

查阅更多的题解, 请点击

Problem

141. Linked List Cycle(Easy)

Given a linked list, determine if it has a cycle in it.

To represent a cycle in the given linked list, we use an integer pos which represents the position (0-indexed) in the linked list where tail connects to. If pos is -1, then there is no cycle in the linked list.

Example 1:

Input: head = [3,2,0,-4], pos = 1

Output: true

Explanation: There is a cycle in the linked list, where tail connects to the second node.

Example 2:

Input: head = [1,2], pos = 0

Output: true

Explanation: There is a cycle in the linked list, where tail connects to the first node.

Example 3:

Input: head = [1], pos = -1

Output: false

Explanation: There is no cycle in the linked list.

Follow up:

Can you solve it using $O(1)$ (i.e. constant) memory?

Solution

Corner Cases:

- 链表为空

$O(n)$ time, $O(n)$ space

最直观的想法, 若存在cycle, 那么存在一个node->next指向之前的节点, 将遍历过的节点存入一个hashset, 遍历到新的节点时, 判断该节点是否已经在hashset中。若在, 则存在cycle

```

class Solution
{
public:
    bool hasCycle(ListNode *head)
    {
        if (head == nullptr)
            return false;
        unordered_set<ListNode *> set;
        auto node = head;
        while (node != nullptr)
        {
            if (set.find(node) != set.end())
                return false;
            set.insert(node);
            node = node->next;
        }
        return true;
    }
};

```

O(n) time, O(1) space

第二个方法就是用快慢指针，快指针一次走两步，慢指针一次走一步，若在遍历过程中两指针相等，则存在cycle。

上述方法看上去很巧妙，不过在有cycle的情况下，两指针一定会相遇吗？

类似于两个不同的运动员在环形跑道上跑步，假设慢指针进入环时，快指针落后慢指针n步：

- 两指针前进一次->快指针落后慢指针n-1步
- 两指针再前进一次->快指针落后慢指针n-2步
-

设链表共n个节点，非环部分k，环长x，最多x次之后，**两者一定相遇。**

同样地可以做时间复杂度分析：

- 链表无环，快指针先到达尾部，时间取决于链表的长度，O(n) time
- 链表有环，非环部分为k，在最坏的情况下，环内经过x次可以相遇，所以是O(k+x)=O(n) time

```
class Solution
{
public:
    bool hasCycle(ListNode *head)
    {
        if (head == nullptr)
            return false;
        auto slow = head;
        auto fast = head;
        while (fast != nullptr && fast->next != nullptr)
        {
            slow = slow->next;
            fast = fast->next->next;
            if (slow == fast)
                return true;
        }
        return false;
    }
};
```

[GitHub传送门](#)