Detect entrance node of inner loop

##Recall how to check entrance of inner loop.

Check cycle, if not fast then no cycle.

##Fast>>>end and head>>>end, entrance node as intersection of those two.

# Definition for singly-linked list.

# class ListNode(object):

# def \_\_init\_\_(self, x):

# self.val = x

# self.next = None

class Solution(object):

def entryNodeOfLoop(self, head):

"""

:type head: ListNode

:rtype: ListNode

"""

slow=fast=head

while fast and fast.next:

fast=fast.next.next

slow=slow.next

if slow==fast:

break

if not fast:

return

while head!=fast:

head=head.next

fast=fast.next

return head