Given an array nums of n integers where nums[i] is in the range [1, n], return *an array of all the integers in the range* [1, n] *that do not appear in* nums.

**Example 1:**

**Input:** nums = [4,3,2,7,8,2,3,1]

**Output:** [5,6]

**Example 2:**

**Input:** nums = [1,1]

**Output:** [2]

**Constraints:**

* n == nums.length
* 1 <= n <= 105
* 1 <= nums[i] <= n

class Solution {

public List<Integer> findDisappearedNumbers(int[] nums) {

List<Integer> result=new ArrayList<>();

int n=nums.length;

if(nums==null ||n==1){

return result;

}

int i=0;

while(i<n)

{

if(nums[i]!=nums[nums[i]-1]){

swap(nums,i,nums[i]-1);

}

else{

i++;

}

}

for(int j=0;j<n;j++){

if(nums[j]!=j+1){

result.add(j+1);

}

}

return result;

}

public static void swap(int []nums,int i,int j){

int temp=nums[i];

nums[i]=nums[j];

nums[j]=temp;

}

}