;

        for(auto x:nums){

            if(x<a ) continue;

            else if(x==a) {a++;}

        }

        return a;

    }

};