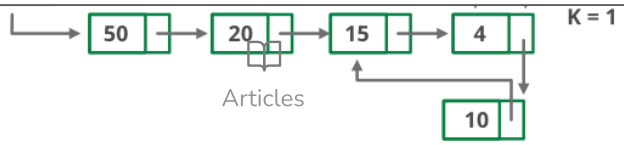
Detect and remove loop in linked list

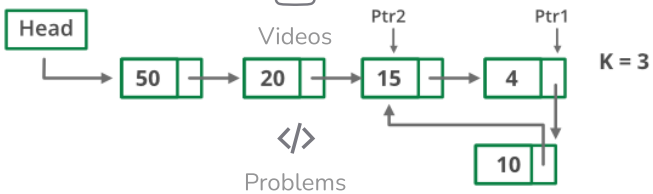
Method using Floyd's Cycle detection algorithm :

1. This method is also dependent on Floyd's Cycle detection algorithm.
2. Detect Loop using Floyd's Cycle detection algorithm and get the pointer to a loop node.
3. Count the number of nodes in the loop. Let the count be k.
4. Fix one pointer to the head and another to a kth node from the head.
5. Move both pointers at the same pace, they will meet at the loop starting node.
6. Get a pointer to the last node of the loop and make the next of it NULL.

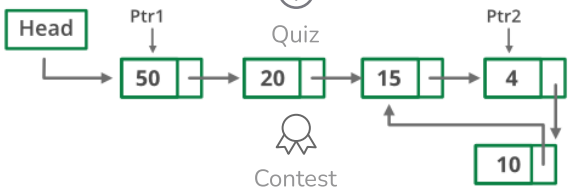
Below image is a dry run of the 'remove loop' function in the code :



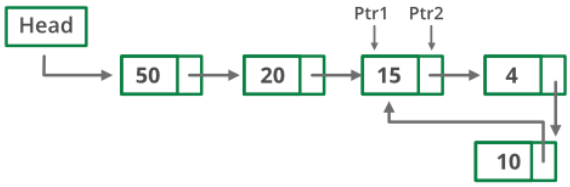
Step 1: Find number of nodes involved in loop



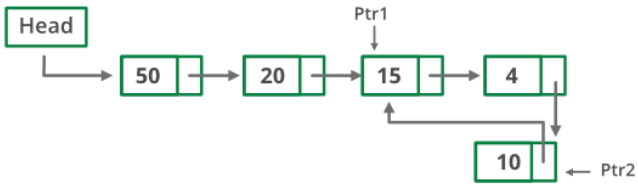
Step 2: Fix one pointer to head and other pointer to K nodes after head



Step 3: Find starting node of loop



Step 4: Find last node of the loop



Step 5: Remove loop

Below is the implementation of the above approach:

```
static Node head;

static class Node {

    int data;
    Node next;

    Node(int d)
    {
        data = d;
        next = null;
    }
}

// Function that detects loop in the list
int detectAndRemoveLoop(Node node)
{
    Node slow = node, fast = node;
    while (slow != null && fast != null
        && fast.next != null) {
        slow = slow.next;
        fast = fast.next.next;

        // If slow and fast meet at same point then loop
        // is present
        if (slow == fast) {
            removeLoop(slow, node);
            return 1;
        }
    }
    return 0;
}

// Function to remove loop
void removeLoop(Node loop, Node head)
{
```

```
        while (ptr1.next != ptr2) {
            // keeping track beforeing moving next
            prevNode = ptr1;
            ptr1 = ptr1.next;
            k++;
        }
        prevNode.next = null;
    }

    // Function to print the linked list
    void printList(Node node)
    {
        while (node != null) {
            System.out.print(node.data + " ");
            node = node.next;
        }
    }

    // Driver program to test above functions
    public static void main(String[] args)
    {
        LinkedList list = new LinkedList();
        list.head = new Node(50);
        list.head.next = new Node(20);
        list.head.next.next = new Node(15);
        list.head.next.next.next = new Node(4);
        list.head.next.next.next.next = new Node(10);

        // Creating a loop for testing
        head.next.next.next.next.next = head.next.next;
        list.detectAndRemoveLoop(head);
        System.out.println(
            "Linked List after removing loop : ");
        list.printList(head);
    }
}
```

Dash

Output

All

Articles

Videos

Problems

Mark as Read

Quiz

Report An Issue

If you are facing any issue on this page. Please let us know.

Contest

Menu

Track Progress

85 of **132** Complete. (65%)