CMSC 132: Object-Oriented Programming II

Linked List Examples
Example 1

Use only one iteration, find n’th node from the end of a Linked List

head

n=4
Example 1

Use only one iteration, find n’th node from the end of a Linked List

- Two pointers p1=head; p2= head
- First move p1 to n nodes from head.
- Now move both pointers one by one until p1 reaches end.
- P2 stops at target
Example 2

Find the intersection point of two Linked Lists
Example 2

Find the intersection point of two Linked Lists

\[ n_1 = \text{length}(l1) \]
\[ n_2 = \text{length}(l2) \]
\[ d = l1 - l2 \]
Move \( p1 \) \( d \) times
Move both \( p1 \) and \( p2 \) until they meet
Example 3:

Find the split point of two Linked Lists
Example 4:

Detect if there is a cycle in a Singly Linked List
boolean hasLoop(Node first) {
    if(first == null)  return false;
    Node slow, fast;  // create two references.
    slow = fast = first;
    while(true){
        slow = slow.next;  // 1 hop.
        if(fast.next != null)
            fast = fast.next.next;  // 2 hops.
        else
            return false;  // next node null => no loop.
        if(slow == null || fast == null)
            // if either hits null, no loop.
            return false;
        if(slow == fast)  // if the two ever meet...we must have a loop.
            return true;
    }
}
Example 5:

Remove the cycle in a Singly Linked List
Example 6:

Clone a linked list with next and random pointer

1 → 2 → 3 → 4 → 5