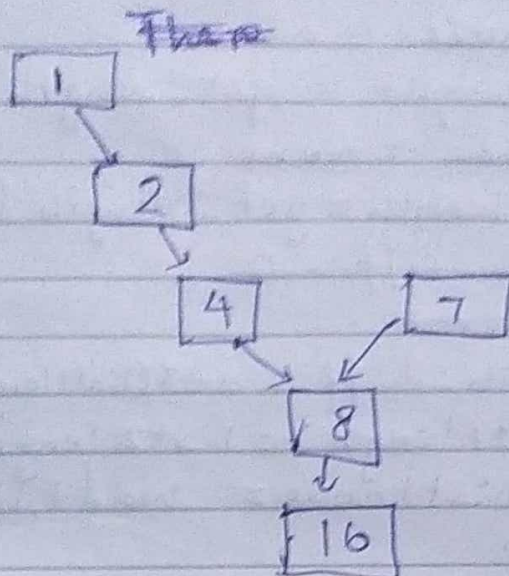


## DAY-38

① Finding intersection point of two linked lists.



above diagram shows an example with two linked list having '8' as intersection point.

Algorithm:-

→ Find lengths of both ~~nodes~~ linked list:  $L_1$  &  $L_2$

→ Calculate difference in length of both linked lists:  
 $d = \text{abs}(L_1 - L_2)$

→ move the head pointer of longer list 'd' steps forward.

→ Now, traverse both lists, comparing nodes until we find a match or reach the end of lists.



Implementation: -

```
def intersect(head1, head2):  
    list1node = None  
    list1length = get_length(head1)  
    list2node = None  
    list2length = get_length(head2)  
    length_diff = 0  
    if list1length > list2length:  
        length_diff = abs(list1length - list2length)  
        if list1length >= list2length:  
            list1node = head1, list2node = head2  
        else:  
            list1node = head2, list2node = head1
```

```
    while length_diff > 0:  
        list1node = list1node.next  
        length_diff -= 1  
    while list1node != None:  
        if list1node == list2node:  
            return list1node  
        list1node = list1node.next  
        list2node = list2node.next  
    return None
```

Time complexity:

Linear,  $O(m+n)$

[ $m = \text{len1}$ ,  $n = \text{len2}$ ]

Memory Complexity:

Constant  $O(1)$