142. Linked List Cycle II

Add to List ▾

Question | Editorial Solution | My Submissions (/problems/linked-list-cycle-ii/submissions/)

Total Accepted: **100489** Total Submissions: **322841** Difficulty: **Medium** Contributors: **Admin**

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

Note: Do not modify the linked list.

Follow up:

Can you solve it without using extra space?

Subscribe (/subscribe/) to see which companies asked this question

Hide Tags | Linked List (/tag/linked-list/) | Two Pointers (/tag/two-pointers/)

Hide Similar Problems | (E) Linked List Cycle (/problems/linked-list-cycle/) | (H) Find the Duplicate Number (/problems/find-the-duplicate-number/)

Have you met this question in a real interview? Yes No

Discuss (https://discuss.leetcode.com/category/150) | Top Solutions | Pick One (/problems/random-one-question/)

Notes

C++

```cpp
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode *detectCycle(ListNode *head) {
        ListNode* slow=head;
        ListNode* fast=head;
        
        while(fast!=NULL){
            if(fast->next==NULL)
                return NULL;
            fast = fast->next->next;
            slow = slow->next;
            if(fast==slow) break;
        }
        
        if(fast==NULL)
            return NULL;
        slow = head;
        while(slow!=fast){
            slow = slow->next;
            fast = fast->next;
        }
        return slow;
    }
};
```

Submit Solution

Frequently Asked Questions (/faq/) | Terms of Service (/tos/)

Privacy

✉ Send Feedback (mailto:admin@leetcode.com?subject=Feedback)