1236. Find All Numbers Disappeared in an Array

Given an array of integers where 1 ≤ a[i] ≤ n (n = size of array), some elements appear twice and others appear once.

Find all the elements of [1, n] inclusive that do not appear in this array.

Could you do it without extra space and in O(n) runtime? You may assume the returned list does not count as extra space.

Example

Input:  
[4,3,2,7,8,2,3,1]

Output:  
[5,6]

<https://www.lintcode.com/problem/find-all-numbers-disappeared-in-an-array/description>

1. */\**
2. *\* To change this license header, choose License Headers in Project Properties.*
3. *\* To change this template file, choose Tools | Templates*
4. *\* and open the template in the editor.*
5. *\*/*
6. **package** javaapplication66;
8. **import** java.util.ArrayList;
9. **import** java.util.HashSet;
10. **import** java.util.List;
12. ***/\*\****
13. ***\****
14. ***\* @author Usuario***
15. ***\*/***
16. **public** **class** JavaApplication66 {
18. **public** List<Integer> findDisappearedNumbers(**int**[] nums) {
19. *// write your code here*
21. HashSet<Integer> hash =
22. **new** HashSet();
24. *//int min = Integer.MAX\_VALUE;*
25. *//int max = Integer.MIN\_VALUE;*
26. **for**(**int** i =0; i<nums.length; i++) {
27. hash.add(nums[i]);
28. *// min = Math.min(min, nums[i]);*
29. *//max = Math.max(max, nums[i]);*
30. }
32. ArrayList<Integer> lista =
33. **new** ArrayList();

36. **for**(**int** i = 1; i <= nums.length; i++) {
37. **if**(!hash.contains(i)) {
38. lista.add(i);
39. }
40. hash.add(i);
41. }
43. **return** lista;
45. }
46. **public** **static** **void** main(String[] args) {
47. *// TODO code application logic here*
48. }
50. }