DATA STRUCTURES AND ITS APPLICATIONS UE19CS202

Shylaja S S & Kusuma K V
Department of Computer Science & Engineering

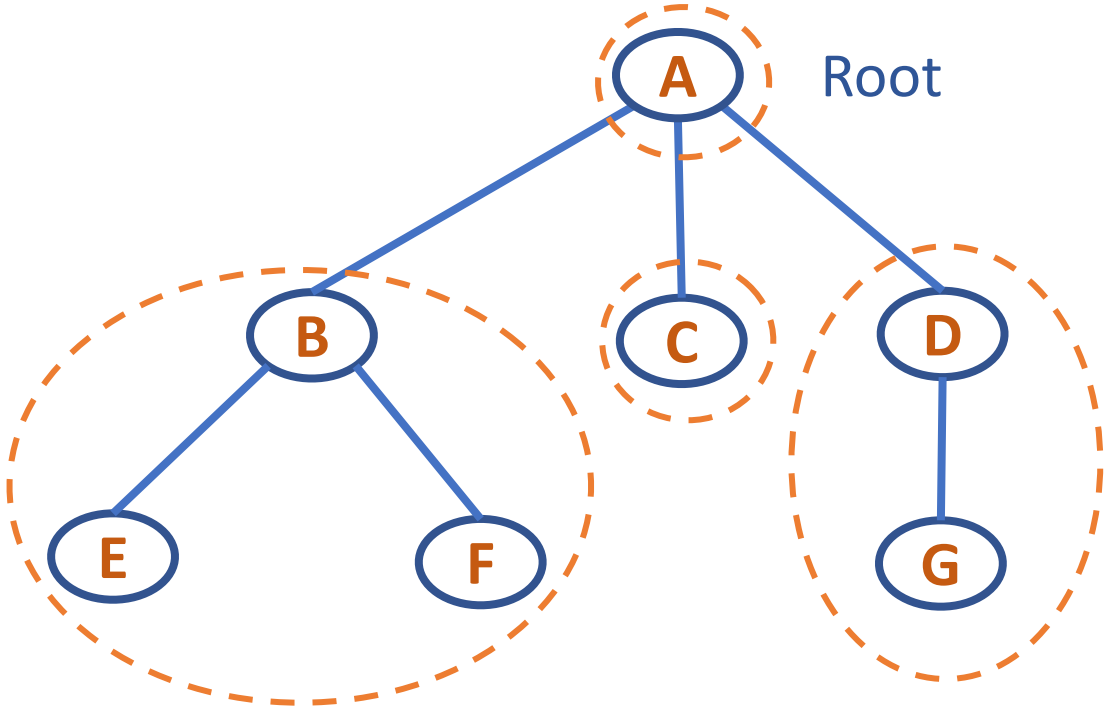
DATA STRUCTURES AND ITS APPLICATIONS

Basic Concept and Definitions: Trees

Shylaja S S
Department of Computer Science & Engineering

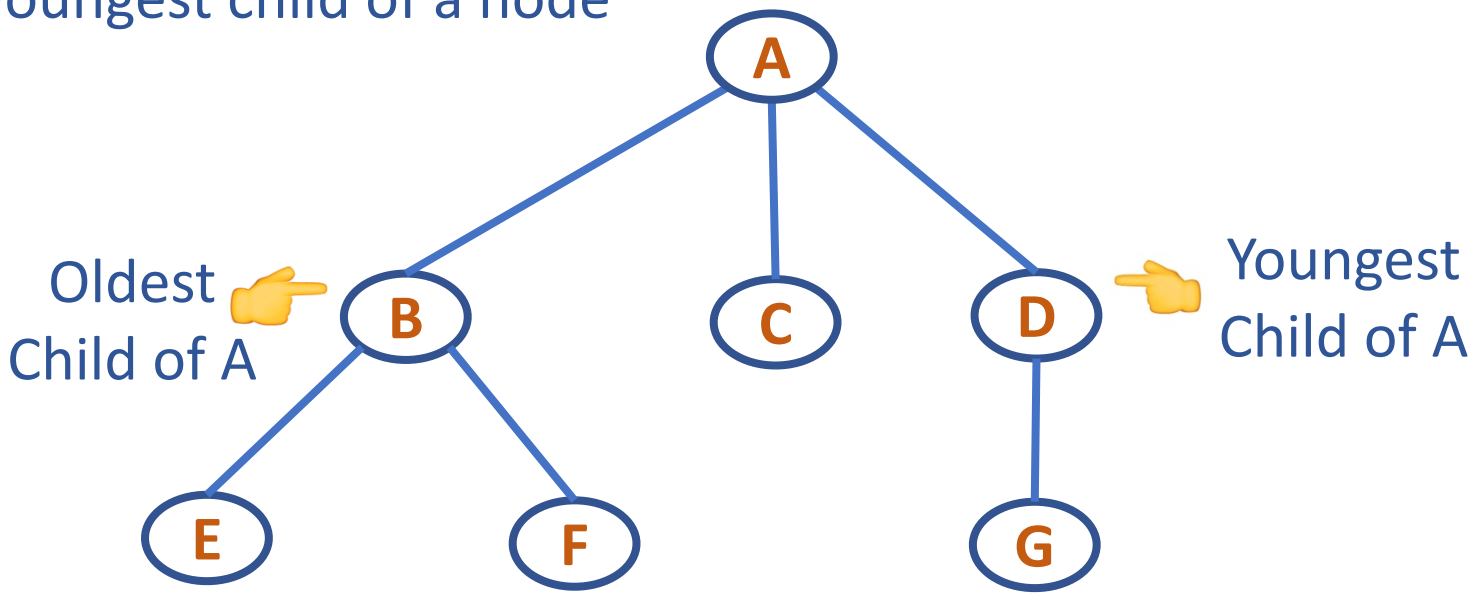
Trees

- Non Linear Data Structure
- Finite nonempty set of elements
  - One element is the root
  - Remaining elements are partitioned into m≥0 disjoint subsets each of which is itself a tree



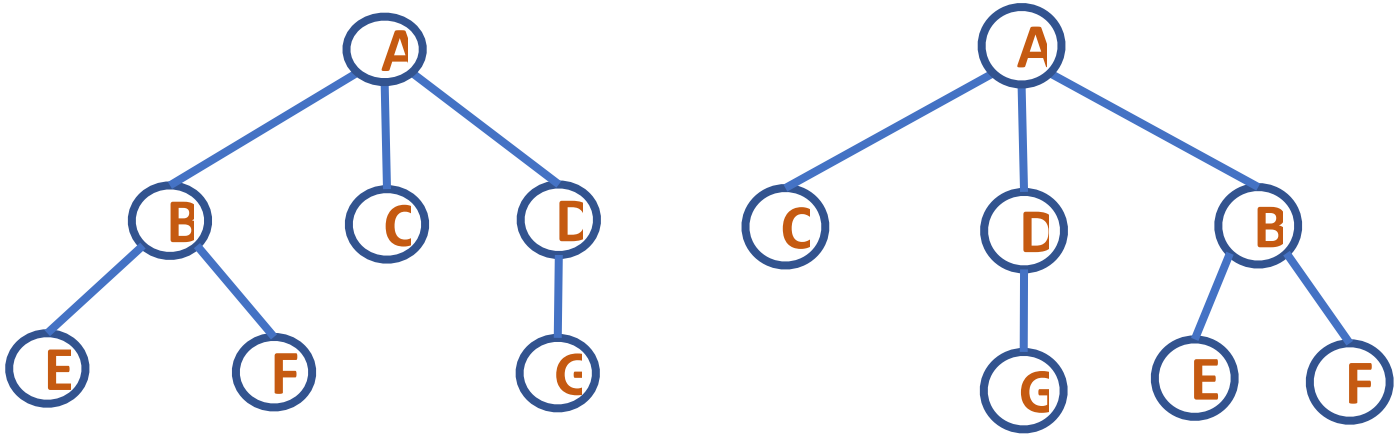
Trees

- Ordered Tree: a tree in which subtrees of each node forms an ordered set
- In such a tree we define first, second, ..., Last child of a particular node
- First child is called the oldest child and the last child the youngest child of a node



Trees

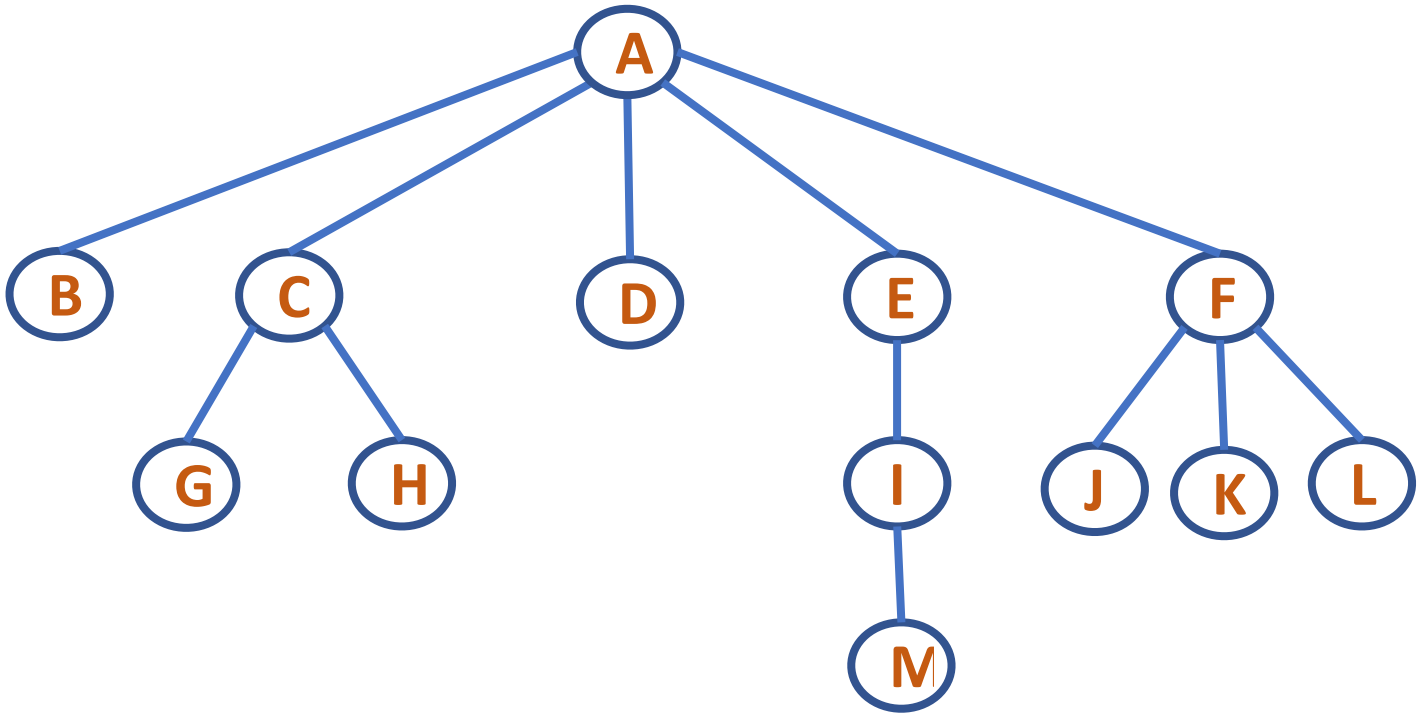
As unordered trees the below figures are equivalent but as ordered trees, they are different



N-ary Tree & Forest

n-ary tree: A rooted tree in which each node has no more than **n** children

- A binary tree is an n-ary tree with n=2
- n-ary tree with n=5



- Forest: is an ordered set of ordered trees

Tree

Representation of trees:

Tree node options:

```
struct treenode{
        int info;
        struct treenode *child[MAX];
};
```

where MAX is a constant

Restrictions with the above implementation:

- A node cannot have more than MAX children. Therefore cannot expand the tree

Tree

2nd implementation:

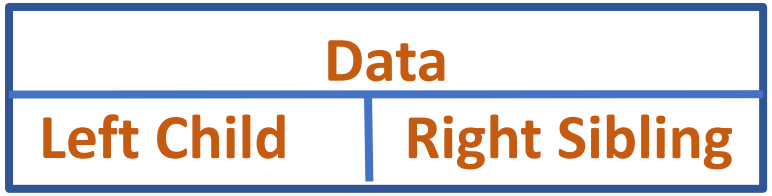
- All the children of a given node are linked and only the oldest child is linked to the parent
- A node has link to first child and a link to immediate sibling

```
struct treenode{
        int info;
        struct treenode *child;
        struct treenode *sibling;
};
```

Conversion of an n-ary Tree to a Binary Tree

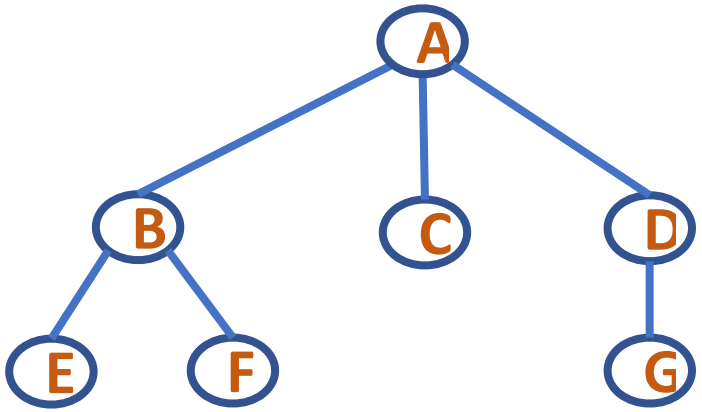
Left Child – Right Sibling Representation

- Link all the siblings of a node
- Delete all links from a node to its children except for the link to its leftmost child
- The left child in binary tree is the node which is the oldest child of the given node in an n-ary tree, and the right child is the node to the immediate right of the given node on the same horizontal line. Such a binary tree will not have a right sub tree
- The node structure corresponds to that of



DATA STRUCTURES AND ITS APPLICATIONS

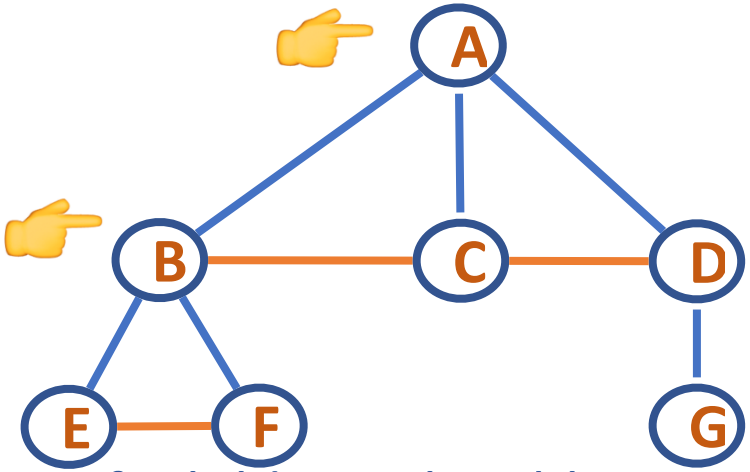
Conversion of an n-ary Tree to a Binary Tree



3-ary tree

Link all siblings of a Node

Delete all links from a Node to its children except for the link to its leftmost child

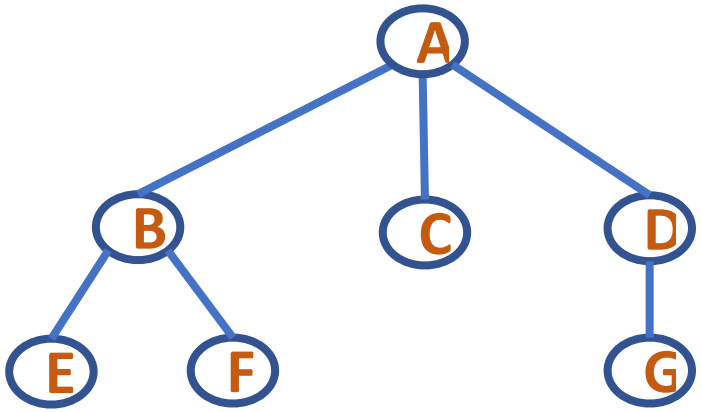


Left child – Right sibling

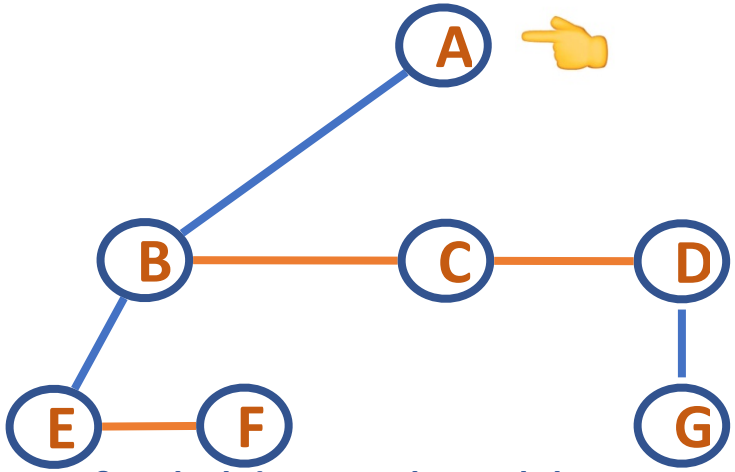
Representation of the above 3-ary tree

DATA STRUCTURES AND ITS APPLICATIONS

Conversion of an n-ary Tree to a Binary Tree

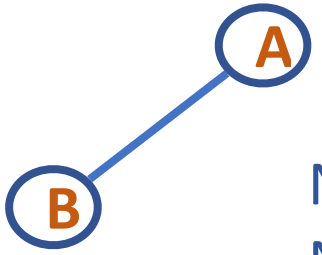


3-ary tree



Left child – Right sibling

Representation of the above 3-ary tree



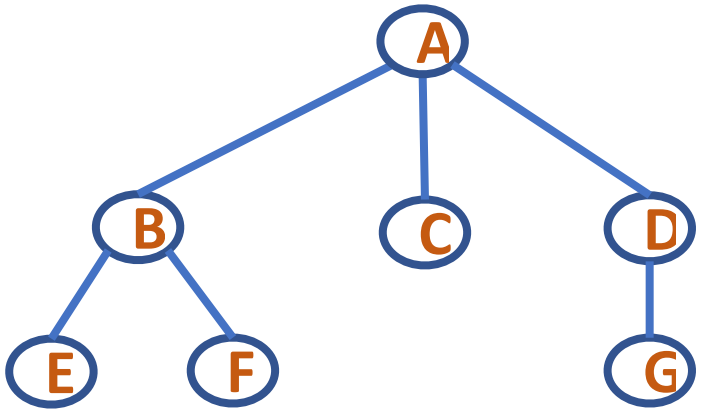
Node Below is B – Left child

No Right sibling so – no Right child

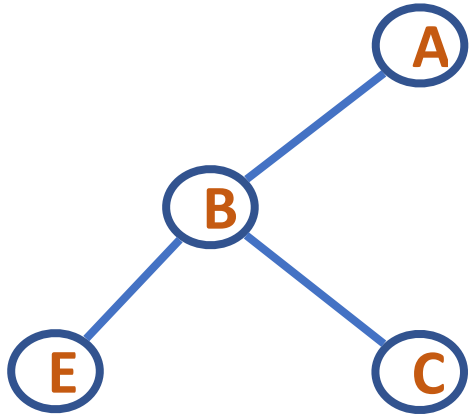
Corresponding binary tree

DATA STRUCTURES AND ITS APPLICATIONS

Conversion of an n-ary Tree to a Binary Tree

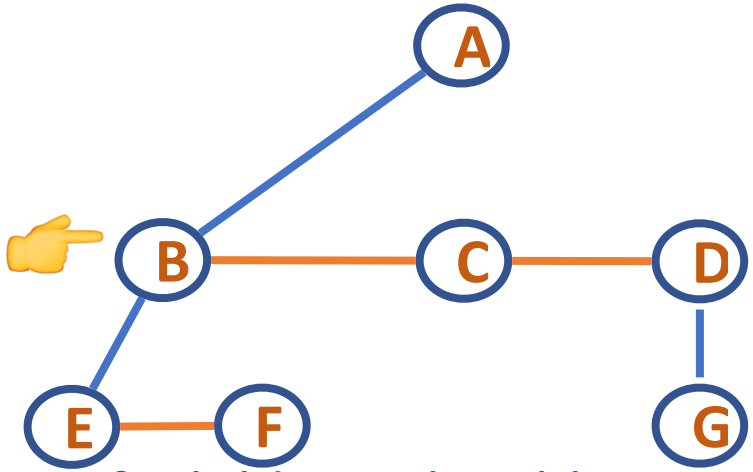


3-ary tree



Node Below is E – Left child

Next Right sibling is C – Right child



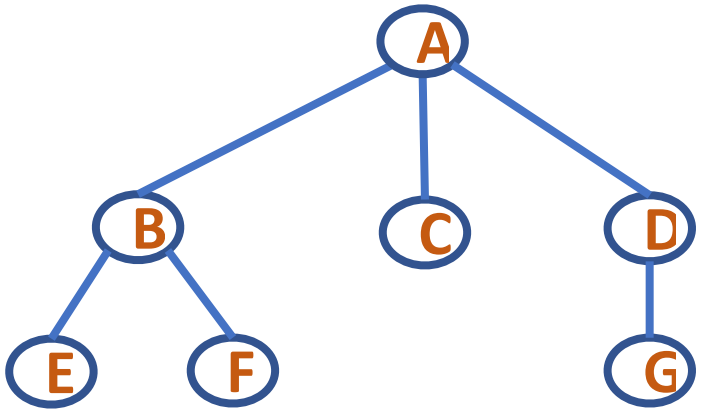
Left child – Right sibling

Representation of the above 3-ary tree

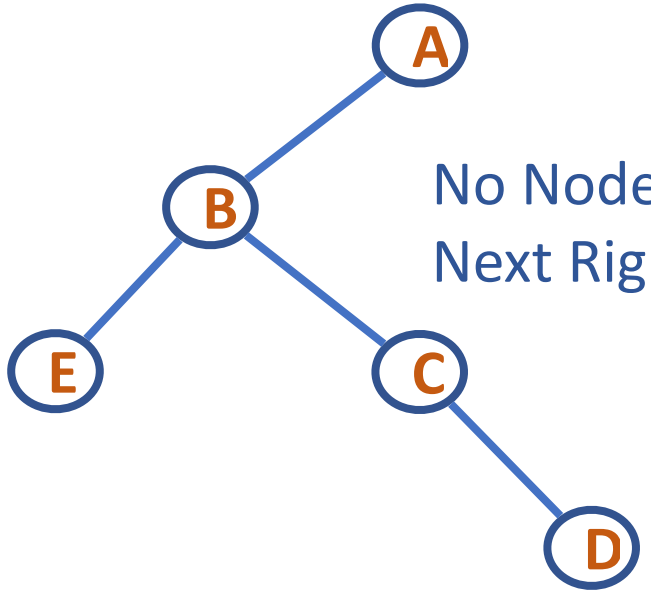
Corresponding binary tree

DATA STRUCTURES AND ITS APPLICATIONS

Conversion of an n-ary Tree to a Binary Tree

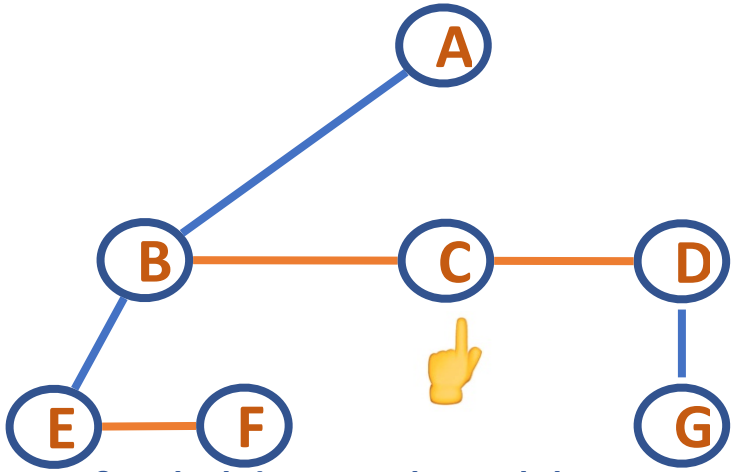


3-ary tree



No Node Below so – no Left child

Next Right sibling is D – Right child



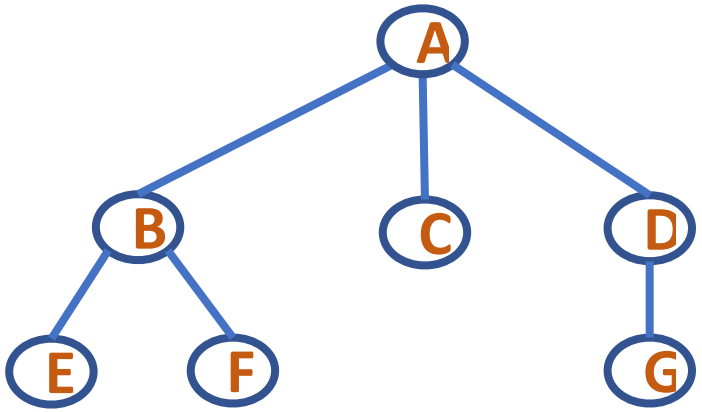
Left child – Right sibling

Representation of the above 3-ary tree

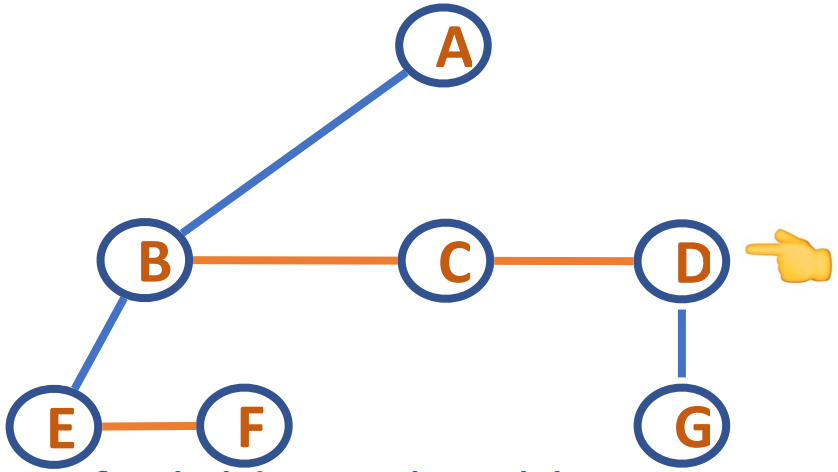
Corresponding binary tree

DATA STRUCTURES AND ITS APPLICATIONS

Conversion of an n-ary Tree to a Binary Tree

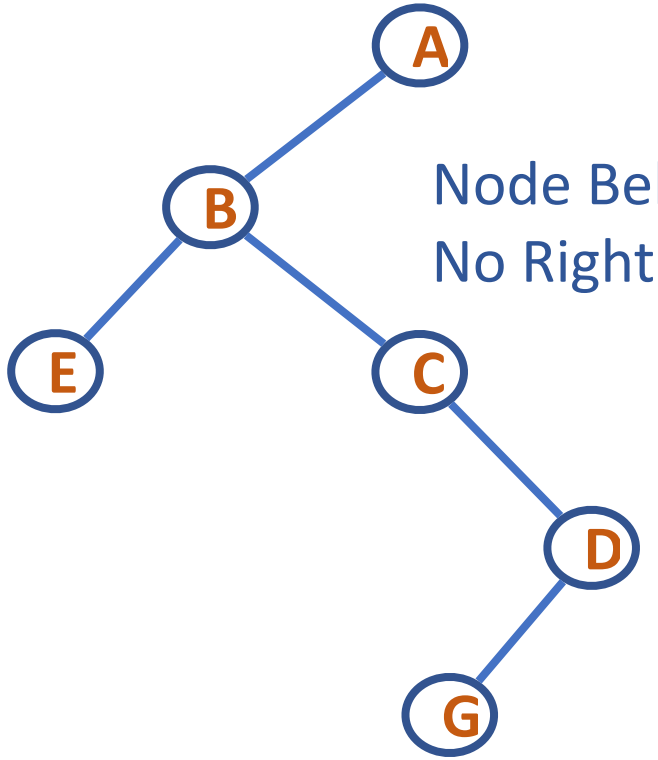


3-ary tree



Left child – Right sibling

Representation of the above 3-ary tree



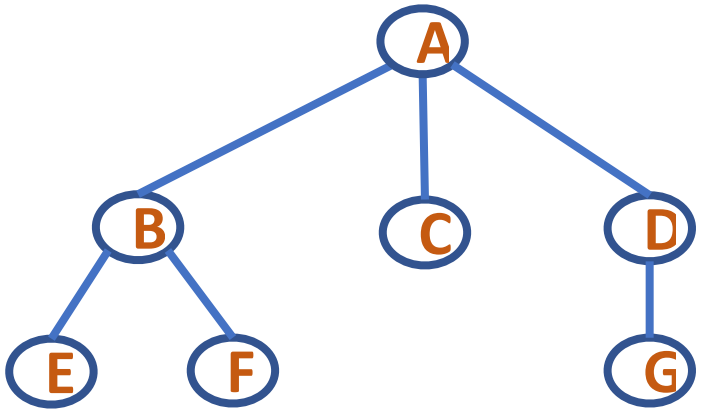
Node Below is G – Left child

No Right sibling so – no Right child

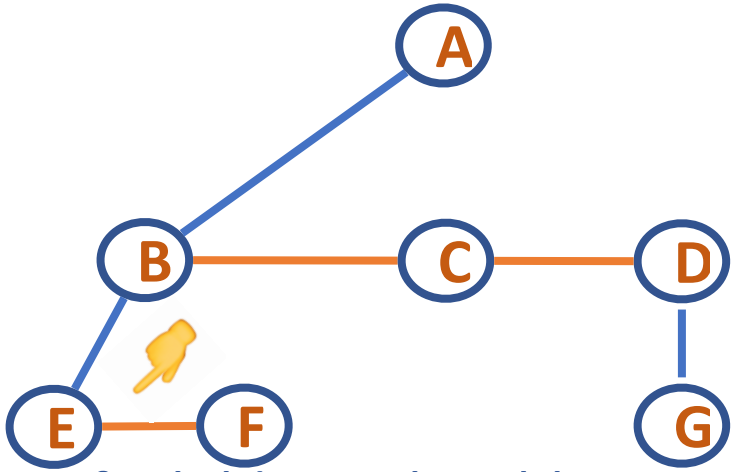
Corresponding binary tree

DATA STRUCTURES AND ITS APPLICATIONS

Conversion of an n-ary Tree to a Binary Tree

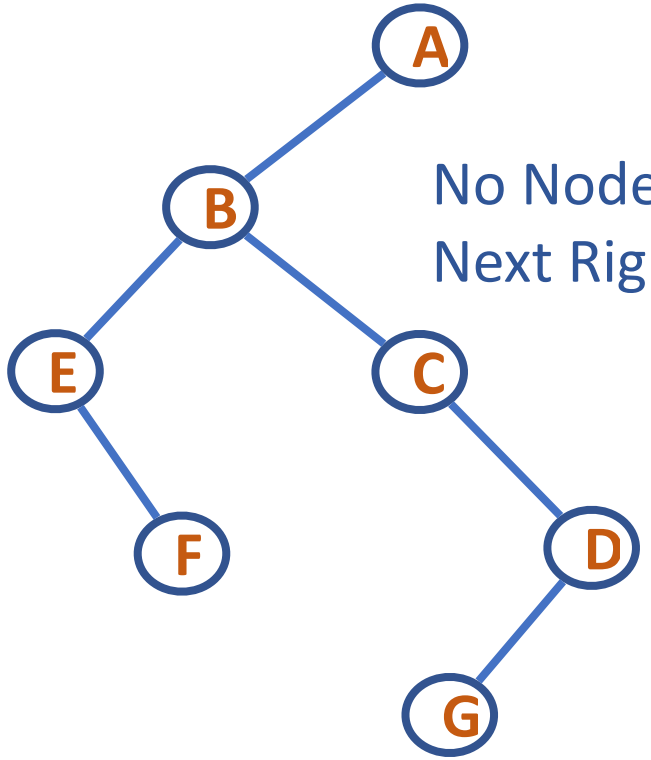


3-ary tree



Left child – Right sibling

Representation of the above 3-ary tree



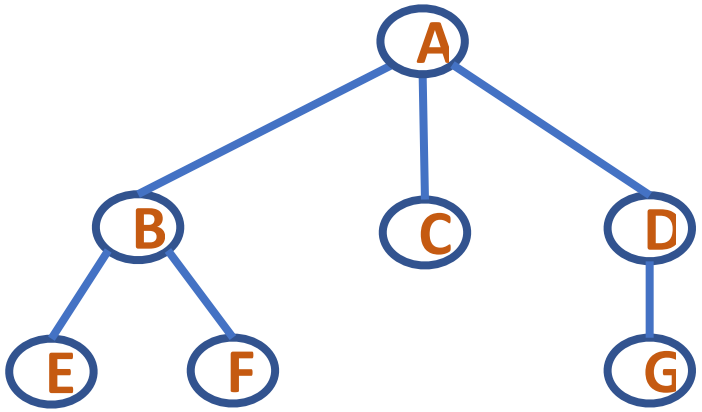
No Node Below so – no Left child

Next Right sibling is F – Right child

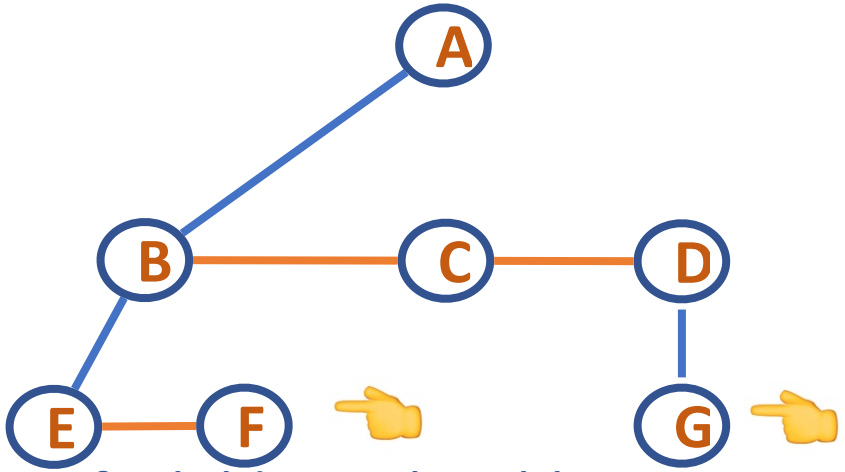
Corresponding binary tree

DATA STRUCTURES AND ITS APPLICATIONS

Conversion of an n-ary Tree to a Binary Tree

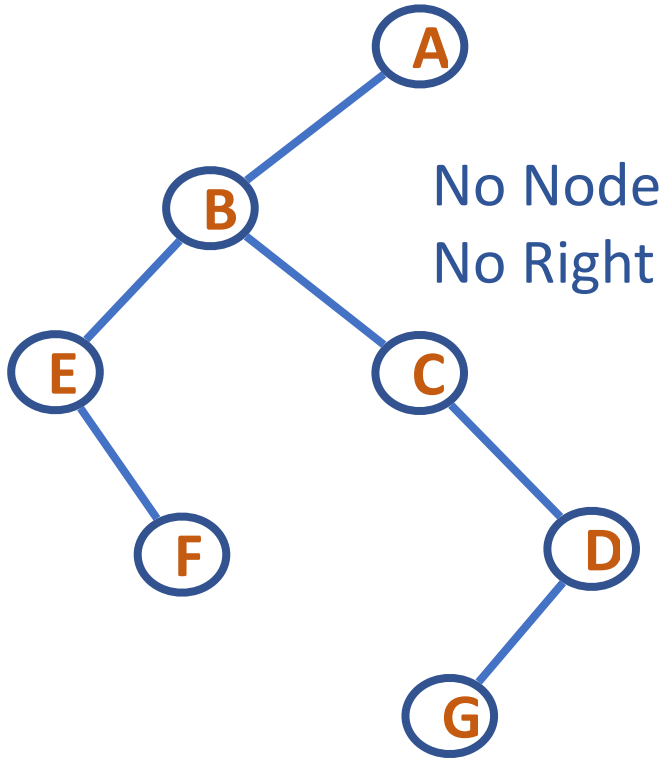


3-ary tree



Left child – Right sibling

Representation of the above 3-ary tree



No Node Below so – no Left child

No Right sibling so – no Right child

Corresponding binary tree

Conversion of a Forest to a Binary Tree

- Right Child of the root node of every resulting binary tree will be empty. This is because the root of the tree we are transforming has no siblings.

- On the other hand, if we have a forest then these can all be transformed into a single binary tree as follows:
    - First obtain the binary tree representation of each of the trees in the forest
    - Link all the binary trees together through the right sibling field of the root nodes

Conversion of a Forest to a Binary Tree

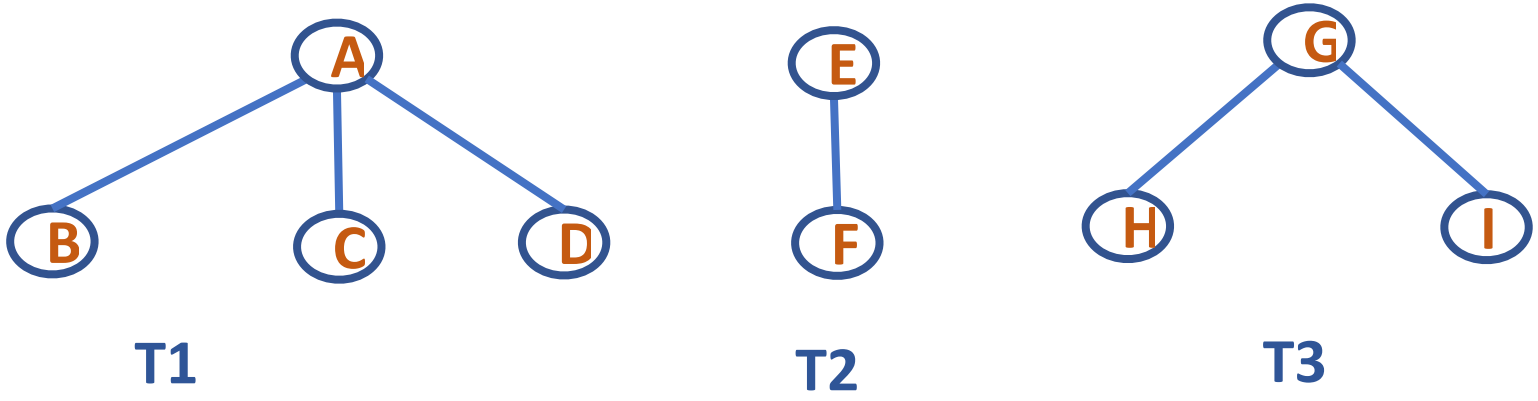
Conversion of a Forest to a Binary Tree can be formally defined as follows:

- If $T_1,...,T_n$ is a forest of n trees, then the binary tree corresponding to this forest, denoted by $B(T_1, ..., T_n)$:
  - is empty if $n = 0$
  - has root equal to root (T_1)
  - has left subtree equal to $B(T_{11}, T_{12}, ...,T_{1m})$ where T_{11} ,..., T_{1m} are the subtrees of root(T_1)
  - has right subtree $B(T_2, ..., T_n)$

DATA STRUCTURES AND ITS APPLICATIONS

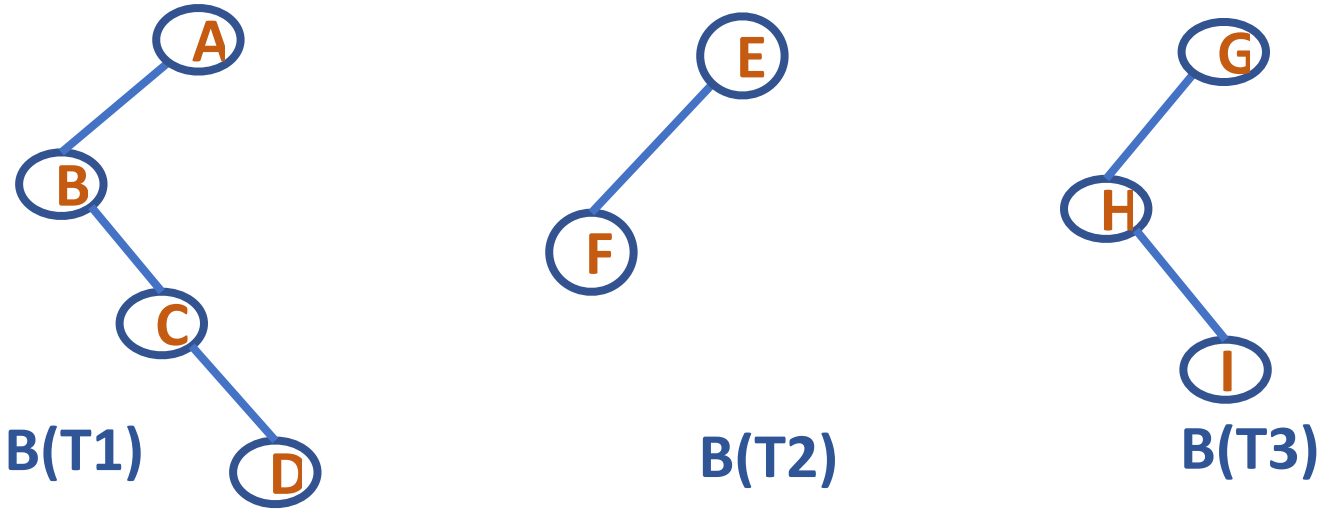
Conversion of a Forest to a Binary Tree

Consider the following Forest with three Trees



T1 T2 T3

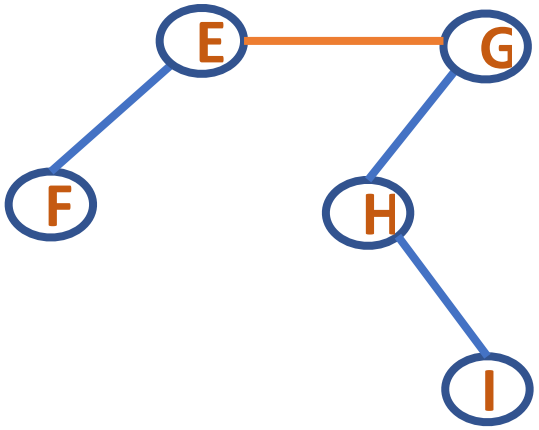
Corresponding Binary Trees



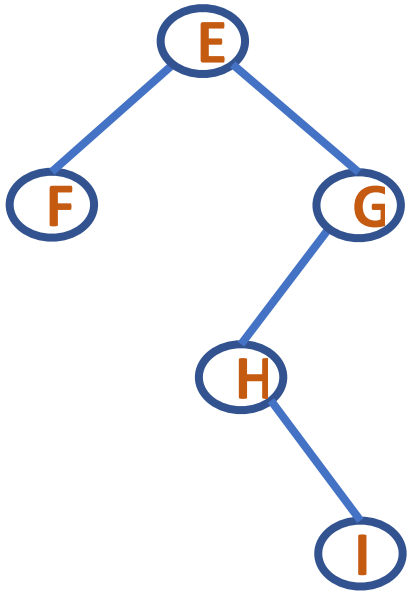
B(T1) B(T2) B(T3)

Conversion of a Forest to a Binary Tree

Link B(T2) and B(T3)



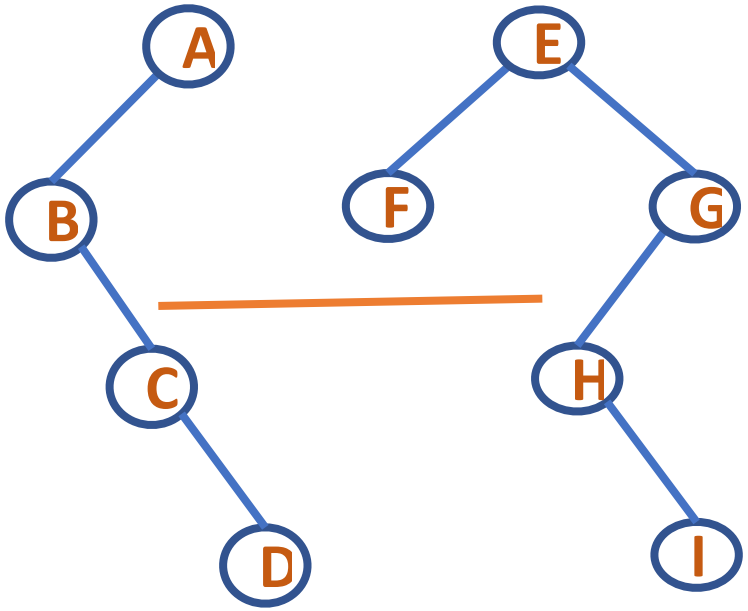
G becomes Right Child of E



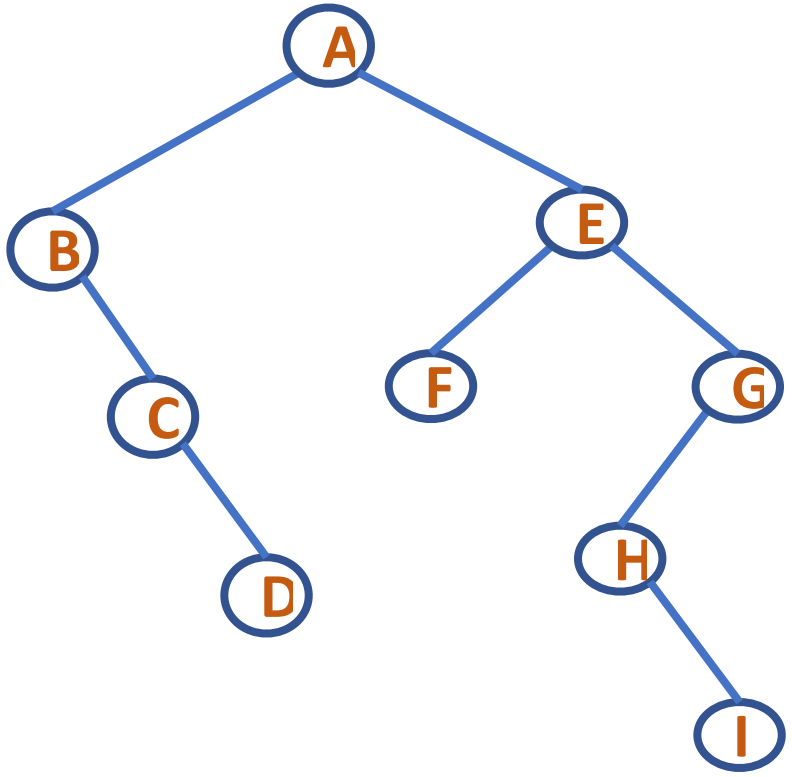
Corresponding Binary Tree
B(T2, T3)

Conversion of a Forest to a Binary Tree

Link B(T1) and B(T2, T3)



E becomes Right Child of A



Corresponding Binary Tree B(T1, T2, T3)

THANK YOU

Shylaja S S

Department of Computer Science & Engineering

shylaja.sharath@pes.edu

+91 9449867804