Given a linked list, determine if it has a cycle in it.

Follow up:  
Can you solve it without using extra space?

/\*\*

\* 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) {

if(head==null||head.next==null) return false;

ListNode once = head;

ListNode twice= head.next;

while(twice!=null) {

if( once==twice) return true;

once=once.next;

if(twice.next==null||twice.next.next==null)

return false;

twice= twice.next.next;

}

return false;

}

}