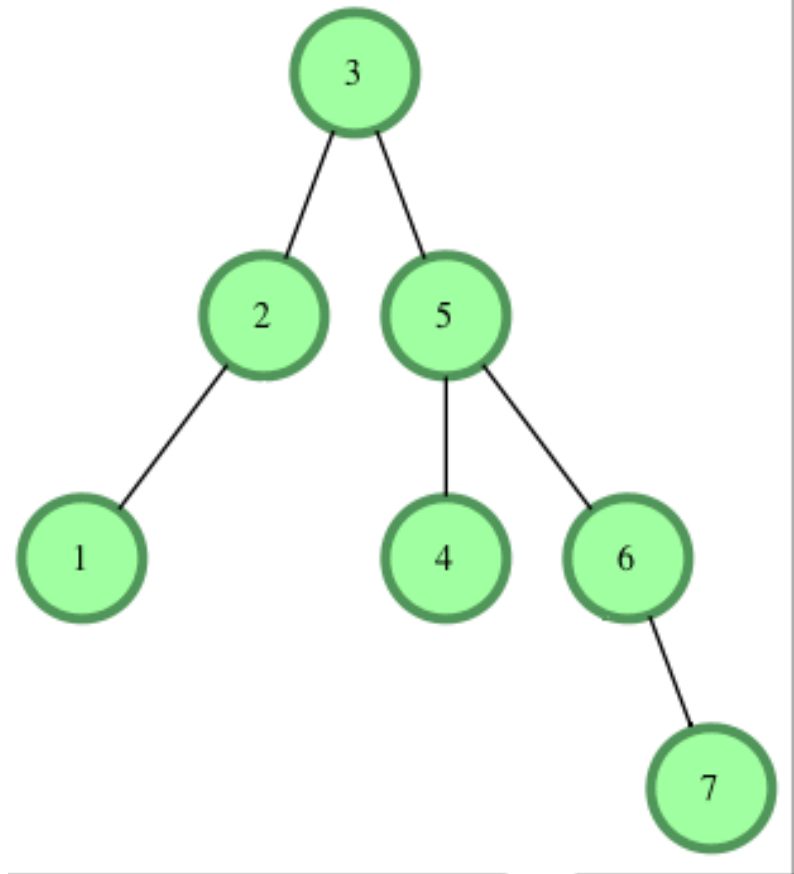


Data Structures- Binary Tree into Circular Doubly- Linked List

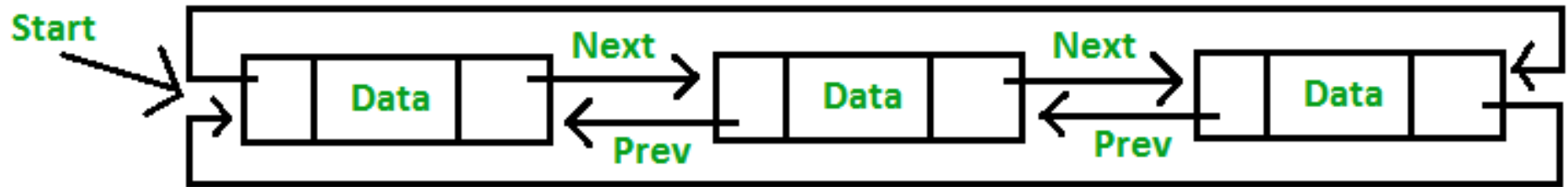
By:
Bhavishya K H
Supreet Ronad

BINARY TREE

A binary tree is a tree data structure in which each node has at most two children, which are referred to as the left child and the right child.



CIRULAR LINKED LIST



Has properties of both Circular Linked List AND Doubly Linked List.
Head Node:

- previous pointer points to last node
- next pointer points to next node

Last Node:

- Previous pointer: Points to previous node (as usual)
- Next pointer: Points to the head node

Hence forming a circular linked list

We take input tree from user

1. Ask for root node
2. Ask for left node of root node
3. Ask for right node of node

Step 2 and 3 are performed recursively for every node in the tree.

Input -1 given if **NO** left child/right child for a node exists.

Next, we perform **INORDER** traversal on the obtained binary tree, and nodes are simultaneously added to the Circular Doubly Linked List in this order.

This achieved by the following functions:

```

void inorder(node *t){
    if(t==NULL)
        return;
    inorder(t->left);
    toDCLL(t->data);
    printf("%d ",t-
>data);
    inorder(t->right);
}

```

```

void toDCLL(int data){
    Cll *temp=(Cll
*)malloc(sizeof(Cll));
    temp->data=data;
    temp->next=temp-
>prev=NULL;

    if(head==NULL){
        head=temp;
        temp->next=temp;
        temp->prev=temp;
        return;
    }
    head->prev=temp;
    temp->next=head;
    tail->next=temp;
    temp->prev=tail->prev;
    tail=temp;
}

```

