

LeetCode

ExploreProblemsInterview<sup>New</sup>ContestDiscussStore

October LeetCode Challenge 2021

Premium

DescriptionSolutionDiscuss (999+)Submissions

iJavaAutocomplete

i{}↺⌛⌕

141. Linked List Cycle

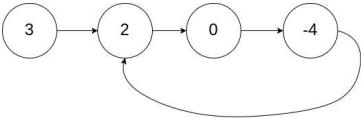
Easy5853672Add to ListShare

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:

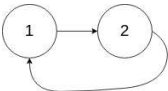


Input: head = [3,2,0,-4], pos = 1

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  $[0, 10^4]$  .
- $-10^5 \leq \text{Node.val} \leq 10^5$
- `pos` is `-1` or a **valid index** in the linked-list.

Follow up:

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

Accepted 1,107,268

Submissions 2,500,918

Seen this question in a real interview before?

YesNo

Companies

Related Topics

Similar Questions

```
1  /**
2  * Definition for singly-linked list.
3  * class ListNode {
4  *     int val;
5  *     ListNode next;
6  *     ListNode(int x) {
7  *         val = x;
8  *         next = null;
9  *     }
10 * }
11 */
12 public class Solution {
13     public boolean hasCycle(ListNode head) {
14
15     }
16 }
```

Problems

Pick One

Prev141/2051Next

ConsoleContribute

Run Code

Submit

https://leetcode.com/problems/linked-list-cycle/

1/1