141. Linked List Cycle

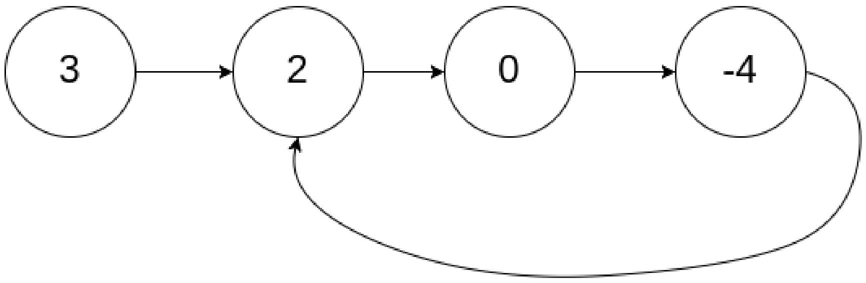
Created	@October 30, 2022 3:11 PM
Difficulty	Easy
LC Url	https://leetcode.com/problems/linked-list-cycle/
Importance	
Tag	LinkedList NEET Two pointers
Video	

Given `head`, the head of a linked list, determine if the linked list has a cycle in it.

There is a cycle in a linked list if there is some node in the list that can be reached again by continuously following the `next` pointer. Internally, `pos` is used to denote the index of the node that tail's `next` pointer is connected to. **Note that `pos` is not passed as a parameter**.

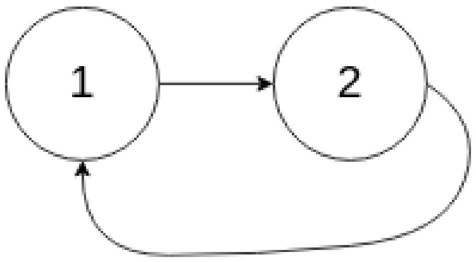
Return `true` *if there is a cycle in the linked list*. Otherwise, return `false`.

Example 1:



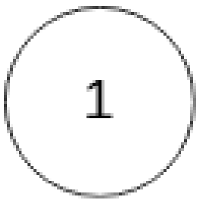
```
Input: head = [3,2,0,-4], pos = 1
Output: true
Explanation: There is a cycle in the linked list, where the tail connects to the 1st node
 (0-indexed).
```

Example 2:



```
Input: head = [1,2], pos = 0
Output: true
Explanation: There is a cycle in the linked list, where the tail connects to the 0th node.
```

Example 3:



```
Input: head = [1], pos = -1
Output: false
Explanation: There is no cycle in the linked list.
```

Constraints:

- The number of the nodes in the list is in the range `[0,` 10^4 `]`.
- 10^5 `<= Node.val <=` 10^5
- `pos` is `1` or a **valid index** in the linked-list.

Follow up: Can you solve it using `O(1)` (i.e. constant) memory?

Solution

```
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, x):
#         self.val = x
#         self.next = None
```

```
class Solution:
    def hasCycle(self, head: Optional[ListNode]) -> bool:
        # 初始化快慢双指针
        slow, fast = head, head

        # 当快指针走到末尾的时候停止
        while fast and fast.next:
            # update 快慢指针
            slow = slow.next
            fast = fast.next.next
            # 如果相遇，则返回True
            if slow == fast:
                return True

        # 已经推出了循环，则返回False，没有环
        return False
```

Ref:

- https://labuladong.github.io/algo/2/19/18/
- https://github.com/neetcode-gh/leetcode/blob/main/python/141-Linked-List-Cycle.py