class Solution {

public:

ListNode \*detectCycle(ListNode \*head) {

if (head == nullptr || head->next == nullptr) {

return nullptr; // No cycle possible

}

ListNode \*slow = head;

ListNode \*fast = head;

// 1. Detect Cycle

while (fast != nullptr && fast->next != nullptr) {

slow = slow->next;

fast = fast->next->next;

if (slow == fast) { // Cycle detected!

break;

}

}

// 2. Find Cycle Start

if (slow == fast) { // Only if cycle detected

slow = head; // Reset slow to head

while (slow != fast) {

slow = slow->next;

fast = fast->next;

}

return slow; // Intersection point is the cycle start

}

return nullptr; // No cycle found

}

};