142. Linked List Cycle II

Solved

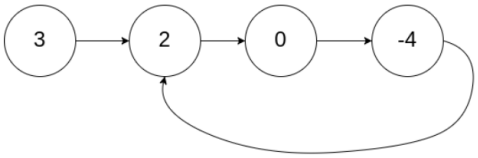
Medium | Topics | Companies

Given the `head` of a linked list, return *the node where the cycle begins. If there is no cycle, return* `null`.

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 (**0-indexed**). It is `-1` if there is no cycle. **Note that** `pos` **is not passed as a parameter**.

Do not modify the linked list.

Example 1:




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

C | Auto

```c
struct ListNode *detectCycle(struct ListNode *head) {
    if (head == NULL || head->next == NULL)
        return NULL;

    struct ListNode *slow = head;
    struct ListNode *fast = head;

    while (fast != NULL && fast->next != NULL) {
        slow = slow->next;
        fast = fast->next->next;

        if (slow == fast) {
            break;
        }
    }
    if (fast == NULL || fast->next == NULL)
        return NULL;

    slow = head;

    while (slow != fast) {
        slow = slow->next;
        fast = fast->next;
    }
    return slow;
}

```

Saved