Leetcode Problem 1. (Easy)

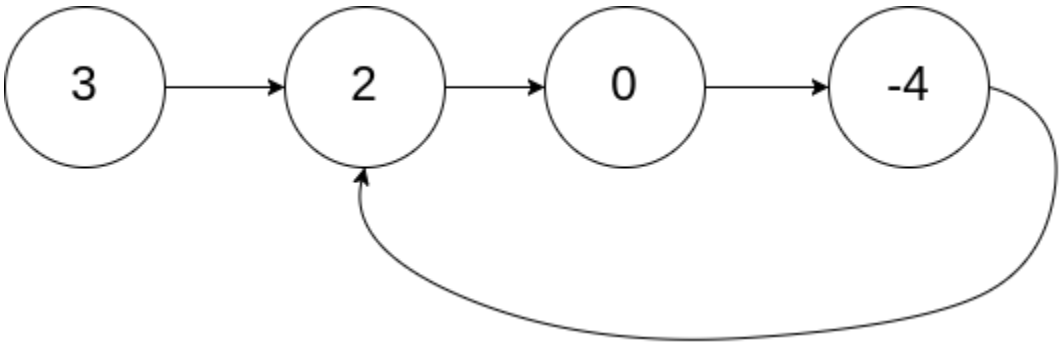
Linked List Cycle

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**.

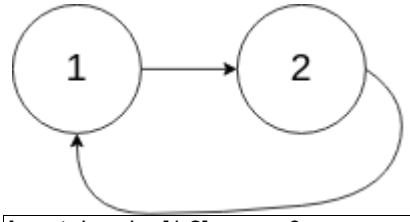
Return true *if there is a cycle in the linked list*. Otherwise, return false.

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.

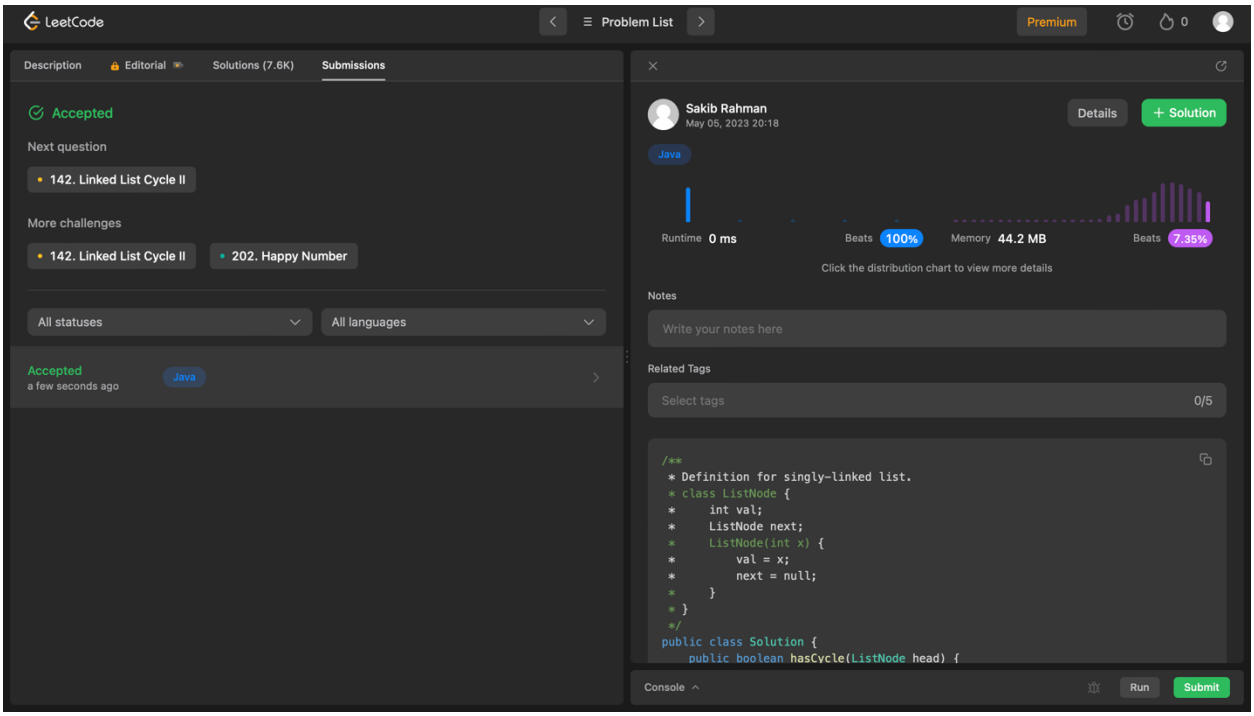
Constraints:

- The number of the nodes in the list is in the range [0, 10^4].
- -10^5 <= Node.val <= 10^5
- pos is -1 or a **valid index** in the linked-list.

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

```
public class Solution {
    public boolean hasCycle(ListNode head) {

    if (head == null || head.next == null) {
        return false;
    }
    ListNode slow = head;
    ListNode fast = head.next;
    while (slow != fast) {
        if (fast == null || fast.next == null) {
            return false;
        }
        slow = slow.next;
        fast = fast.next.next;
    }
    return true;
}
}
```


LeetCode
Problem List
Premium
0
Description
Editorial
Solutions (7.6K)
Submissions
Accepted
Next question
142. Linked List Cycle II
More challenges
142. Linked List Cycle II
202. Happy Number
All statuses
All languages
Accepted
a few seconds ago
Java
Sakib Rahman
May 05, 2023 20:18
Details
+ Solution
Java
Runtime 0 ms
Beats 100%
Memory 44.2 MB
Beats 7.35%
Click the distribution chart to view more details
Notes
Write your notes here
Related Tags
Select tags
0/5
/**
* Definition for singly-linked list.
* class ListNode {
*   int val;
*   ListNode next;
*   ListNode(int x) {
*   val = x;
*   next = null;
*   }
* }
*/
public class Solution {
public boolean hasCycle(ListNode head) {
Console
Run
Submit