7th feb 2023

FIRST:-

Intersection Point in Y Shaped Linked Lists : : Medium

Given two singly linked lists of size **N** and **M,** write a program to get the point where two linked lists intersect each other.

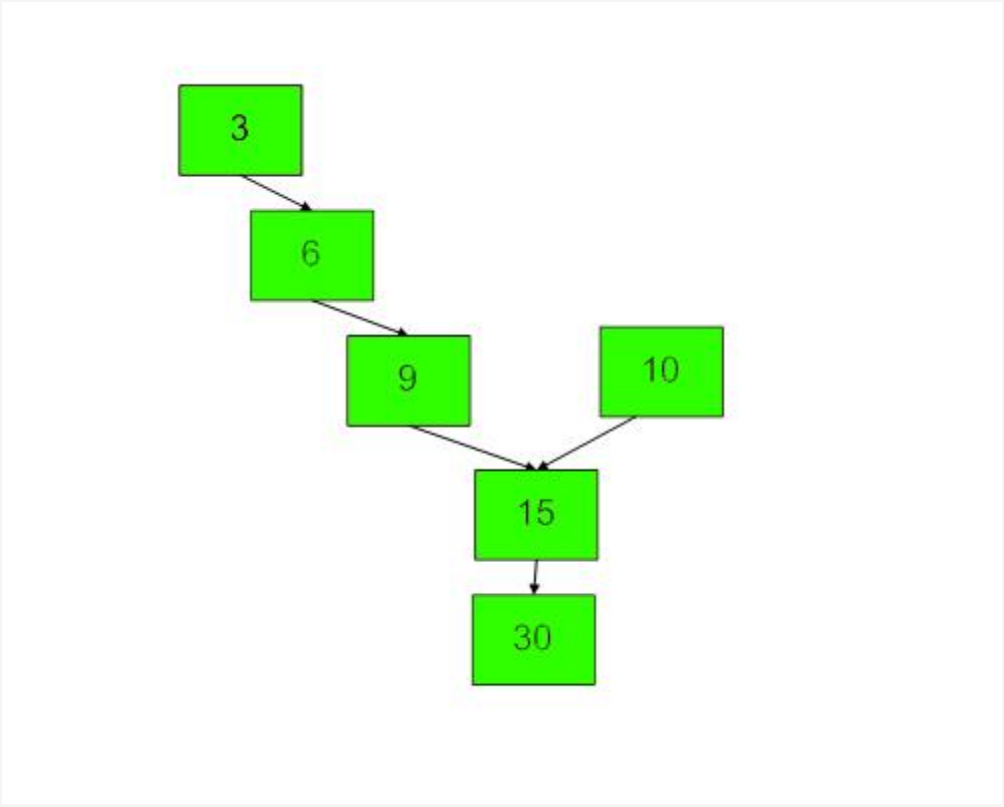
Example 1:

Input:

```
LinkList1 = 3->6->9->common
LinkList2 = 10->common
common = 15->30->NULL
```

Output: 15

Explanation:



Example 2:

```
Input:
Linked List 1 = 4->1->common
Linked List 2 = 5->6->1->common
common = 8->4->5->NULL
Output: 8
Explanation:

4               5
|               |
1               6
 \              /
  8   -----  1
  |
  4
  |
  5
  |
  NULL
```

Your Task:
You don't need to read input or print anything. The task is to complete the function **intersetPoint**() which takes the pointer to the head of linklist1(**head1**) and linklist2(**head2**) as input parameters and returns data value of a node where two linked lists intersect. If linked list do not merge at any point, then it should return **-1**.
Challenge : Try to solve the problem without using any extra space.

Expected Time Complexity: O(N+M)
Expected Auxiliary Space: O(1)

Constraints:

$1 \leq N + M \leq 2*10^5$

$-1000 \leq value \leq 1000$

CODE SECTION:-

```
int intersectPoint(Node* head1, Node* head2)
   {
       // Your Code Here
       // brute force method

       // Node *first=head1;
       // Node *second=head2;
       // Node *q=head1;

       // if(first!=NULL || second!=NULL){

       //     while(first && second){

       //         while(first){

       //             if(first==second) return first->data;
       //             first=first->next;

       //         }
       //         first=q;
       //         second=second->next;

       //     }

       //     return -1;

       // }


       // else return -1;

       //optimized approach
       //find the length of both of the ll
        int len1=0;
        int len2=0;
        Node *node1 = head1;
        Node *node2 = head2;

        //to find length
        while(node1!=NULL){
            len1++;
            node1=node1->next;
```

```
        }
        while(node2!=NULL){
            len2++;
            node2=node2->next;
        }
        node1=head1;
        node2=head2;
        int diff=0;

        //traverse the node with the greter length by the difference between
their length
        if(len1>len2){
            diff=len1-len2;
            for(int i=0;i<diff;i++){
                node1=node1->next;
            }
        }
        else{
            diff=len2-len1;
            for(int i=0;i<diff;i++){
                node2=node2->next;
            }
        }

        //traverse from now to the end and check if they are equal then
return the data of that node otherwise return -1
        while(node1!=NULL && node2!=NULL){
            if(node1==node2){
                return node1->data;
            }
            node1=node1->next;
            node2=node2->next;
        }
        return -1;
    }
```

-: DONE FOR THE DAY:-