287. Find the Duplicate Number
Medium

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

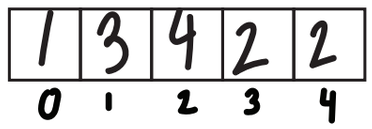
There is only one repeated number in nums, return this repeated number.

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

points:
- elements in list are [1, n] inclusive.
  eg. 1, 2, 3, 4, 5, 6, 7, ..., n
- array contains n + 1 integers.
- there is only one repeated number.
- although the input is a list, this is a linked list problem

Example 1

```
       [1,3,4,2,2]
index   0 1 2 3 4
```



Example 1:

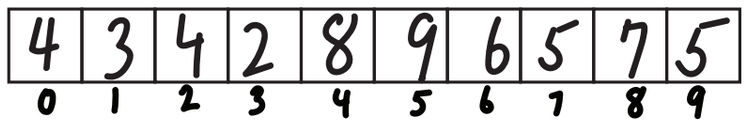
```
Input: nums = [1,3,4,2,2]
Output: 2
```

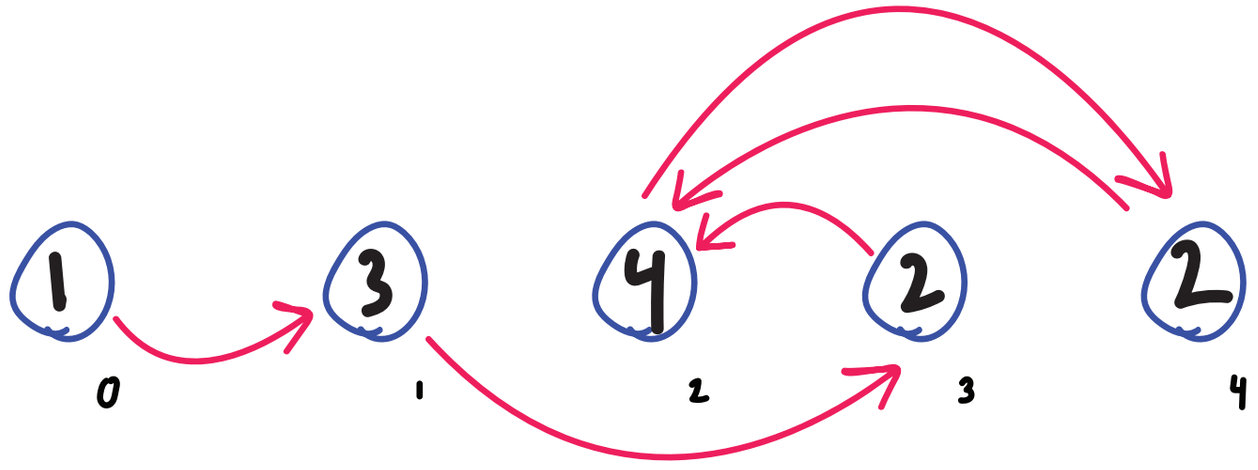
Example 2:

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

Example 3

```
       [4,3,1,2,8,9,6,5,7,5]
index   0 1 2 3 4 5 6 7 8 9
```






```
slow, fast = 0,0
while True:
    slow = nums[slow]
    fast = nums[nums[fast]]
    if slow == fast:
        break
```

nums[0]
↳ 1 ← slow
nums[nums[0]]
↳ 3 ← fast

nums[1]
↳ 3 ← slow
nums[nums[3]]
↳ 2 ← fast

nums[3]
↳ 2 ← slow
nums[nums[2]]
↳ 4 ← fast

nums[2]
↳ 4 ← slow
nums[nums[4]]
↳ 4 ← fast
slow == fast
break

```
slow2 = 0
while True:
    slow = nums[slow]
    slow2 = nums[slow2]
    if slow == slow2:
        return slow
```

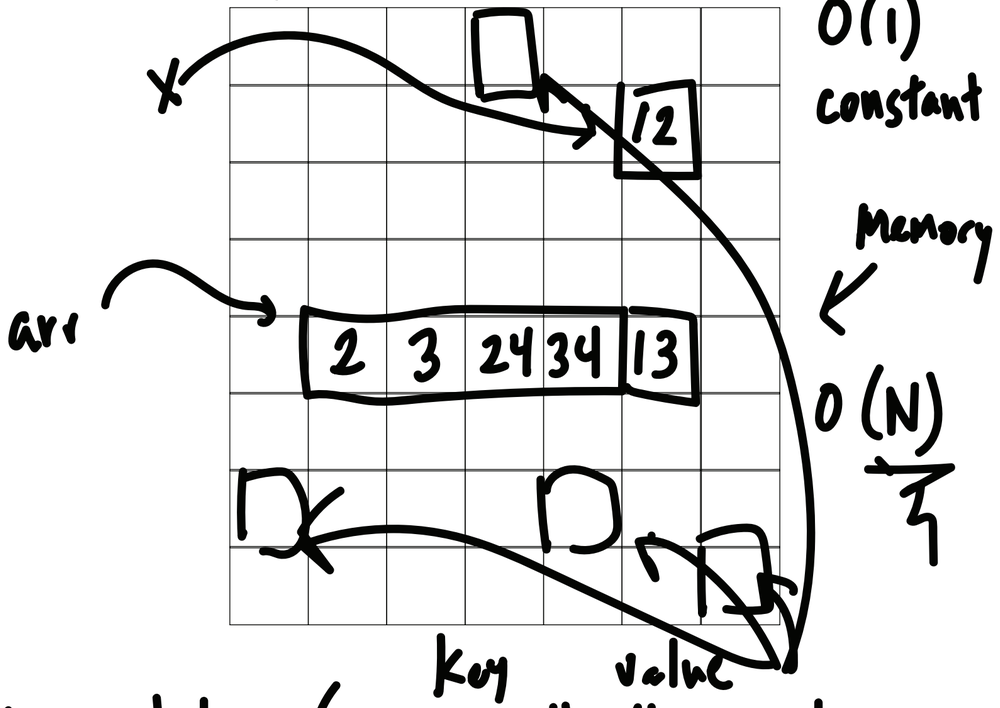
nums[4]
↳ 2 ← slow
nums[0]
↳ 1 ← slow2

nums[2]
↳ 4 ← slow
nums[1]
↳ 3 ← slow2

nums[4]
↳ 2 ← slow
nums[3]
↳ 2 ← slow2
slow == slow2
return slow

1. Using Floyd's algorithm to find the first intersection (slow == fast)
2. Leave slow pointer (forget fast pointer)
3. Introduce slow2, where both slow and slow2 increment by 1.
4. The intersection (slow == slow2) is the answer. return slow

```
let x = 34
let arr = [2,3,24,34]
x = 12
arr.push(13)
```



Space

O(1)
constant

Memory

O(N)

```
let obj = { "name": "Jared",
            "age": 20 }.
            "location": "Paris",
                .
                .
```

O(N)

O(1) →
```
let num = 5
let answer = num + 2
console.log(answer)
```
→ arr.pop()

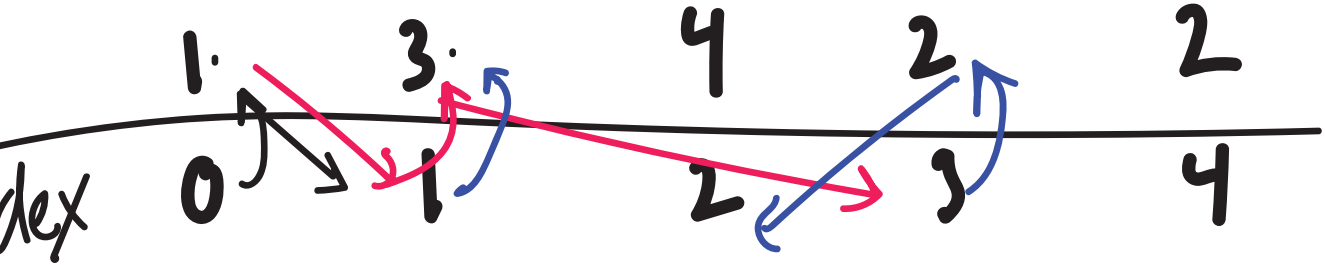
```
for (let i=0 ; i<arr.length ; i++){
    for (let j=0; j<arr.length; j++){
        let arr = [1,2,3,4,5,6,...50]
        for (let i=0; i<arr.length; i++){
            console.log(arr[j])
        }
    }
```

O(N²)
time complexity
O(?)
O(n)

```
j = 0 ~ j < arr.length
j = 0 ~ j < arr.length
```
↓
i = 0, 1, 2
↓
arr = [0,1,2,3,4,5]

Example 1

[1,3,4,2,2]



slow = 0

fast = 0

slow = nums[slow]

fast = nums[nums[fast]]