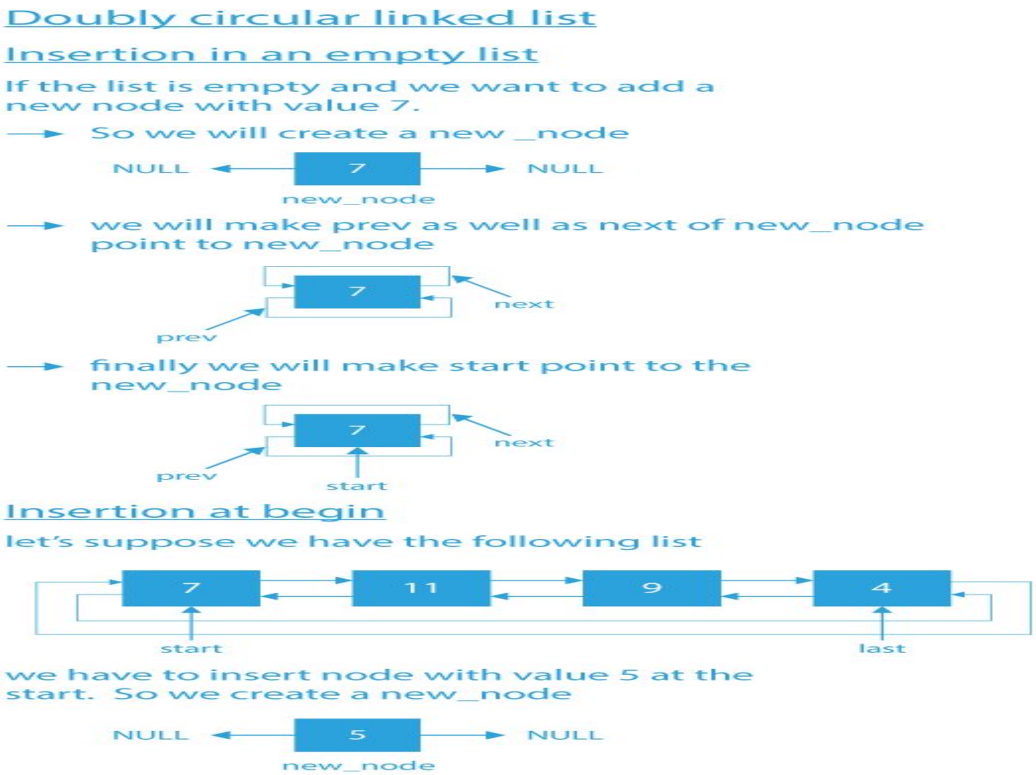
Doubly Circular Linked List

By Er. Kushal Ghimire

Insertion in empty list



Doubly circular linked list

Insertion in an empty list

If the list is empty and we want to add a new node with value 7.

→ So we will create a new _node

→ we will make prev as well as next of new_node point to new_node

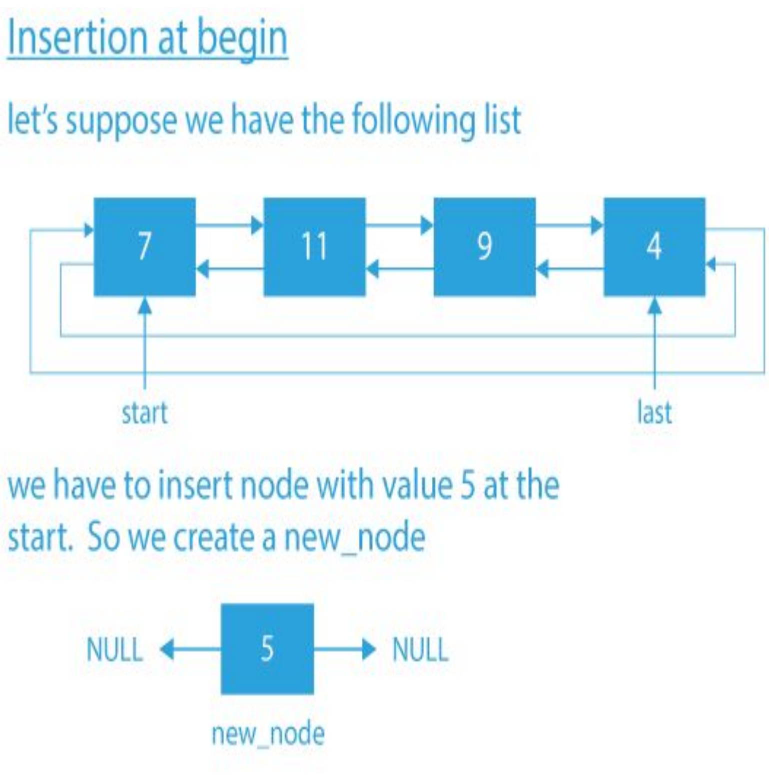
→ finally we will make start point to the new_node

Insertion at begin

let's suppose we have the following list

we have to insert node with value 5 at the start. So we create a new_node

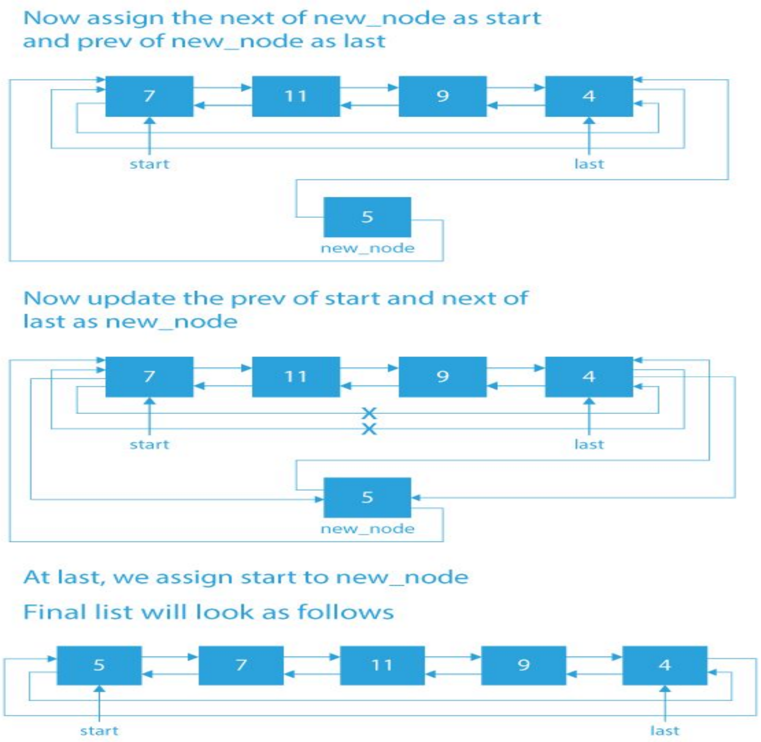
Insertion in beginning



Insertion at begin

let's suppose we have the following list

we have to insert node with value 5 at the start. So we create a new_node



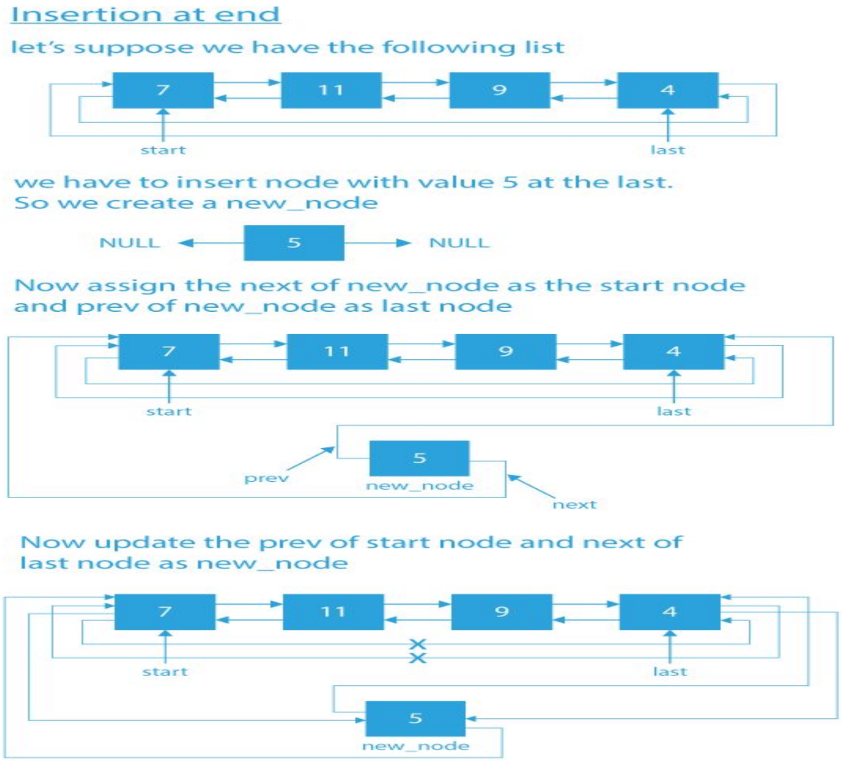
Now assign the next of new_node as start and prev of new_node as last

Now update the prev of start and next of last as new_node

At last, we assign start to new_node

Final list will look as follows

Insertion in end



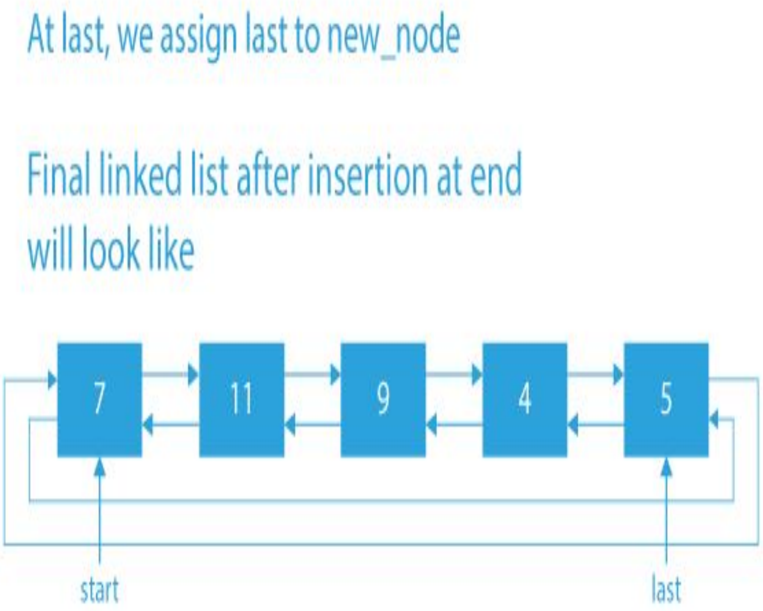
Insertion at end

let's suppose we have the following list

we have to insert node with value 5 at the last.
So we create a new_node

Now assign the next of new_node as the start node
and prev of new_node as last node

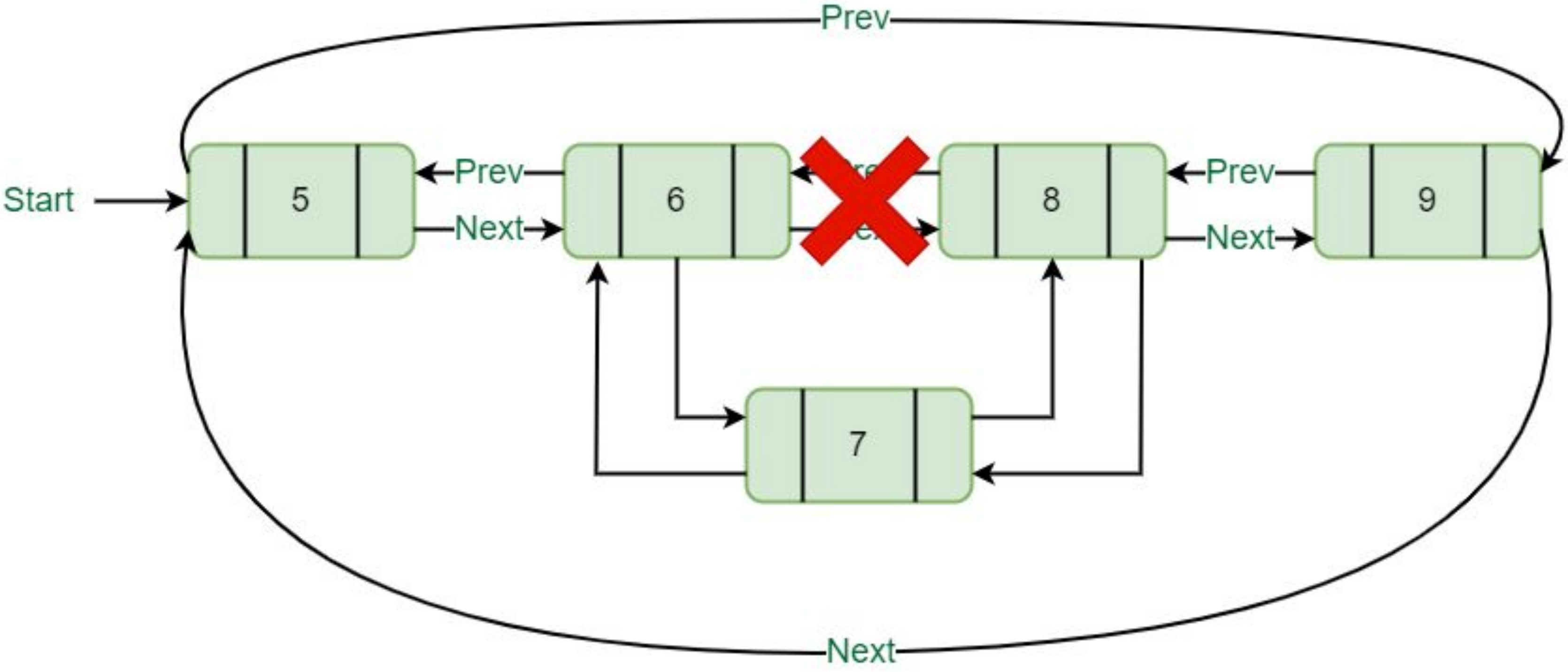
Now update the prev of start node and next of
last node as new_node



At last, we assign last to new_node

Final linked list after insertion at end
will look like

Insertion in specified position



Insertion in specified position

1. Create a new node dynamically.
2. Read the value and position of new node.
3. Set newnode - > info = data and newnode - > prev = NULL, newnode - > next = NULL
4. Check if the list is empty i.e head == NULL.
    a. Set newnode - > next = newnode
    b. Set newnode - > prev = newnode
    c. head = newnode
5. Otherwise
    a. Set ptr1 and ptr2 to head i.e. ptr1 = head, ptr2 = head
    b. Traverse the list to the given position and set ptr1 and ptr2 before and after of the position.

```
for(i=1; i<pos-1; i++) {

        ptr1 = ptr1 - > next;

        ptr2 = ptr1 - > next;

}
```

    c. Connect the links:
        i. Set ptr1 - > next = newnode
        ii. Set newnode - > prev = ptr1
        iii. Set ptr2 - > prev = newnode
        iv. newnode - > next = ptr2

NOTE: In the figure,
node with value 6 = ptr1
node with value 8 = ptr2