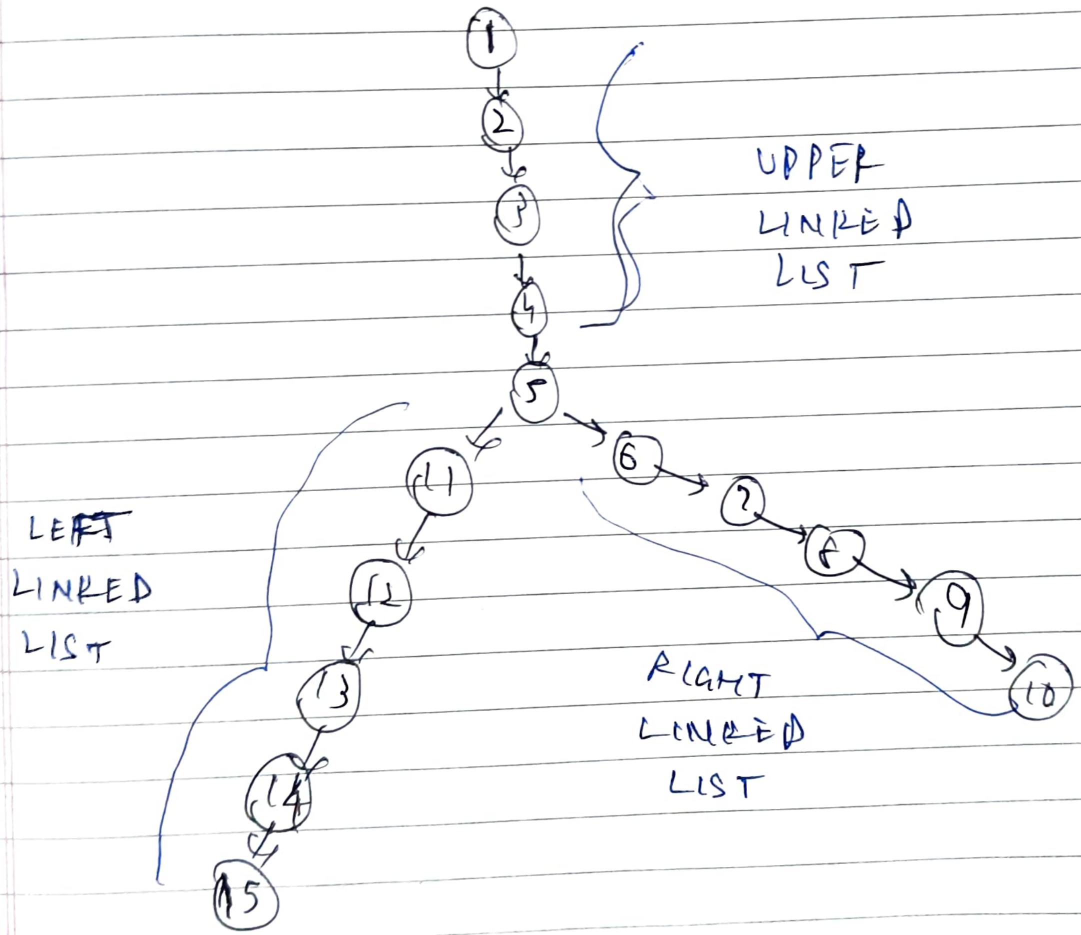
SACHIN PRASANNA 211IT058

Nodes ~~Nodes~~ are indexed in the following way.

Count from top and when the root (middle) node is encountered, count the right part of linked list, and then left part.



★ <u>Insertion</u> :

In an already existing 15 node linked list, in the worst case, insertion will happen in 15^{th} node

SACHIN PRASANNA 211IT058

(as numbered in diagram).

- If we insert a new node on the 15th (left leaf node), then we have traversed all the elements.

- In that case, the time complexity will become O(n), where n is the size of linked list.

Worst case Time Complexity → O(n)

★ Deletion

- Similarly as in insertion, we take the position of the node to be deleted, which is the leaf node in left linked list.

- Again, all the nodes are traversed, which makes the time complexity O(n), where n is the size of the linked list.

Worst Case Time Complexity → O(n)

★ Search

- Similarly as in insertion and deletion, we traverse Upper linked list, then Right Linked List, then finally

SACHIN PRASANNA 21ITO58

Left linked list.

If the element to be searched is the last element of the left linked list, then all element will be traversed in worst case.

So,

Worst case Time Complexity → $O(n)$

x —— x —— x