You have successfully solved Cycle Detection Share Tweet

Try the Next Challenge | Try a Random Challenge

Find Merge Point of Two Lists

by harsha_s

Problem Submissions Leaderboard Discussions Editorial

This challenge is part of a tutorial track by MyCodeSchool

Given pointers to the head nodes of 2 linked lists that merge together at some point, find the Node where the two lists merge. It is guaranteed that the two head Nodes will be different, and neither will be NULL.

In the diagram below, the two lists converge at Node `x`:

```
[List #1] a--->b--->c
                    \
                     x--->y--->z--->NULL
                    /
     [List #2] p--->q
```

Complete the `int FindMergeNode(Node* headA, Node* headB)` method so that it finds and returns the data value of the Node where the two lists merge.

Input Format

The `FindMergeNode(Node*,Node*)` method has two parameters, $headA$ and $headB$, which are the non-null head Nodes of two separate linked lists that are guaranteed to converge.
Do not read any input from stdin/console.

Output Format

Each Node has a data field containing an integer; return the integer data for the Node where the two lists converge.
Do not write any output to stdout/console.

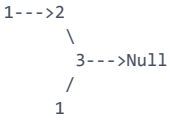
Sample Input

The diagrams below are graphical representations of the lists that input Nodes $headA$ and $headB$ are connected to. Recall that this is a method-only challenge; the method only has initial visibility to those 2 Nodes and must explore the rest of the Nodes using some algorithm of your own design.

Test Case 0

```
1
 \
  2--->3--->NULL
 /
1
```

Test Case 1



Sample Output

```
2
3
```

Explanation

Test Case 0: As demonstrated in the diagram above, the merge Node's data field contains the integer **2** (so our method should return **2**).
Test Case 1: As demonstrated in the diagram above, the merge Node's data field contains the integer **3** (so our method should return **3**).

Easy Submitted 40211 times
Max Score **5**

Need Help?

View Discussions

View Editorial Solution

View Top Submissions

Rate This Challenge:

Download problem statement

Download sample test cases

Suggest Edits

Current Buffer (saved locally, editable) Python 3

```
"""
 Find the node at which both lists merge and return the data of that node.
 head could be None as well for empty list
 Node is defined as

 class Node(object):

   def __init__(self, data=None, next_node=None):
       self.data = data
       self.next = next_node


"""

def FindMergeNode(headA, headB):
    temp1=headA
    temp2=headB
    size1=0
    size2=0
```

```
    while(temp1):
        size1+=1
        temp1=temp1.next
    while(temp2):
        size2+=1
        temp2=temp2.next
    if size1>size2:
        left=size1-size2
        while(left!=0):
            headA=headA.next
            left-=1
    if size2>size1:
        left=size2-size1
        while(left!=0):
            headB=headB.next
            left-=1
    while(headA and headB):
        if headA.data==headB.data:
            return headA.data
        else:
            headA=headA.next
            headB=headB.next
```

Line: 41 Col: 29

Upload Code as File | Test against custom input | Run Code | Submit Code

Testcase 0 ✔

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

```
1
```

Your Output (stdout)

```
3
3
7
```

Expected Output

```
3
3
7
```