



**Tree**

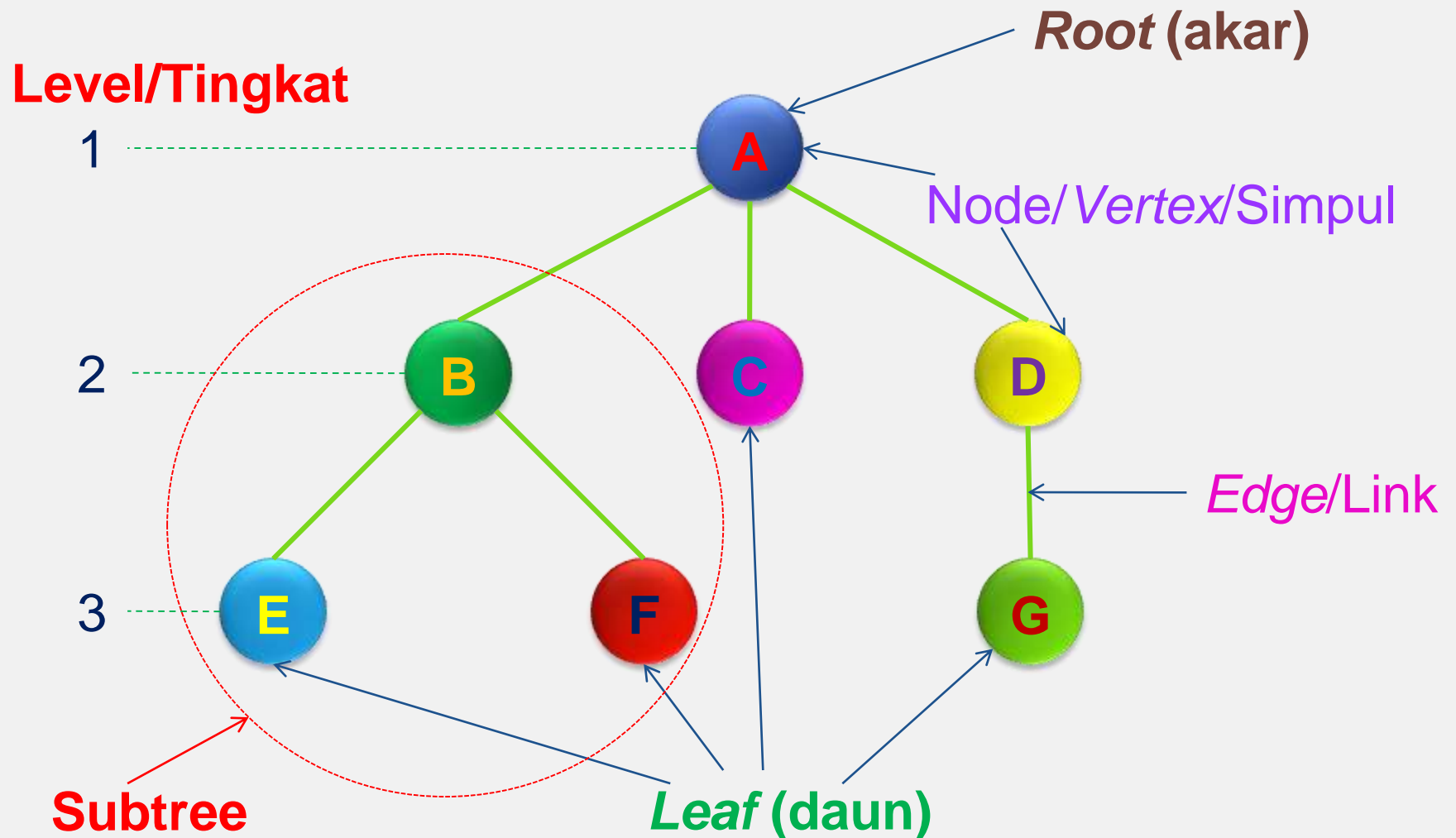
Adam M.B.

# DEFINITION

# Tree

Tree is data structure that is non linear and can be used to represents data in **hierarchy** between those elements. For example:  
organization structure, family tree, and the tournament.

# Components of Tree



# TERMINOLOGY

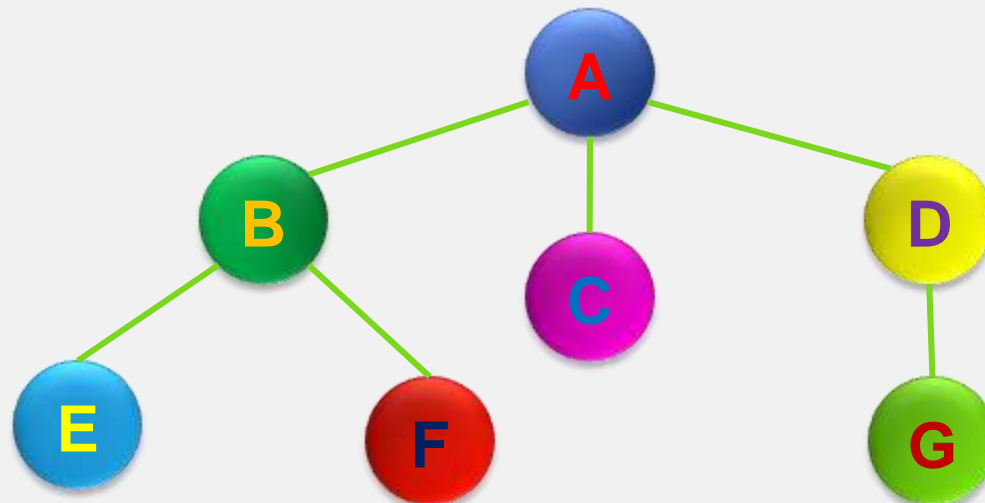
# Terminology of Tree

- **Predecessor** node that is above certain node.
- **Successor** node that is below certain node
- **Ancestor** all nodes that is before certain node and in the same path.
- **Descendant** all nodes that is after certain node and in the same path.

# Terminology of Tree

- **Parent** predecessor that is one level above certain node.
- **Sibling** nodes that have same parent
- **Degree** number of child in one node.

# Ilustration

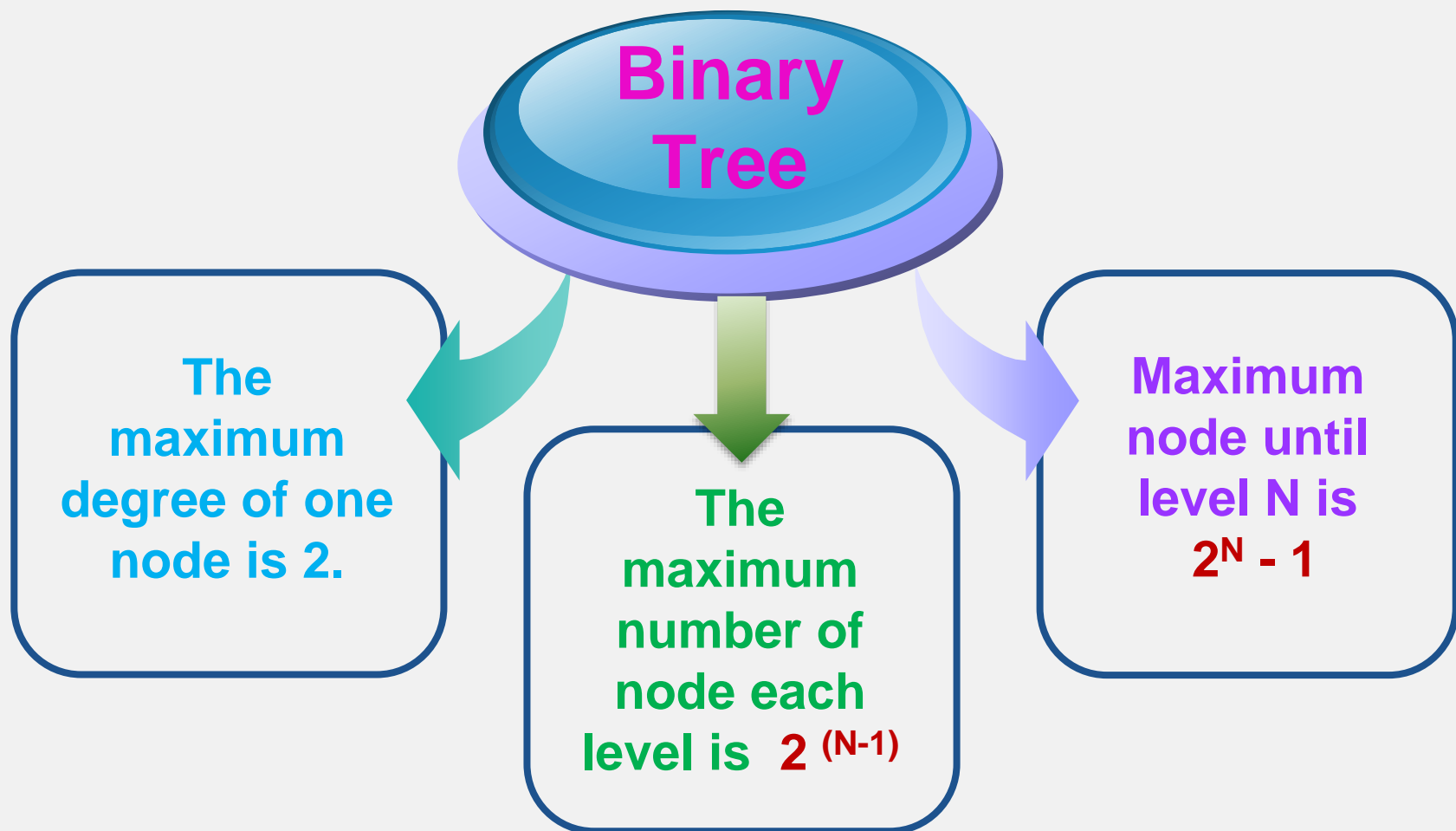


**Predecessor(B)** : A  
**Successor(A)** : B,C,D  
**Ancestor(E)** : B,A  
**Descendant(B)** : E,F

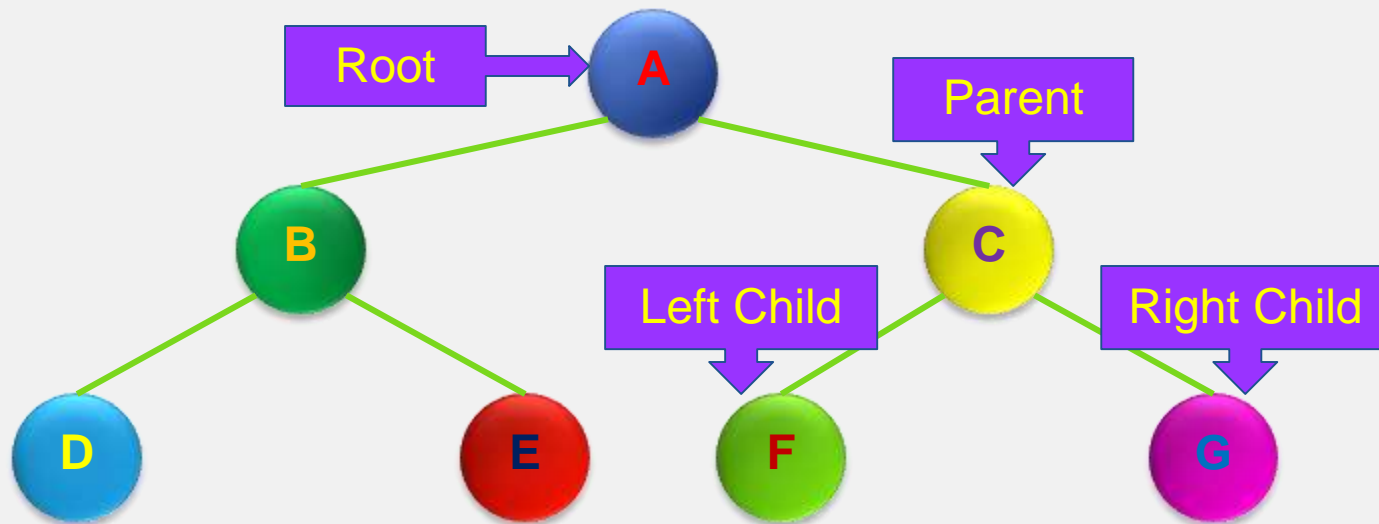
**Parent(E)** : B  
**Sibling(E)** : F  
**Degree(A)** : 3



# Binary Tree



# Binary Tree

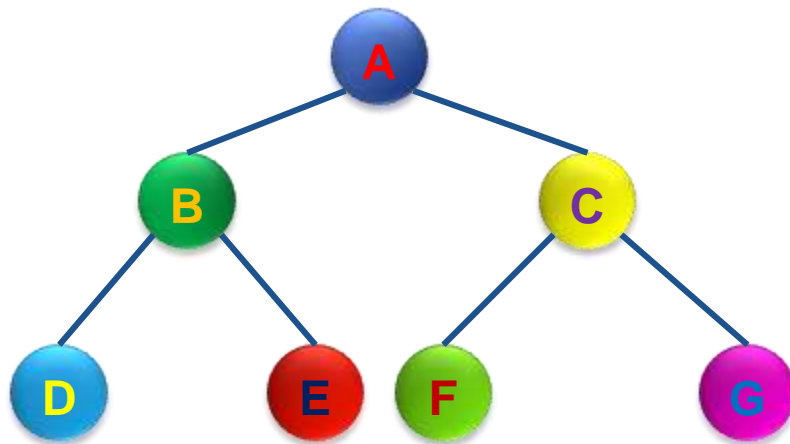


Maximum node on 3<sup>rd</sup>  
level  
 $= 2^{(N-1)}$   
 $= 2^{(3-1)}$   
 $= 2^2$   
 $= 4$

Maximum node until 3<sup>rd</sup>  
level  
 $= 2^N - 1$   
 $= 2^3 - 1$   
 $= 8 - 1$   
 $= 7$

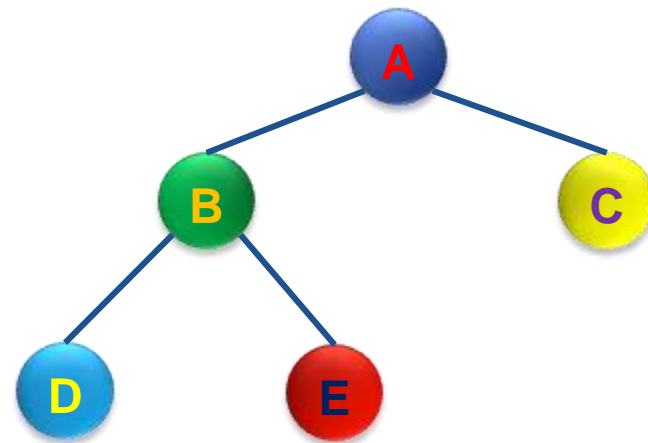
# Types of Binary Tree

## *Full Binary Tree*



- All nodes (except leaf) have two children.
- Each subtree has same length of path.

## *Complete Binary Tree*



- All nodes (except leaf) have two children.
- Each subtree can have different length of path.

# MAKING OF BINARY TREE

# Making of Binary Tree

- From input data
- From general tree
- From result of traversal process

# From Input Data

- If value of inserted node is bigger than parent then it will be right subtree.
- If value of inserted node is smaller than parent then it will be left subtree.
- This tree is known as binary search tree.

# From Input Data

Example: **HAKCBLJ**

**H** will be root

**A** < **H** :

**A** will be left child of **H**

**K** > **H** :

**K** will be right child of **H**

**C** < **H**  $\rightarrow$  **C** > **A** :

**C** will be right child of **A**

**B** < **H**  $\rightarrow$  **B** > **A**  $\rightarrow$  **B** < **C** :

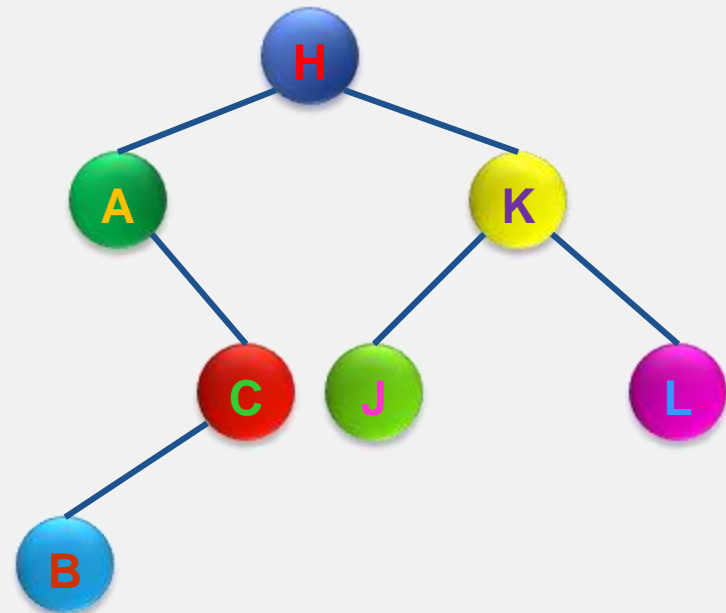
**B** will be left child of **C**

**L** > **H**  $\rightarrow$  **L** > **K** :

**L** will be right child of **K**

**J** < **H**  $\rightarrow$  **J** < **K** :

**J** will be left child of **K**



# Exercise

Make binary tree from these input data:

- GHCKJALBEFD
- KGMDLSBRJP

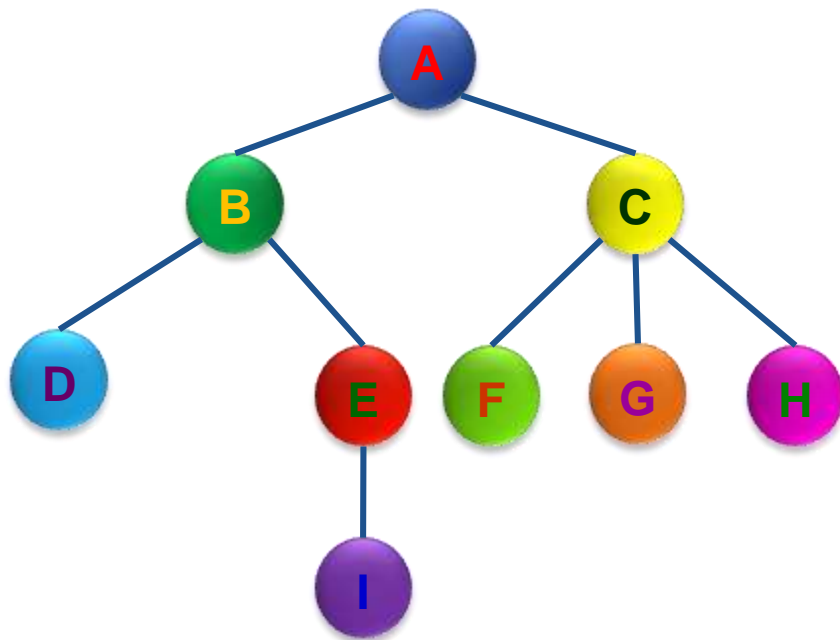


# From General Tree

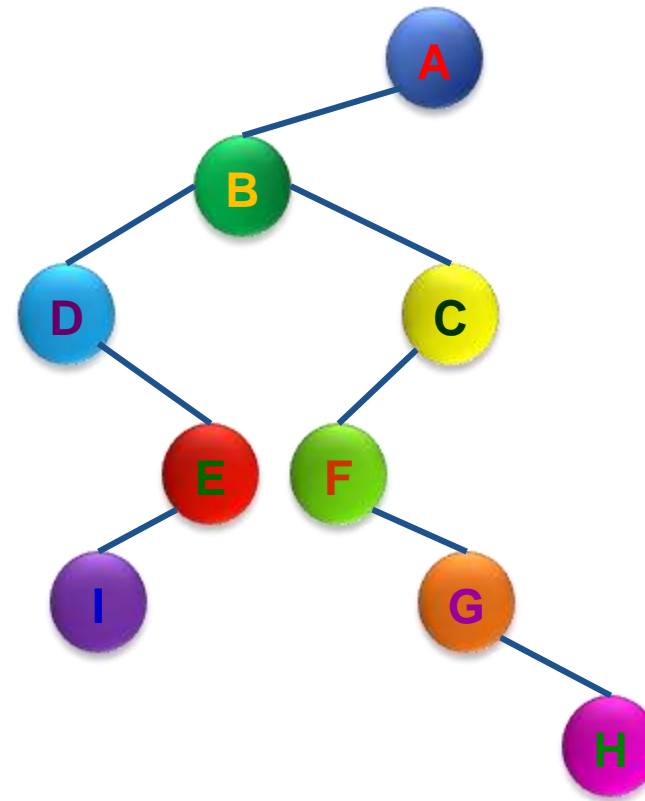
- First son in general tree will be left son in binary tree
- Next brother of first son in general tree will be right son in binary tree.

# From General Tree

**General Tree**

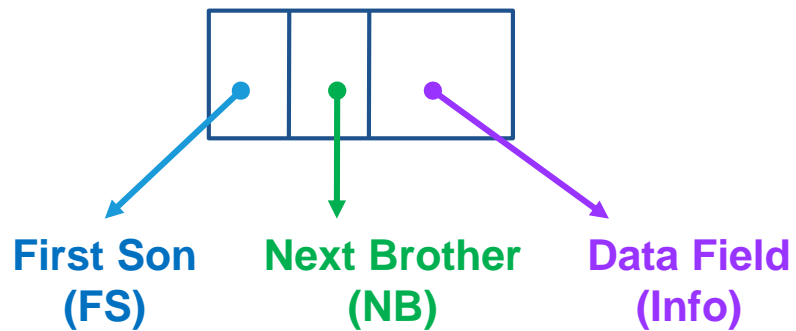


**Binary Tree**

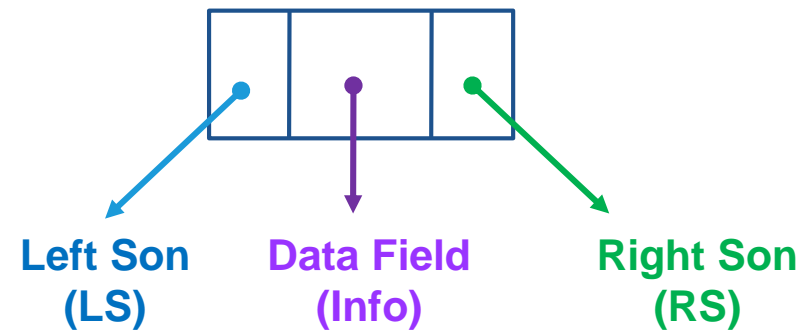


# From General Tree (Program)

## One node in general tree

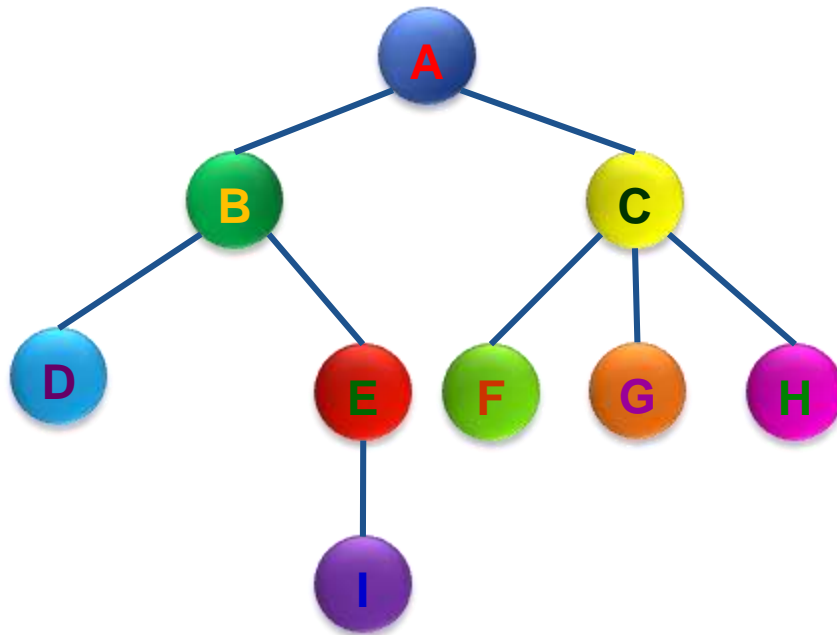


## One node in binary tree

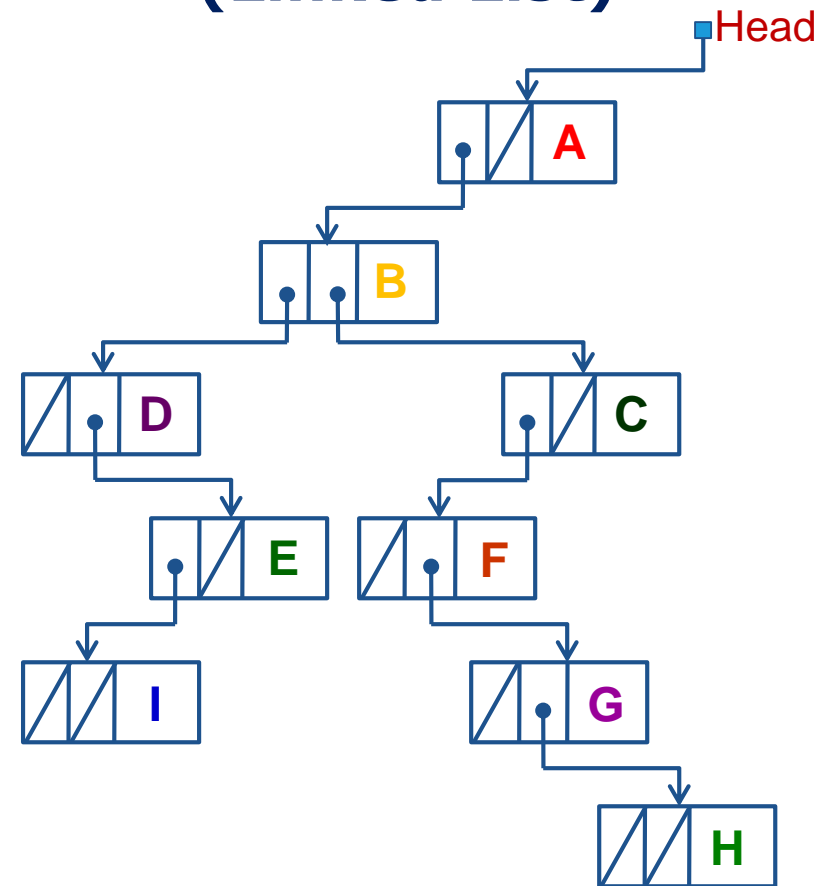


# From General Tree (Program)

**General Tree**

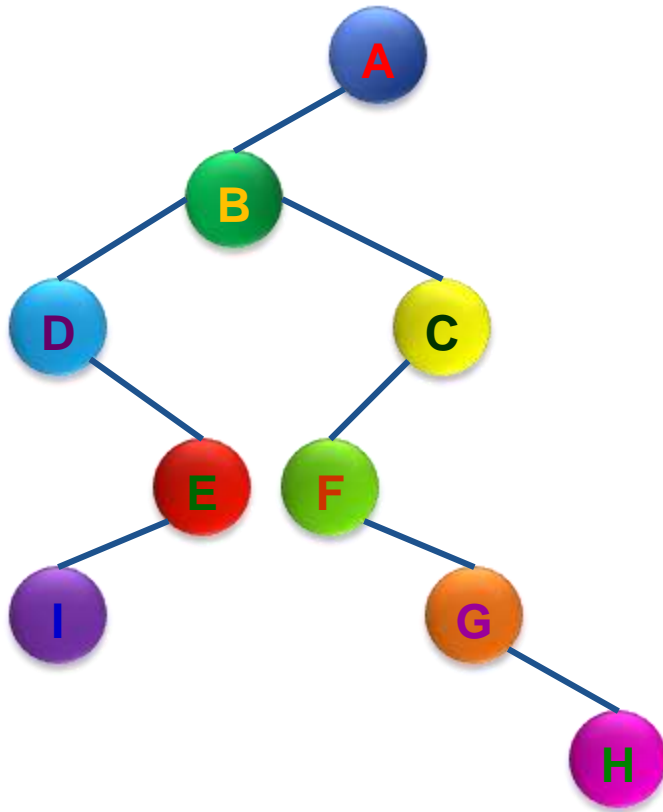


**General Tree  
(Linked List)**

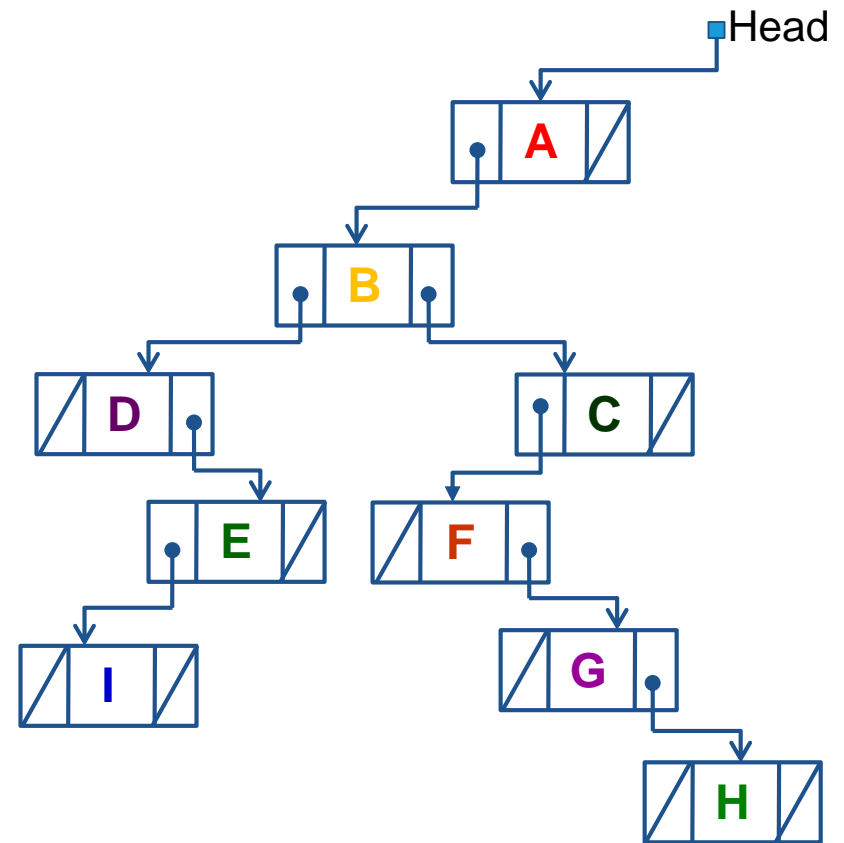


# From General Tree (Program)

**Binary Tree**

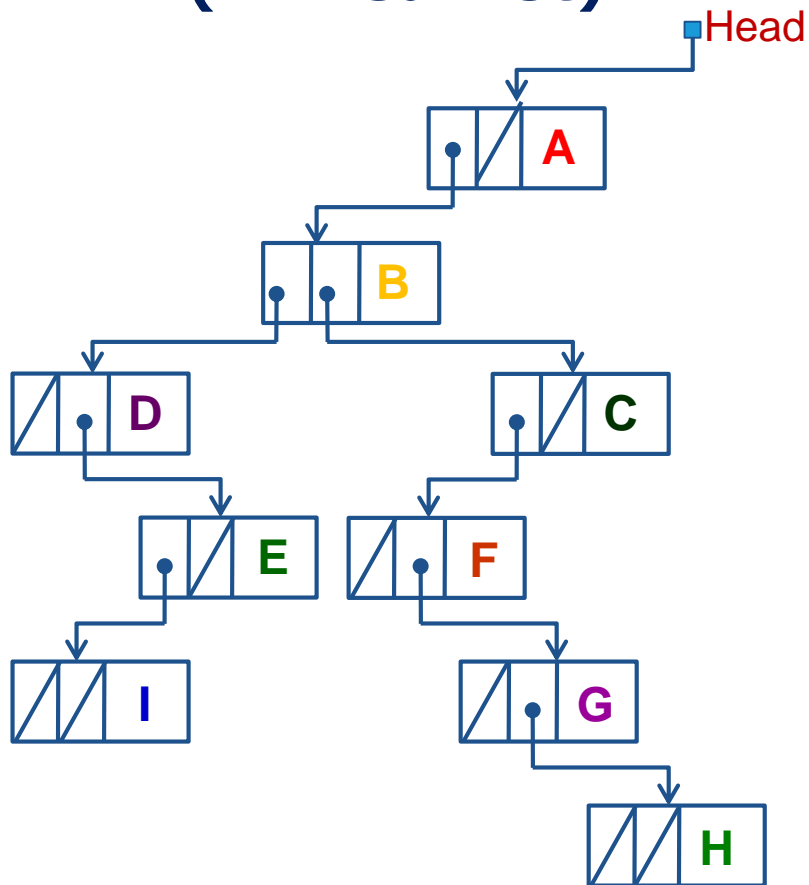


**Binary Tree  
(Linked List)**

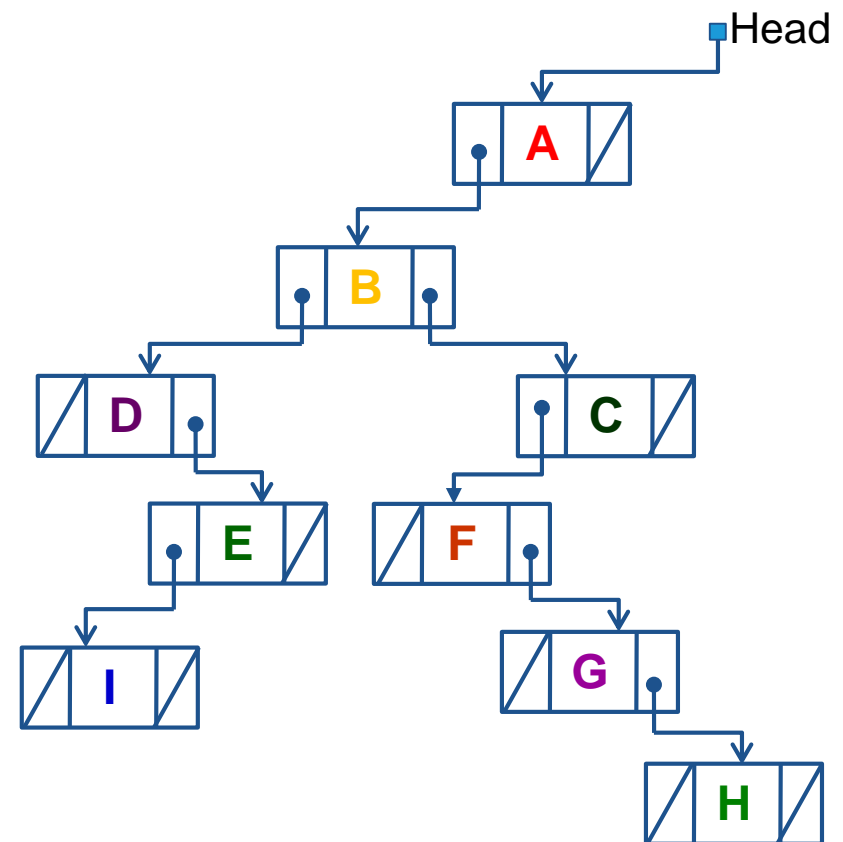


# From General Tree (Program)

**General Tree  
(Linked List)**

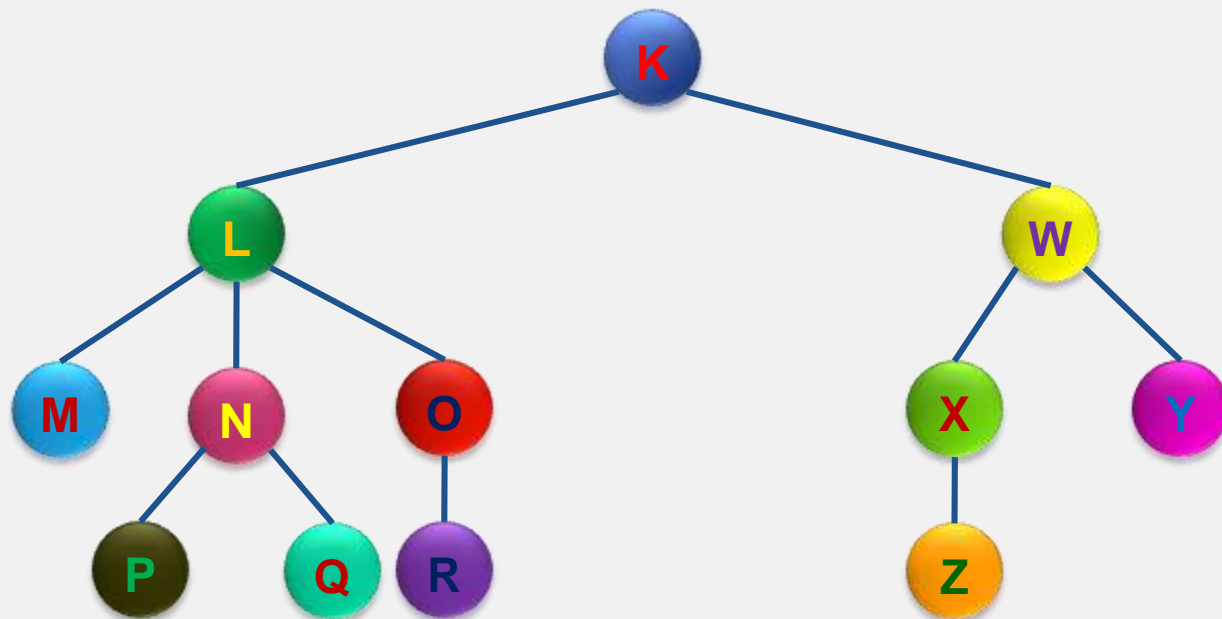


**Binary Tree  
(Linked List)**



# Exercise

Make binary tree from this general tree:



# Exercise

a. K,C,P,E,M

b.  $E = \frac{A + B}{G}$

Make binary tree from this statement:

- K, C, P, E, M, B, R, G, Q, F, W

- $E = \frac{A + BD^H - F}{G - K}$





**THANK YOU**

**GRACIAS**

**Contact Person:**

Adam Mukharil Bachtiar  
Informatics Engineering UNIKOM  
Jalan Dipati Ukur Nomor. 112-114 Bandung 40132  
Email: [adfbipotter@gmail.com](mailto:adfbipotter@gmail.com)  
Blog: <http://adfbipotter.wordpress.com>

**Copyright © Adam Mukharil Bachtiar 2012**