141. Linked List Cycle

Tags	
Property	@September 14, 2022

Question

原文：

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`.

我的理解：

大意跟142很像，只是這題只要看有無loop

翻譯：

给出head，即一个链表的头，确定该链表是否有一个循环。

如果列表中存在一些节点，可以通过连续跟踪下一个指针再次到达，那么链接列表中就有一个循环。在内部，pos用来表示tail的下一个指针所连接的节点的索引。注意，pos不作为一个参数传递。

如果在链表中有一个循环，返回true。否则，返回false。

自評翻譯正確性：100%

- Word Memory：

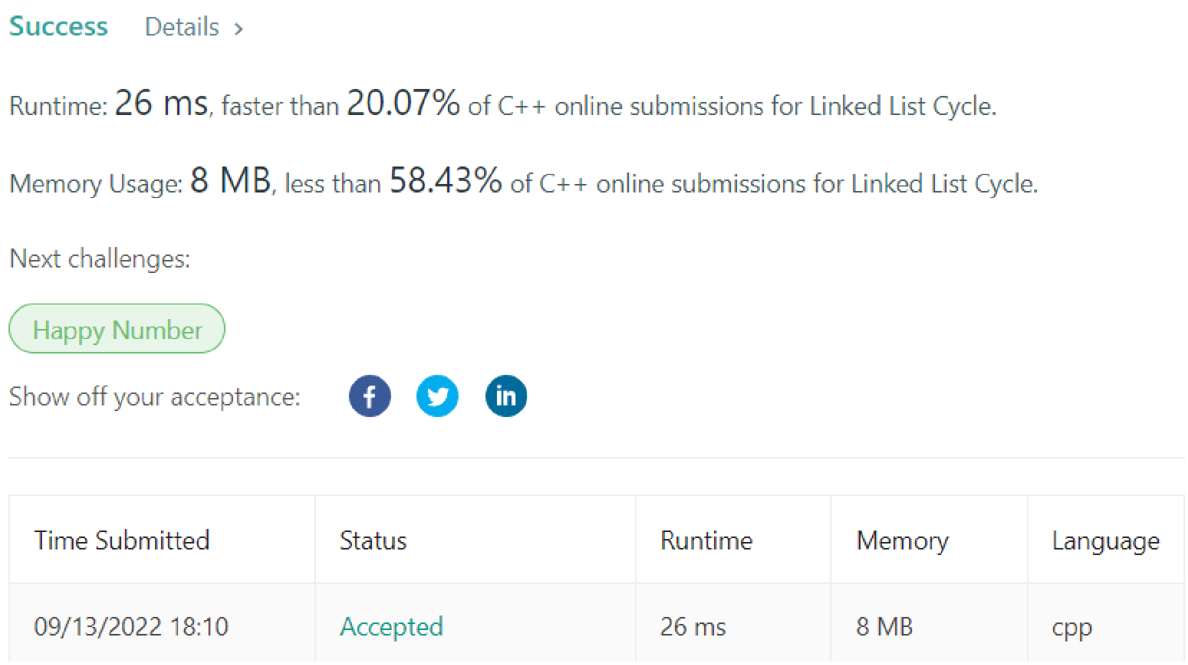
Code

```
class Solution {
public:
    bool hasCycle(ListNode *head) {
        ListNode * slow;
        ListNode * fast;

        slow=head;
        fast=head;

        while(fast!=NULL&&fast->next!=NULL){
            slow=slow->next;
            fast=fast->next->next;
            if(slow==fast){
                return true;
            }
        }
        return false;
    }
};
```

> 思路：就是分別用兩個指標slow fast 如果fast可以追上slow代表有loop



Runtime: 26 ms, faster than 20.07% of C++ online submissions for Linked List Cycle.

Memory Usage: 8 MB, less than 58.43% of C++ online submissions for Linked List Cycle.

Next challenges:

Time Submitted	Status	Runtime	Memory	Language
09/13/2022 18:10	Accepted	26 ms	8 MB	cpp

優良code參考

> 思路：