# Approach 1 -> Tortoise and Hare

fast,slow = head,head

# Fast reaches end of list sooner

while fast and fast.next:

# slow moves 1 node at a time

slow = slow.next;

# fast moves 2 nodes at a time

fast = fast.next.next;

# if ll has a cycle, at some point both point at same node

if slow==fast:

return True

return False

"""

APPROACH 0

TC : O(n)

SC : O(n)

# if only one node, no cycle

if not head or not head.next:

return False

# Traverse the linkedlist

visited = set()

temp = head;

while temp:

# if already visited

if temp in visited:

return True

visited.add(temp)

temp = temp.next

return False