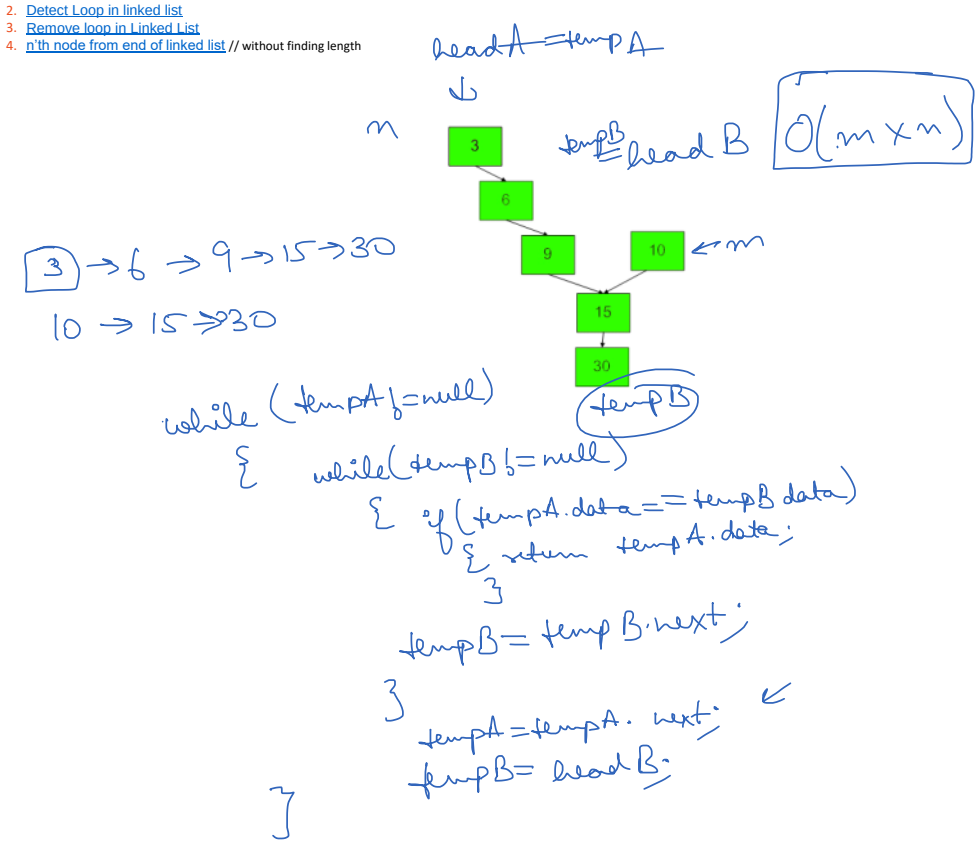
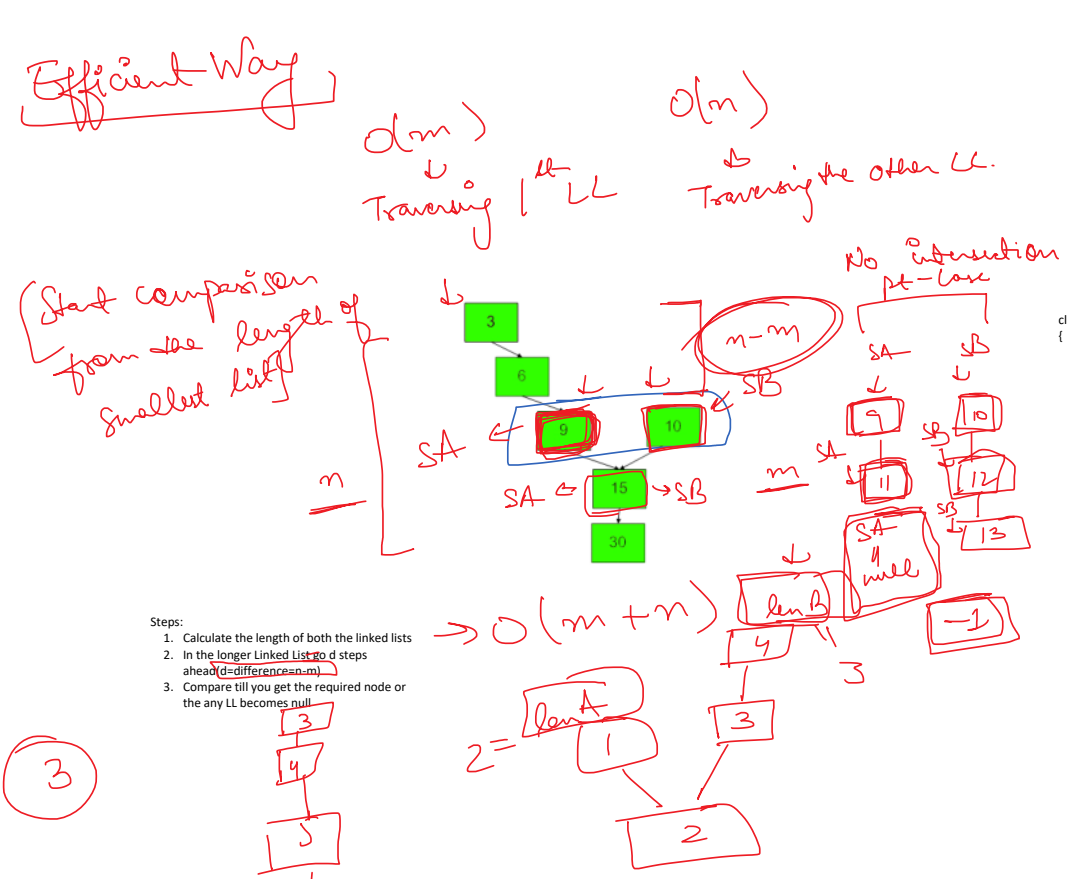
June - 01 LL

01 June 2020 21:12

1. Intersection point in Y shaped linked lists
2. Detect Loop in linked list
3. Remove loop in Linked List
4. n'th node from end of linked list // without finding length





Steps:

1. Calculate the length of both the linked lists
2. In the longer Linked List go d steps ahead(d=difference=n-m)
3. Compare till you get the required node or the any LL becomes null

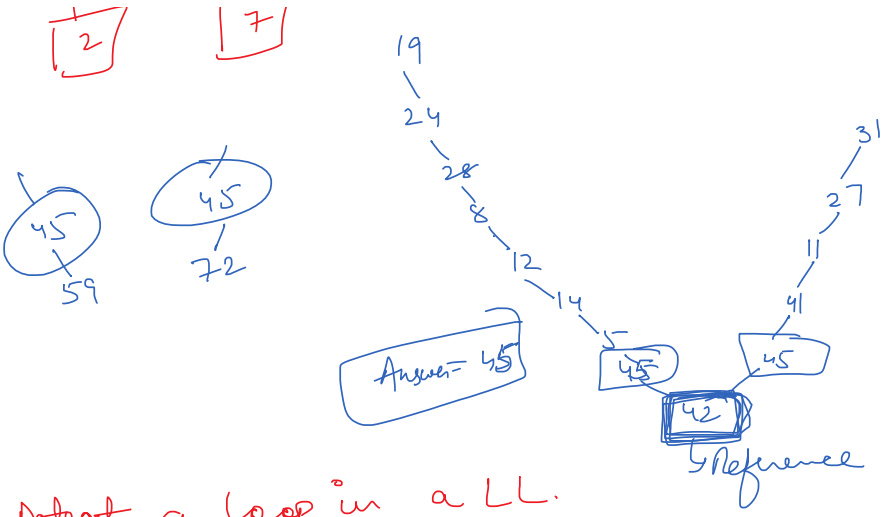
```
class Intersect
{
    int intersectPoint(Node headA, Node headB)
    {
    // code here
    int lenA=0;
    int lenB=0;
    Node temp = headA;
      while(temp!=null)
      {
        temp=temp.next;
        lenA++;
      }
      temp = headB;
      while(temp!=null)
      {
        temp=temp.next;
        lenB++;
      }
      int diff = Math.abs(lenA-lenB);
      //call the longer LL as A
      Node tempA;
      Node tempB;
      if(lenA>=lenB)
      {
        tempA=headA;
        tempB=headB;
      }
      else{
        tempA=headB;
        tempB=headA;
      }
      //move the ptr to longer ll diff steps ahead
      for(int i=0;i<diff;i++)
      {
        tempA=tempA.next;
      }
      //reached the required state
      while(tempA!=null && tempB!=null)// if even any of them becomes null then stop
      {
        if(tempA==tempB)//reference of intersection point
        {
          return tempA.data;
        }
        tempA=tempA.next;
        tempB=tempB.next;
      }
      return -1;

    }
```

19 24 28 8 12 14 5 45
31 27 11 41 45
42

From <https://practice.geeksforgeeks.org/problems/intersection-point-in-y-shapped-linked-lists/1/>

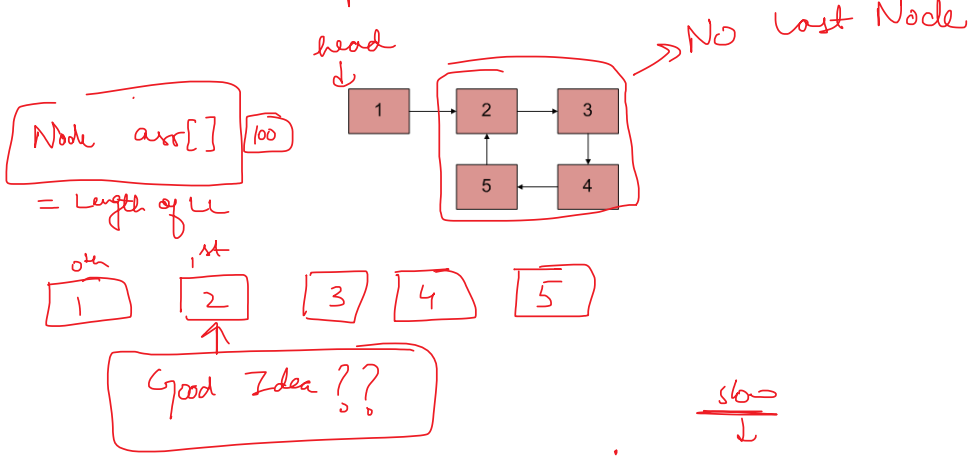
19

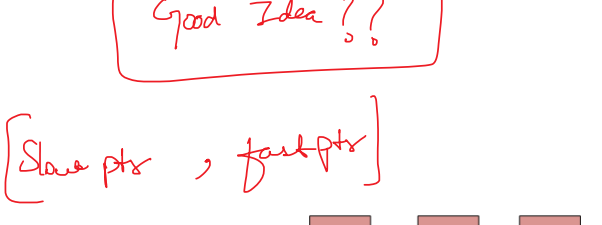


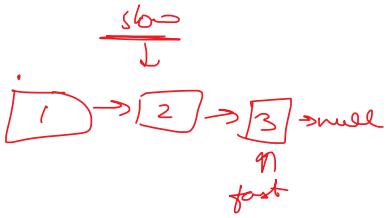

```
            }
            tempA=tempA.next;
            tempB=tempB.next;
          }
          return -1;

        }
    }
```

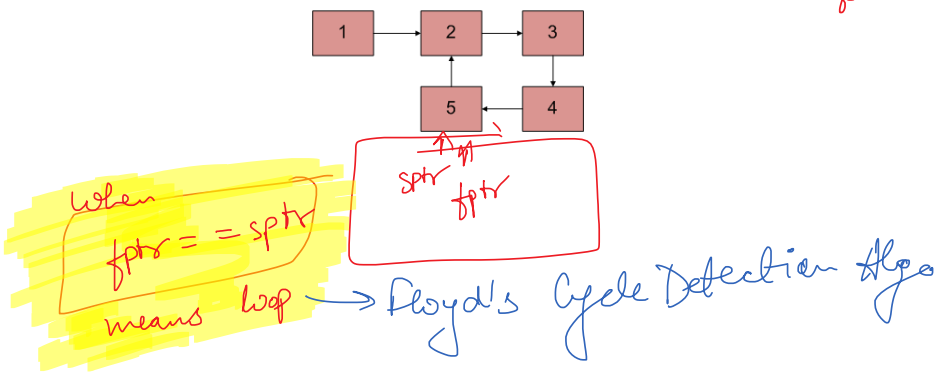
2 Detect a loop in a LL.





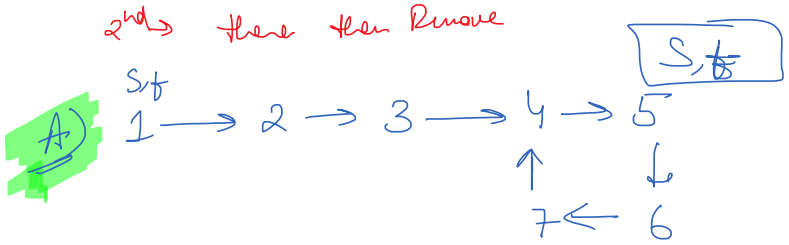


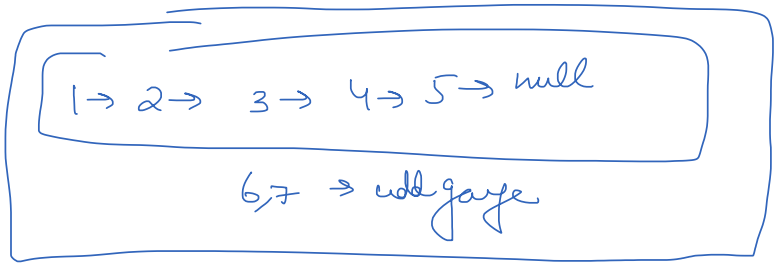

```
class Solution {
  public int detectLoop(Node head) {
    // Add code here
    Node slow = head;
    Node fast = head;
    while(fast!=null && fast.next!=null)
    {
      slow=slow.next;
      fast=fast.next.next;
      if(slow == fast)
      {
        return 1;
      }
    }
    return 0;
  }
}
```

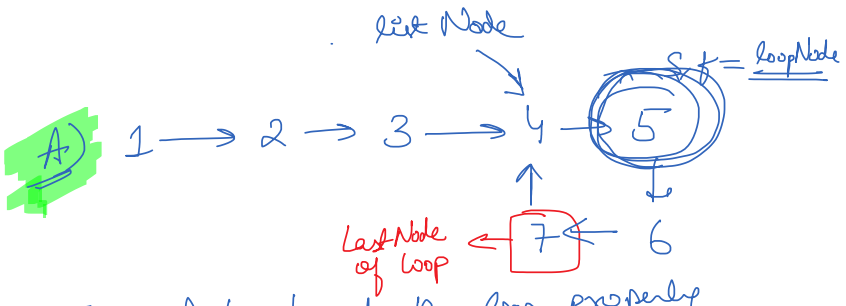


3 Remove loop in a LL

Steps →
1st → Detect a loop
2nd → then then Remove







If I want to break the loop properly which link do I need to break?

7 & 4

7.next = null

Last Node ka Next lies on the list.

Algo →

```
void removeLoop(Node listNode, Node loopNode)
{  Node temp = loopNode;
   while (true)
   {  while (loopNode != temp && loopNode.next != listNode)
      {  loopNode = loopNode.next;
      }
      if (loopNode.next == listNode)
      {  loopNode.next = null;
         break;
      }
      listNode = listNode.next;
   }
```