Cyclic Sort

11 September 2024 07:25

Cyclic Sort

Aim: To sort the array in just 1 pass

0	1	2	3	4
3	5	2	1	4

When given nos from Range 1, N ⇒ Use Cyclic Sort

VVVV I

Very Imp

Let's say an array: 3 5 2 1 4

Here, the arrays are in a jumbled orders but in the ideal world, they all will be sorted & placed at their correct indices.

After Sorting:

0	1	2	3	4
1	2	3	4	5

Observe that, here the **Index = value - 1**

↓

Why? Bcoz index starts from '0'

0	1	2	3	4
3	5	2	1	4

Take the 1st element '3'. & then swap it with its correct index.

0	1	2	3	4
2	5	3	1	4

We won't move on ahead. We will now make sure that if the swapped value (Here '2') is at its correct index or not

0	1	2	3	4
5	2	3	1	4

Now, do the same process

0	1	2	3	4

4 swaps

+ 5 swaps

$N + (N-1)$

$= (2N-1)$

$= O(N)$

↓

Now, do the same process

0	1	2	3	4
4	2	3	1	5

i

↓

Linear Time

1	2	3	4	5

→ Ans Sorted Array

now here, it'll check at every index & check whether all no's are at its correct position or not

CHECK - SWAP - MOVE

Complexity Analysis

Worst Case: It makes (N-1) swaps for the array & then it checks N more times when array is sorted

'i' is not getting incremented here & thus these additional 'N' swaps

Worst Case : O(N)

268. Missing Number

Amazon!!

Easy | Topics | Companies

Given an array `nums` containing `n` distinct numbers in the range `[0, n]`, return *the only number in the range that is missing from the array.*

Example 1:

> **Input:** nums = [3,0,1]
> **Output:** 2
> **Explanation:** n = 3 since there are 3 numbers, so all numbers are in the range [0,3]. 2 is the missing number in the range since it does not appear in nums.

Example 2:

> **Input:** nums = [0,1]
> **Output:** 2
> **Explanation:** n = 2 since there are 2 numbers, so all numbers are in the range [0,2]. 2 is the missing number in the range since it does not appear in nums.

Example 3:

> **Input:** nums = [9,6,4,2,3,5,7,0,1]
> **Output:** 8
> **Explanation:** n = 9 since there are 9 numbers, so all numbers are in the range [0,9]. 8 is the missing number in the range since it does not appear in nums.

=) nos going till 'N' =) Total 'N+1' no.s.

Ex N = 4, arr = [4, 0, 2, 1]

, ,

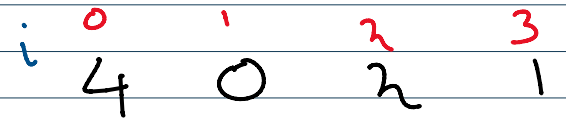
Ex N = 4, arr: [4, 0, 2, 1]

arr = [0, 1, 2, 3, 4] => array with the missing num

In the sorted version

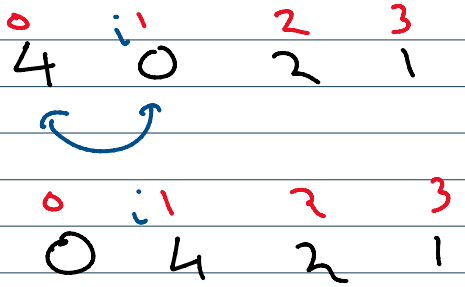
Correct index = value

Eg.

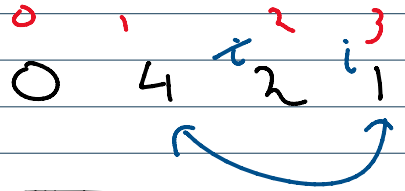


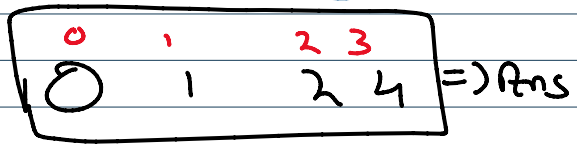
Here index No '4' does not exist. i.e index 'N' does not exist. Thus ignore that element

Is Index 4 present here? No. ignore it & go to next index



Again, '4th' index d.n.e





So ,how to find the missing number then ? . Answer is: Start with the 'start' index and check the array . The first index that you encounter which is not having its required value is the answer. In our case 'correct index = value'. So at index 0 , 0 will be there. At index 1, 1 will be there and so on

Case 2: When 'N' is not there in the Array.

N = 4 arr = [1, 0, 3, 2] —(After sorting)→ [0, 1, 2, 3] (0 1 2 3)

4 is not present in array. Then how to find it?

In this case; **Ans = N**

Tip: If range ⇒ [0, N]

✱ every element will be at index = value

If range ⇒ [1, N]

✱ every element will be at index = value - 1

448. Find All Numbers Disappeared in an Array

Google!!

Easy | Topics | Companies | Hint

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:

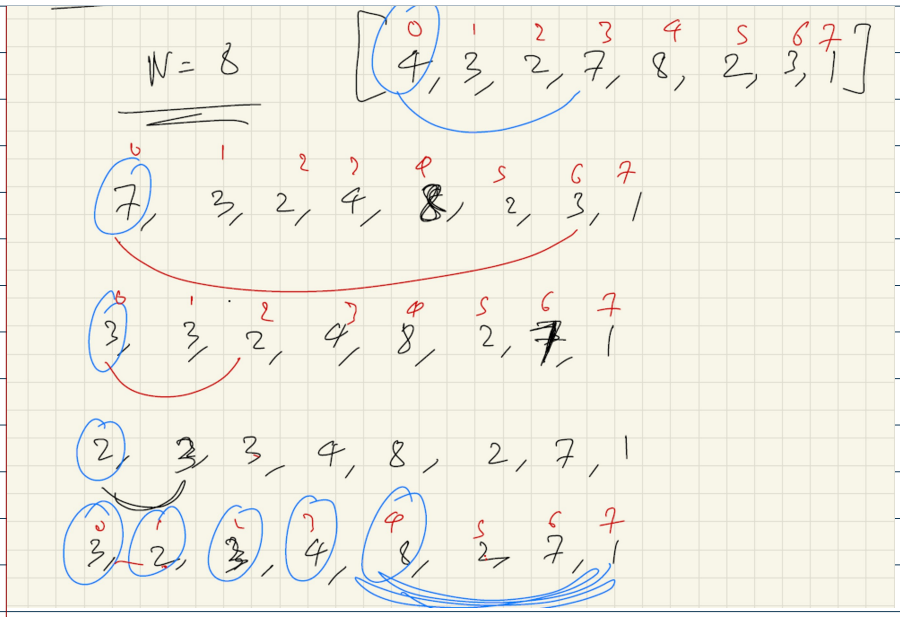
```
Input: nums = [4,3,2,7,8,2,3,1]
Output: [5,6]
```

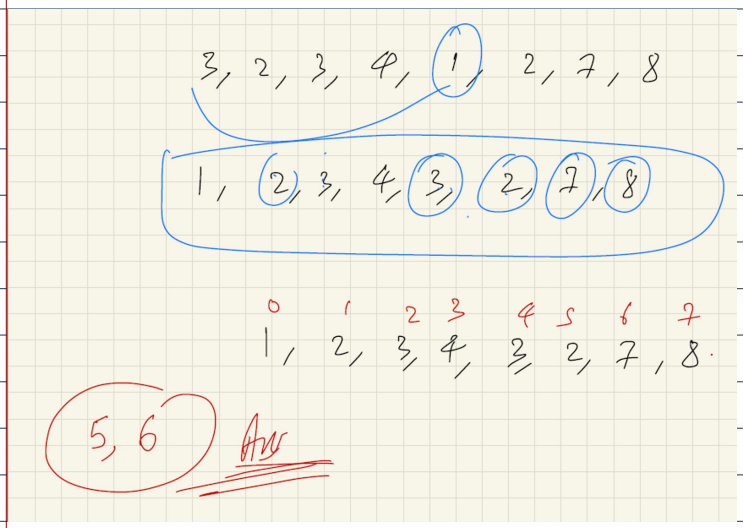
Example 2:

```
Input: nums = [1,1]
Output: [2]
```

Constraints:

- `n == nums.length`
- `1 <= n <= 10^5`
- `1 <= nums[i] <= n`





=) Check if at that index we have correct element or not. Return those indices not containing correct elements.

287. Find the Duplicate Number

Amazon/Microsoft !!

Medium | Topics | Companies

Given an array of integers `nums` containing `n + 1` integers where each integer is in the range `[1, n]` inclusive.

There is only **one repeated number** in `nums`, return *this repeated number*.

You must solve the problem **without** modifying the array `nums` and uses only constant extra space.

Example 1:

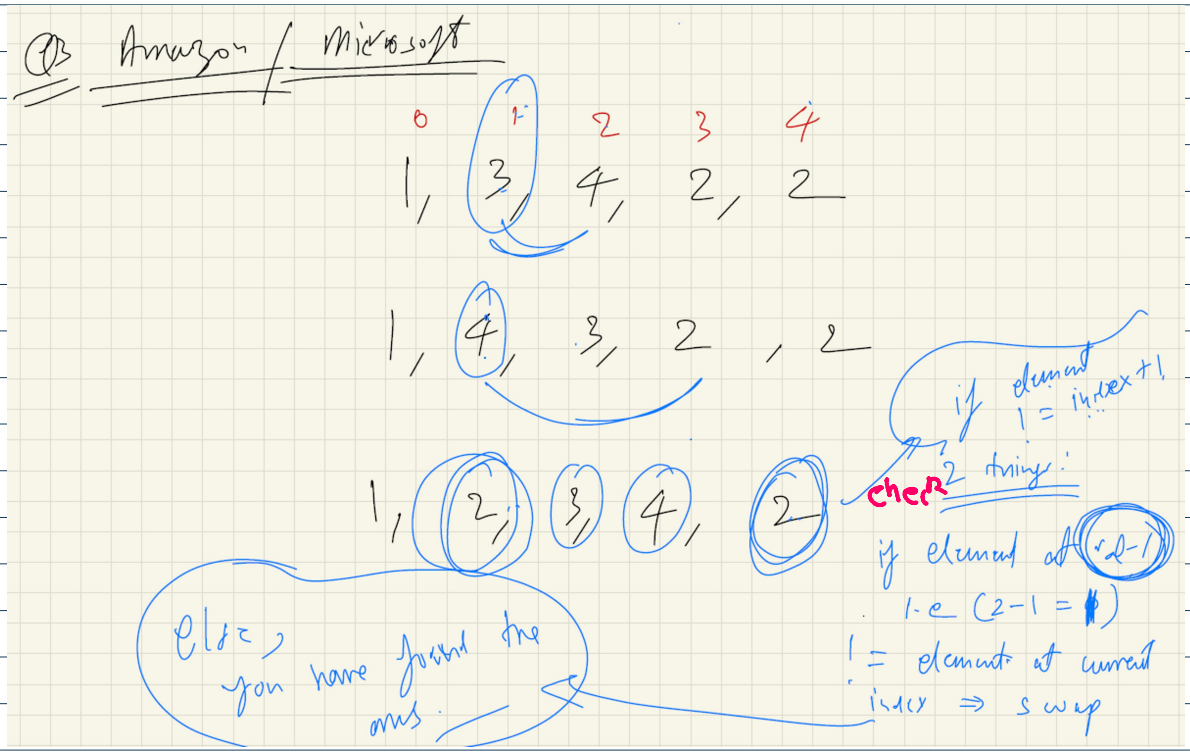
```
Input: nums = [1,3,4,2,2]
Output: 2
```

Example 2:

```
Input: nums = [3,1,3,4,2]
Output: 3
```

Example 3:

```
Input: nums = [3,3,3,3,3]
Output: 3
```



645. Set Mismatch

Easy | Topics | Companies

You have a set of integers `s`, which originally contains all the numbers from `1` to `n`. Unfortunately, due to some error, one of the numbers in `s` got duplicated to another number in the set, which results in **repetition of one** number and **loss of another** number.

You are given an integer array `nums` representing the data status of this set after the error.

Find the number that occurs twice and the number that is missing and return *them in the form of an array.*

Example 1:

```
Input: nums = [1,2,2,4]
Output: [2,3]
```

Example 2:

```
Input: nums = [1,1]
Output: [1,2]
```

1 no. is missing
1 no. is repeating

Return the missing & repeating number in an array

=> N = 6

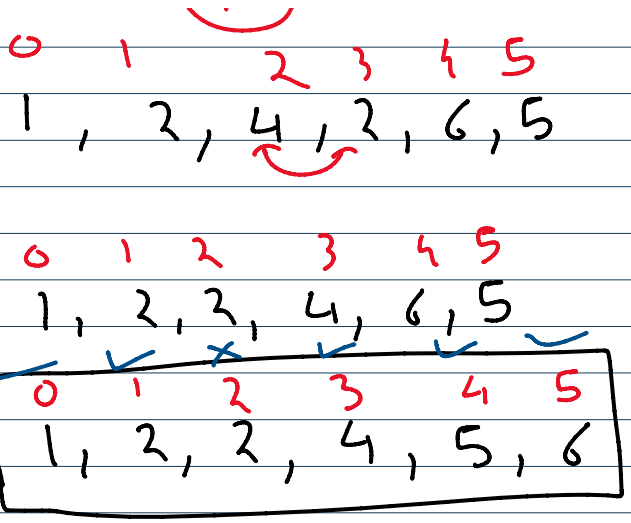
[3, 1, 4, 2, 6, 5]

// Set mismatch

0 1 2 3 4 5
[2, 1, 4, 2, 6, 5]

//

0 1 2 3 4 5



Missing = index + 1

Duplicate = no. at the missing index

Q

41. First Missing Positive

Hard | Topics | Companies | Hint

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.
```

Example 3:

```
Input: nums = [7,8,9,11,12]
Output: 1
Explanation: The smallest positive integer 1 is missing.
```

=> Eg. 3, 4, -1, 1

|| After sorting

[-1, 1, 3, 4]

Ans = 2 -> Since 2 is missing from the array

the array

Note: While calculating the missing number, ignore negative numbers Bcoz we need to find the smallest missing +ve integer.

Thus, in the above example we should start checking from 1.

Eg. [1, 2, 0] => [0, 1, 2]

Ans = 3

Ignore elements that are -ve & > N(arr.length)

0	1	2	3
3,	4,	-1,	1

-1, 4, 3, 1

-1, 1, 3, 4

0	1	2	3
1,	-1,	3,	4

=> Index + 1 = Ans of the q^n

Eg. [3, 4, 2, 1] => [1, 2, 3, 4]

Ans = 5 (N+1)