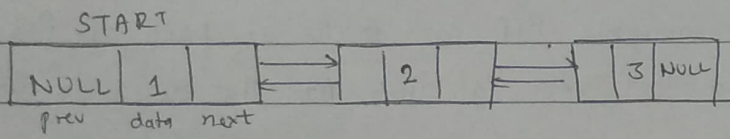
Name: Shreyas Sawant
D7A, 55

SDS | Page No.
Date / /

AIM: Implement doubly linked list using ADT

THEORY:

1. A doubly linked or a 2-way linked list is a more complex type of linked list which contains a pointer to the next as well as the previous node in the sequence. Therefore, it consists of three parts - data, a pointer to previous node and to next node.
2. For example



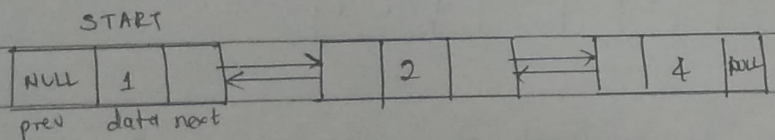
3. The PREV field of first node and the NEXT field of last node will contain NULL.

- Operations on Doubly Linked List.

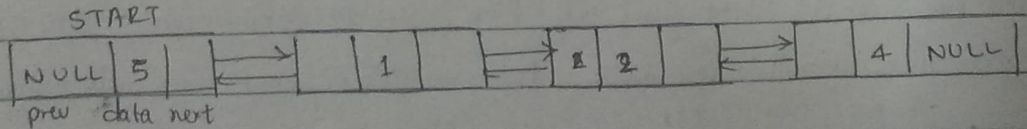
1. Insertion:

a) Inserting a Node at Beginning of Doubly Linked List

Consider the doubly linked list shown below suppose we want to add a new node with data 9 as the first node of the list.

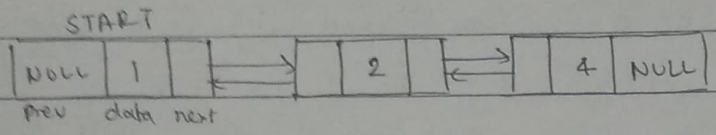


Allocate the new node before the START node. Now START becomes the new node



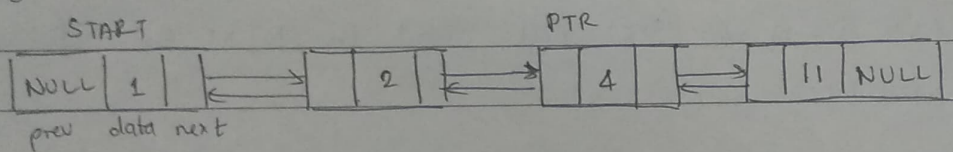
b) Inserting a Node at end of doubly linked list

Consider the doubly linked list shown below. Suppose we want to add a new node with data '11' as last node of list



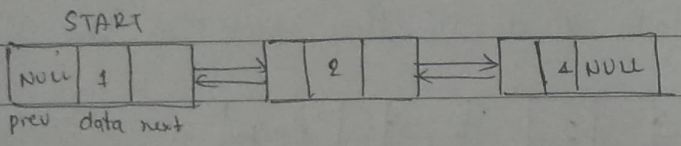
Allocate memory for new node and initialize its data part to 11 and next field to NULL

Take a pointer variable PTR and make it point to first node and then move it to last node of list. Add the new node after the node pointed by PTR

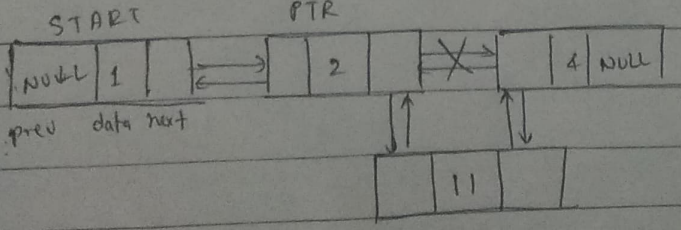


c) Inserting a Node After Given Node in doubly linked list

Consider the list shown below. Suppose we want to add a new node with value '11' after 2.



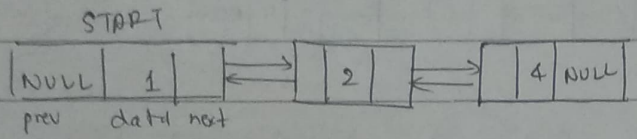
Allocate memory for new node and initialize its data to 11. Take a pointer variable PTR and make it point to first node of list and move PTR further until data part of PTR = value after which the node has to be inserted. Insert new node between PTR and node succeeding to it.



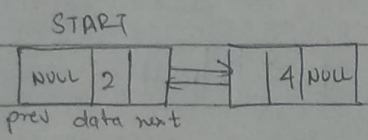
2 Deletion :

a) Deleting the First Node of doubly linked list

Consider the doubly linked list shown below when we want to delete a node from the beginning of list, then following changes will be done

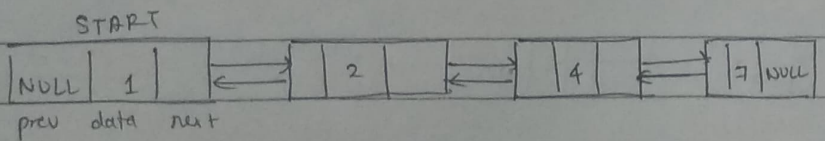


For Free the memory occupied of first node of list and make the next node as st START node.



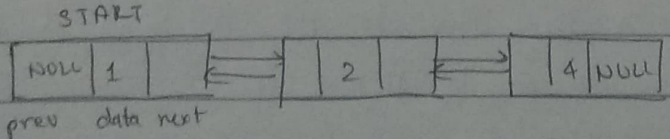
b) Deleting the last Node of Doubly linked list

Consider the list shown below, we want to delete last node from the list. then following changes will be done



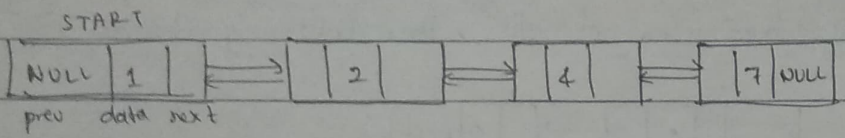
Take a pointer variable PTR that points to the first node of list. Move PTR to end element.

Free space by PTR of PTR node and store NULL in NEXT field of previous node

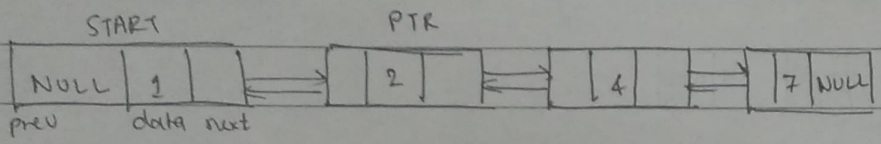


c) Deleting the node after a given node in doubly linked list

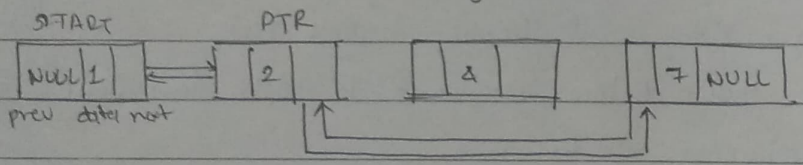
Consider the doubly linked list shown below. Suppose we want to delete node that succeeds the node with element 2.

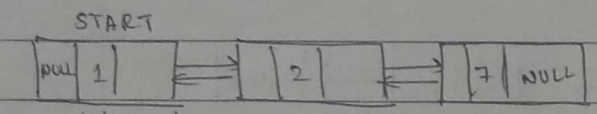


Take a pointer variable PTR and make it point to the first node of list. Move PTR further so that & its data part is equal to the value after which the node has to be ~~inserted~~ deleted.



Delete the node succeeding PTR





- Limitations of Doubly Linked List

1) Compared to singly linked list, each node store an extra pointer which consumes extra memory.
2) Operations require more time due to the overhead of handling extra pointers as compared to singly linked lists.
3) No random access of elements.

SDS | Page No.
Date / /

CONCLUSION:

Errors encountered:

1) Incorrect way of declaring pointer, Missing *, Node prev.

Solution: Node *prev;

2) Incorrect syntax: if (ptr !== NULL)
{ ... }

Solution: Correct syntax: if (ptr != NULL)
{ ... }