**142. Linked List Cycle II**

Medium

3598272Add to ListShare

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

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**.

**Notice** that you **should not modify** the linked list.

**Example 1:**

图示

描述已自动生成

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

**Output:** tail connects to node index 1

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

**Example 2:**

图片包含 游戏机, 物体, 钟表, 画

描述已自动生成

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

**Output:** tail connects to node index 0

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

**Example 3:**

图片包含 图示

描述已自动生成

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

**Output:** no cycle

**Explanation:** There is no cycle in the linked list.

**Constraints:**

* The number of the nodes in the list is in the range [0, 104].
* -105 <= Node.val <= 105
* pos is -1 or a **valid index** in the linked-list.