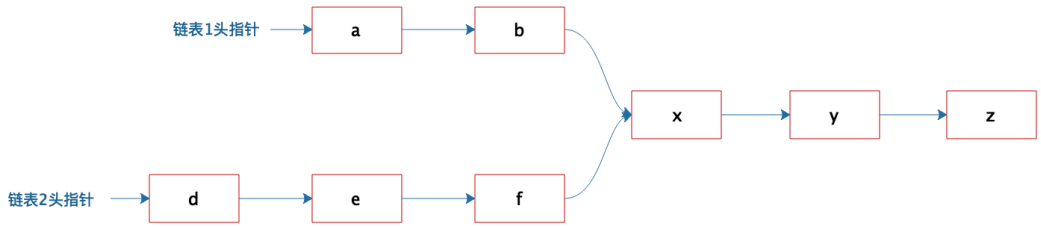
（1）有两个单向链表（链表长度分别为 m，n），这两个单向链表有可能在某个元素合并，如下图所示的这样，也可能不合并。现在给定两个链表的头指针，在不修改链表的情况下，如何快速地判断这两个链表是否合并？如果合并，找到合并的元素，也就是图中的 x 元素。请用（伪）代码描述算法，并给出时间复杂度和空间复杂度。



分析： 假设链表 A 中 ，非相交部分为 A1， 相交部分为 C；

链表 B 中，非相交部分为 B1，相交部分为 C

使用两个指针(pA 和 pB)分别遍历链表 A+B 和链表 B+A，总长度相等。

即（A1+C) +（B1 +C ）= (B1+C)+（A1+C） ➔ (A1+C) +B1 = (B1+C) + A1

遍历一定会同时结束于相交点。**当 pA=pB 指针相等时，即为交叉点。**如果此时指针为 null，则表示 2 个链表无交叉元素。

时间复杂度： O(m+n)

空间复杂度： O(1)

代码参看 GetInterSectNode.py 文件

（2）请画出 DataNode 服务器节点宕机的时候，HDFS 的处理过程时序图。

HDFS 将每个文件数据分成若干数据块(block)，每个 block 使用多个副本（默认是 3）进行存储（可以存储在不同的服务器，甚至不同的机架上）。

当 NameNode 超时未收到 dataNode 的心跳，则认为此 datanode 已宕机。

NameNode 查找到此 datanode 的所有数据块 block，并通知拥有其他副本的 datanode 重新备份数据。