



[Daily Problem] Intersection of Linked Lists

1 missatge

Daily Interview Pro <daily@techseries.dev>

29 d'agost de 2020 a les 16:34



Hi, here's your problem today. This problem was recently asked by Apple:

You are given two singly linked lists. The lists intersect at some node. Find, and return the node.
Note: the lists are non-cyclical.

Example:

A = 1 -> 2 -> 3 -> 4

B = 6 -> 3 -> 4

This should return 3 (you may assume that any nodes with the same value are the same node).

Here is a starting point:

```
def intersection (a, b):  
    # fill this in.  
  
class Node(object):  
    def __init__(self, val):  
        self.val = val  
        self.next = None  
    def prettyPrint (self):  
        c = self  
        while c:  
            print c.val,  
            c = c.next  
  
a = Node(1)  
a.next = Node(2)  
a.next.next = Node(3)  
a.next.next.next = Node(4)  
  
b = Node(6)  
b.next = a.next.next  
  
c = intersection (a, b)  
c.prettyPrint ()  
# 3 4
```

Upgrade to PRO for in-depth solutions with time-space complexity analysis.

» Ready to fast-track your career? Join the premiere tech interview training program [Tech Interview Pro](#).

» For more, join me at CoderPro for 100+ whiteboard coding sessions: <https://coderpro.com/> (20% off now, limited time)

Have a great day!
Daily Interview Pro

© 2020 Daily Interview Pro. All rights reserved.
Did this email bother you? [Unsubscribe](#) anytime.