Trees and binary trees

Ch. 5

Tree

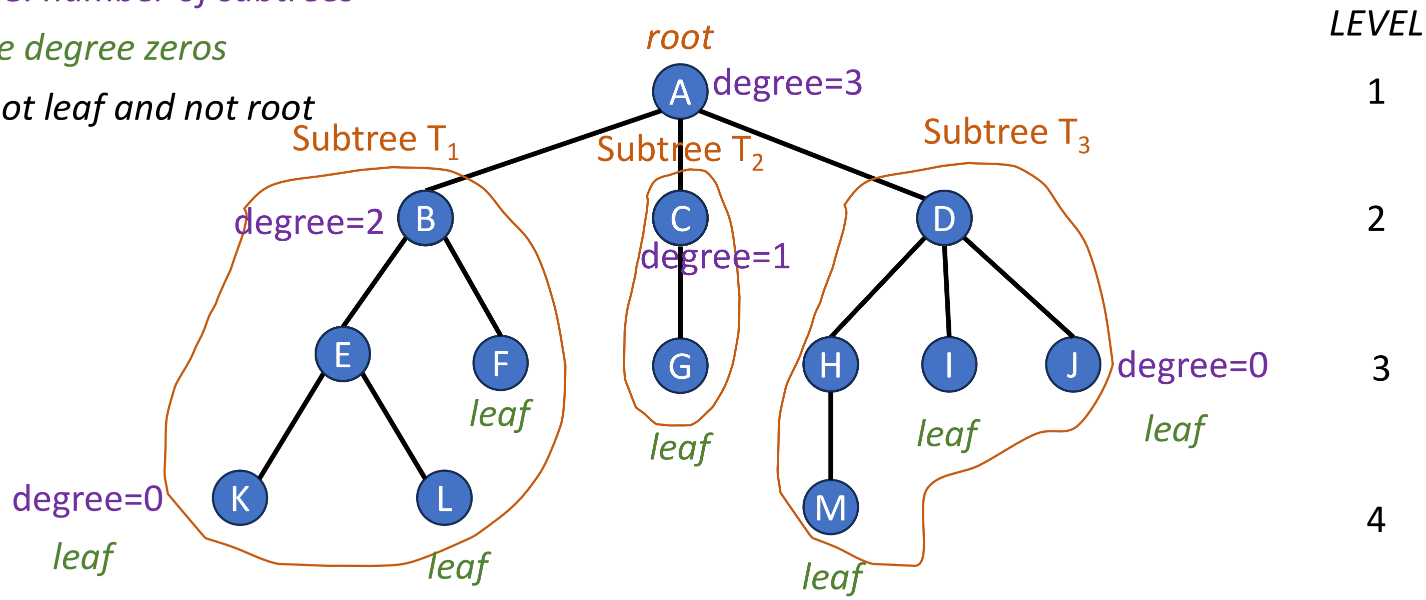
- A finite set of one or more nodes.
  - A node called *root*.
  - Remaining nodes are partitioned into disjoint sets, called subtrees.

A recursive definition

Degree of a node: *number of subtrees*

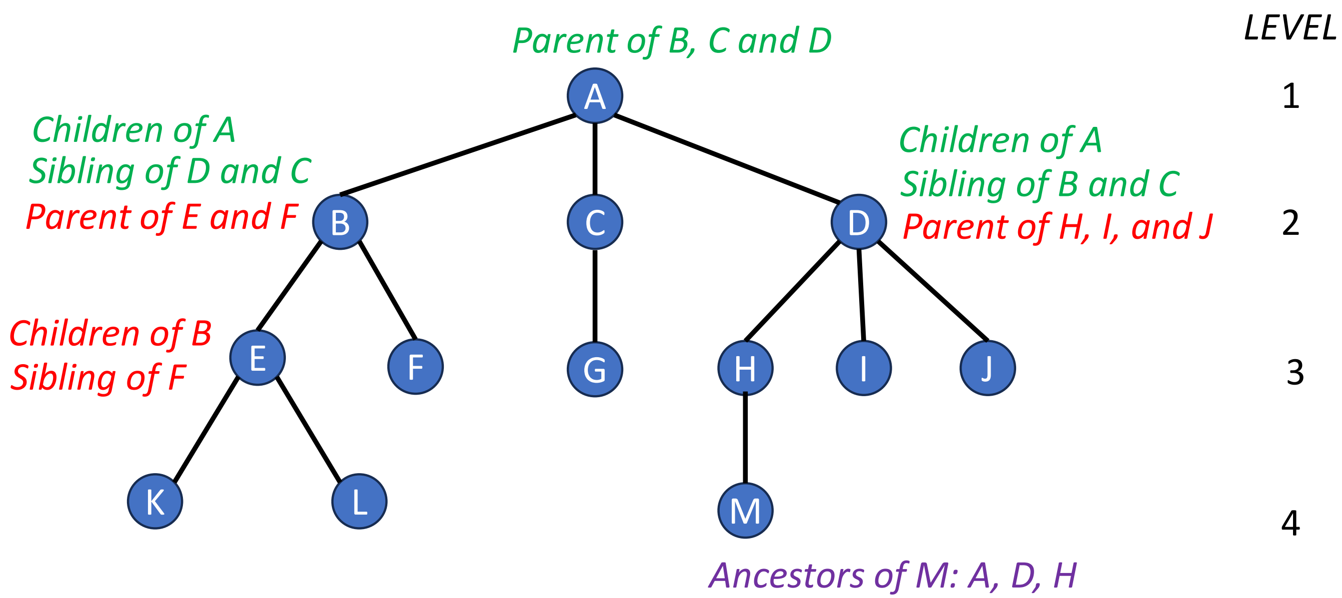
Leaf: *nodes have degree zeros*

Internal node: *not leaf and not root*



Tree

- Children: roots of subtrees of a node X
- Parent: X is parent
- Sibling: children of the same parent
- Ancestors: all the nodes along the path from the root to the node.

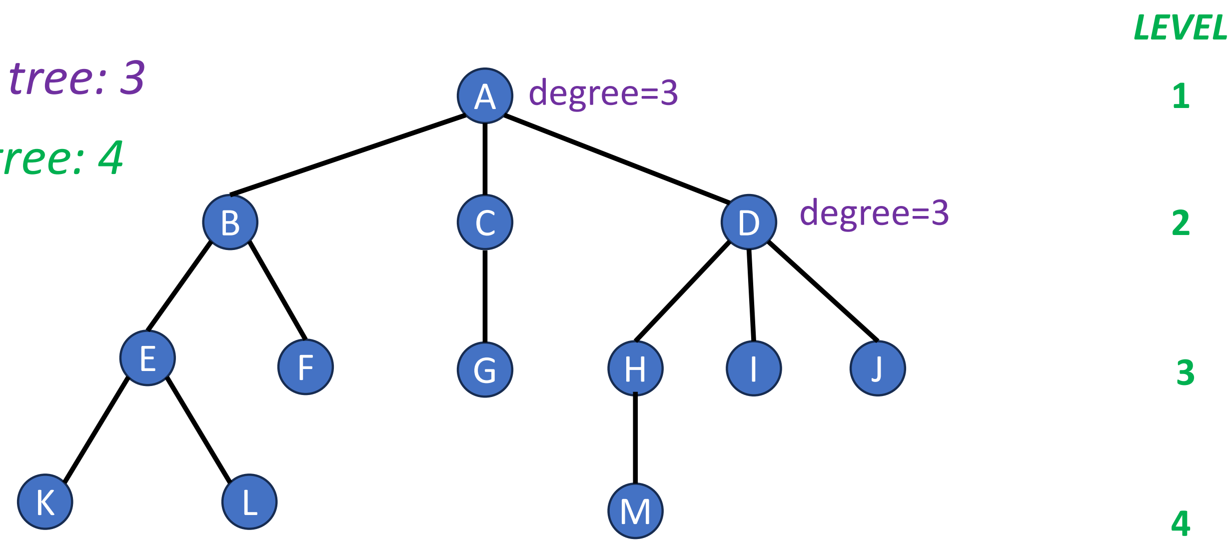


Tree

- Degree of a tree: maximum of the degree of the nodes in the tree
- Height (depth) of a tree: maximum level of any node in the tree

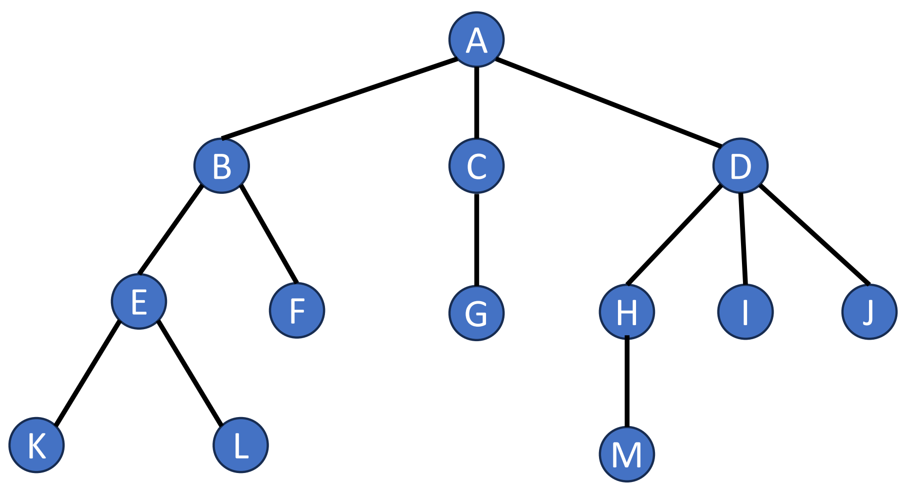
Degree of this tree: 3

Depth of this tree: 4



List representation

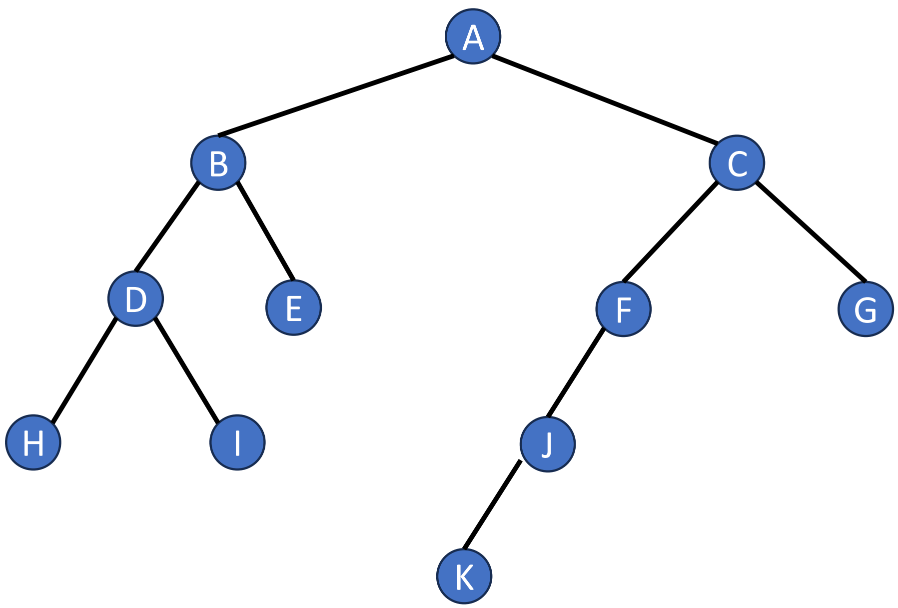
- Root node comes first
- Followed by a list of subtrees of that node



(A (B (E (K, L), F), C (G), D (H (M), I, J)))

Binary tree

- Degree-two tree
- Recursive definition:
  - Each node has left subtree and/or right subtree.
  - Left subtree (or right subtree) is a binary tree.



Please reply your answers of Q6 -Q11 via the following link:

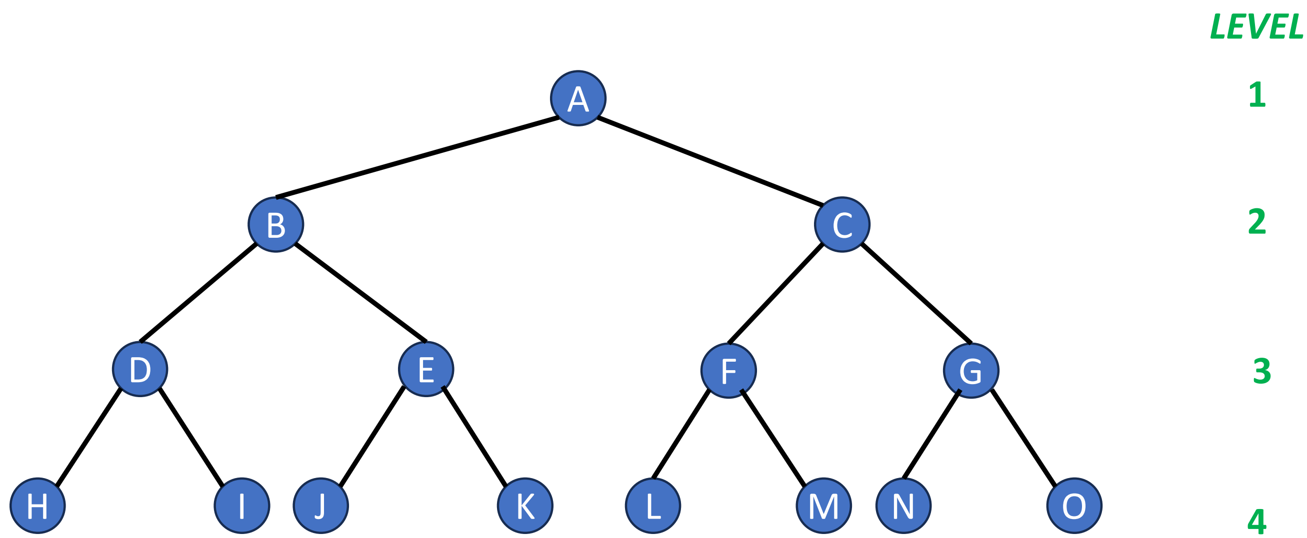
Group members: 1~3 people

Q6: The degree of the tree
Q7: The depth of the tree
Q8: List the leaf nodes
Q9: The sibling of node E
Q10: The children of node C
Q11: The level of node K

Full binary tree

Maximum number of nodes
$= 1 + 2 + 4 + 8 + \ldots + 2^{h-1}$
$= 2^h - 1$

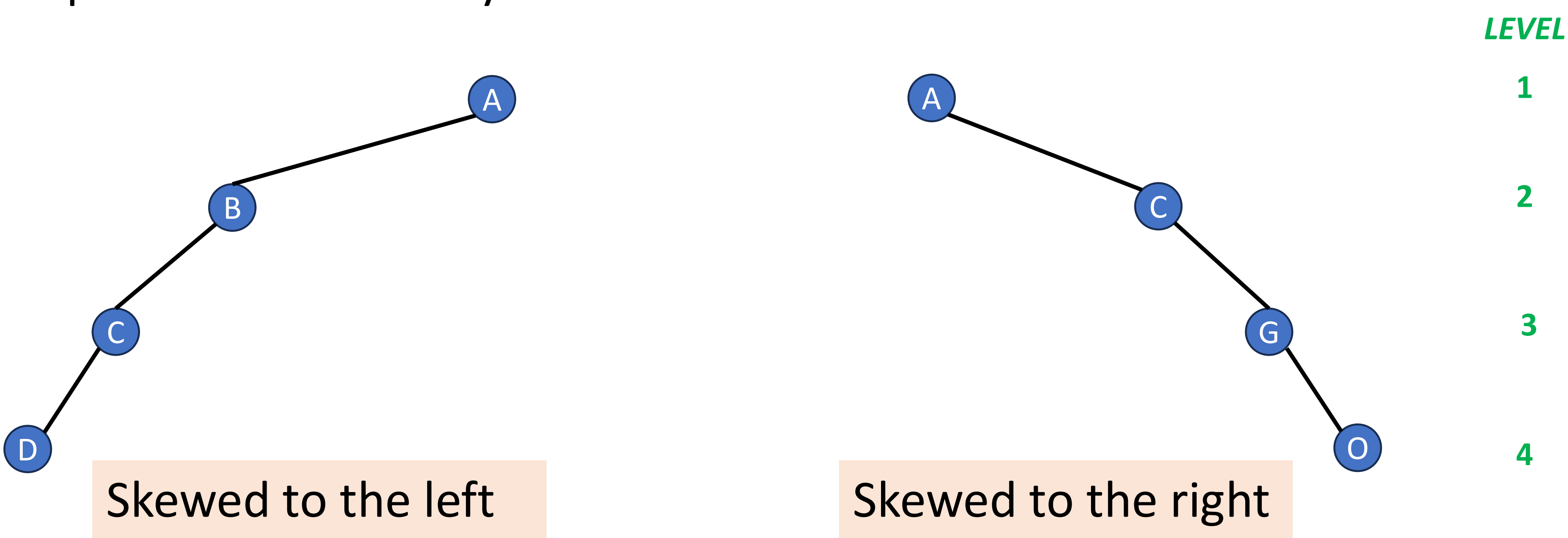
- A full binary tree of a given height h has 2^h-1 nodes.



Height 4 full binary tree.

Minimum number of nodes

- At least one node at each of first h levels
- Minimum number of nodes for a height h tree: h
- Special case of binary tree: skewed tree



Number of nodes and height

- Let n be the number of nodes in a binary tree whose height is h.

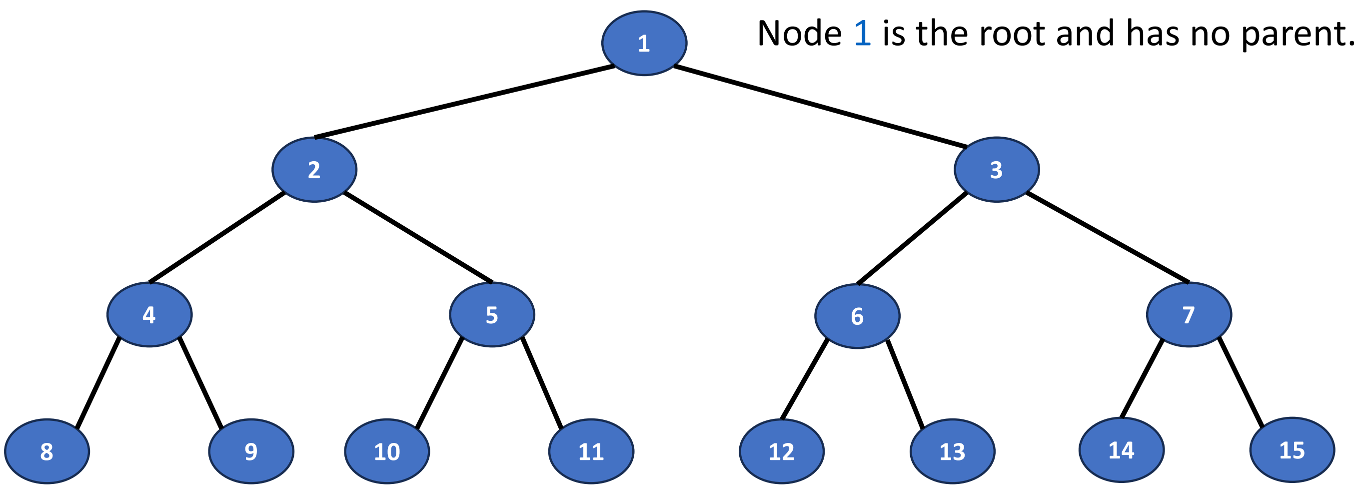
Minimum number of nodes $$h \le n \le 2^h-1$$ *Maximum number of nodes*

$$Log_2(n+1) \le h$$

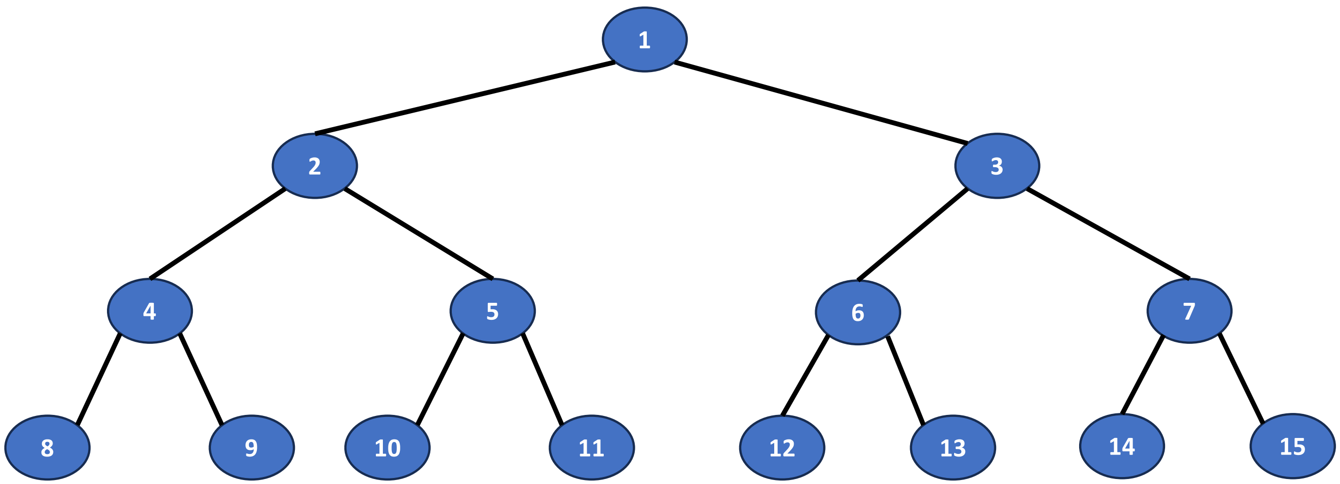
The height h of a binary tree is at least $log_2(n+1)$.

Numbering nodes in a full binary tree

- Sequentially numbering the nodes 1 through 2^h-1.
  - Number by levels from top to bottom.
  - Within a level number from left to right.



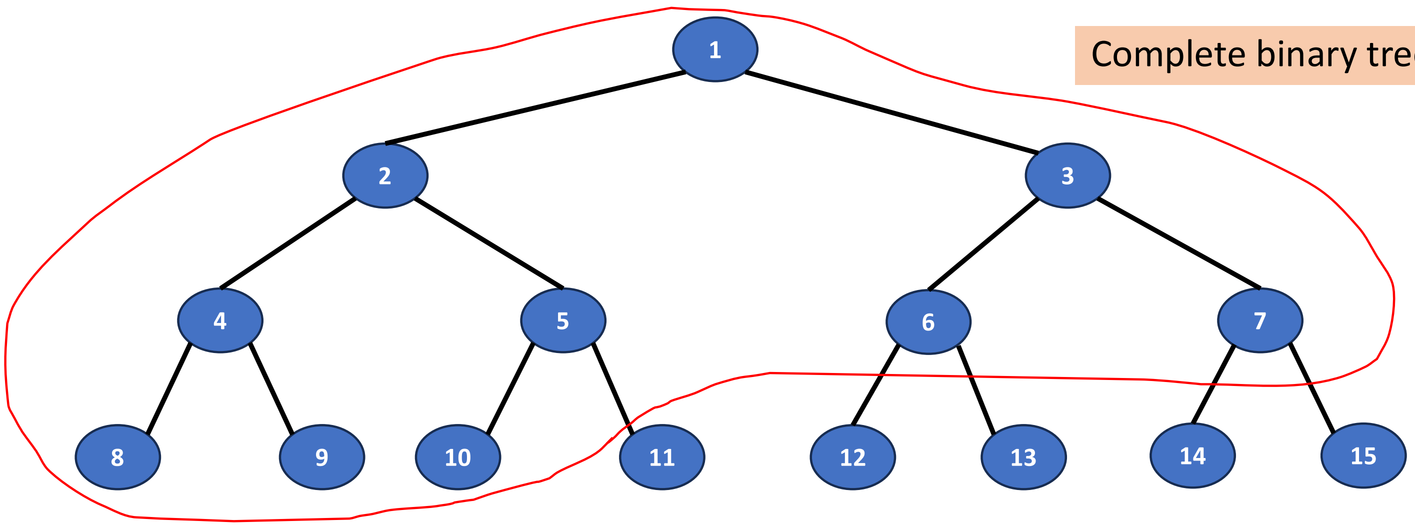
Node number properties



- Let *n* be the number of nodes in a full binary tree
  - Parent of node *i* is node floor(i / 2)
  - Left child of node *i* is node 2i
  - Right child of node *i* is node 2i+1

Complete binary tree with *n* nodes

- Steps:
    1. Create a full binary tree has at least *n* nodes.
    2. Number the nodes sequentially.
    3. The binary tree defined by the node numbered *1* through *n* is the *n*-node complete binary tree.
- Full binary tree is a special case of complete binary tree.

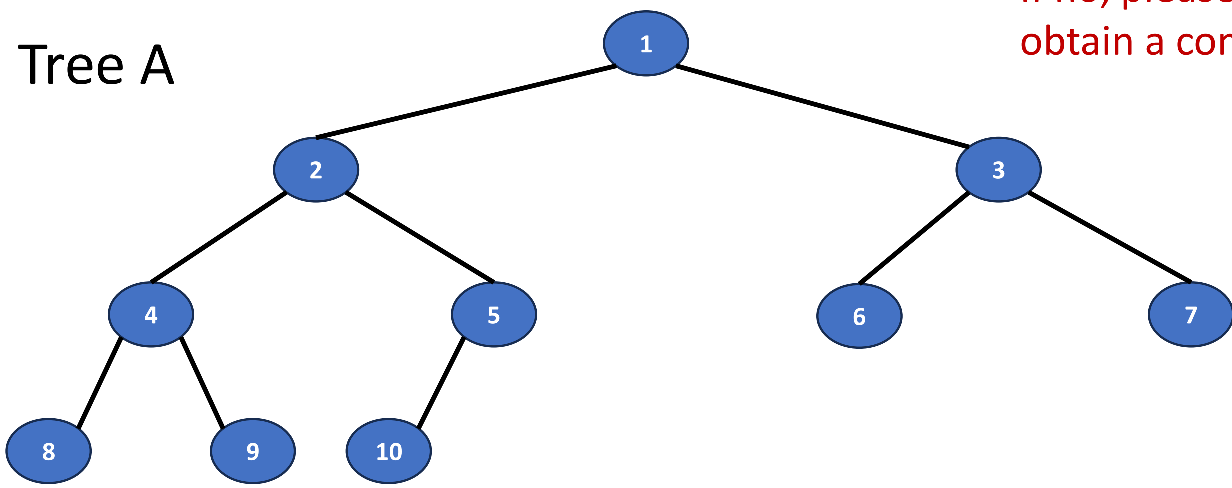


Complete binary tree with 10 nodes.

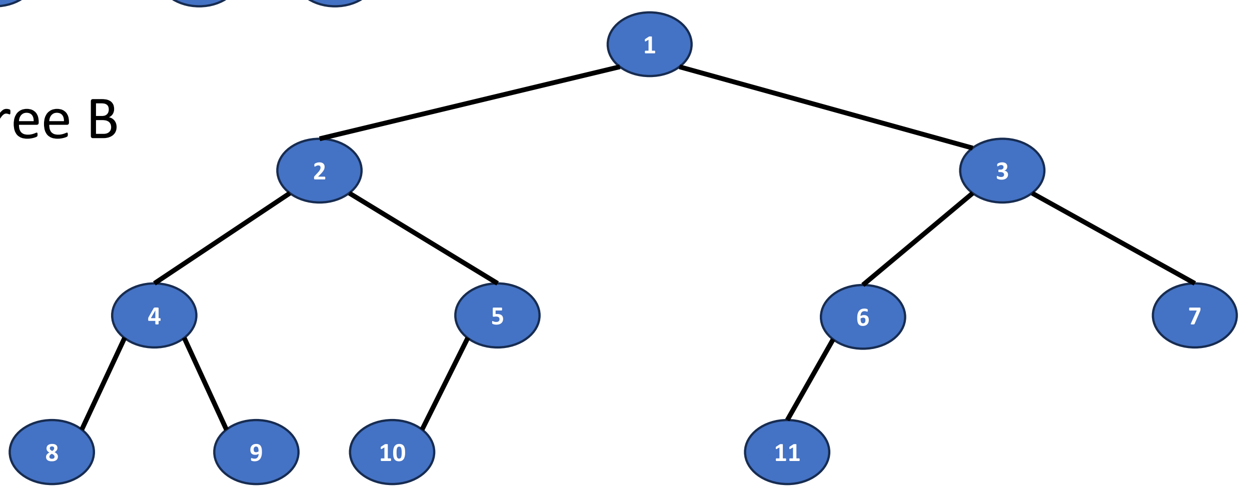
Q12: Are Tree A and Tree B complete binary trees?

If no, please describe how to modify the tree to obtain a complete binary tree.

- Tree A



- Tree B



Please reply your answers of Q12 via the following link:

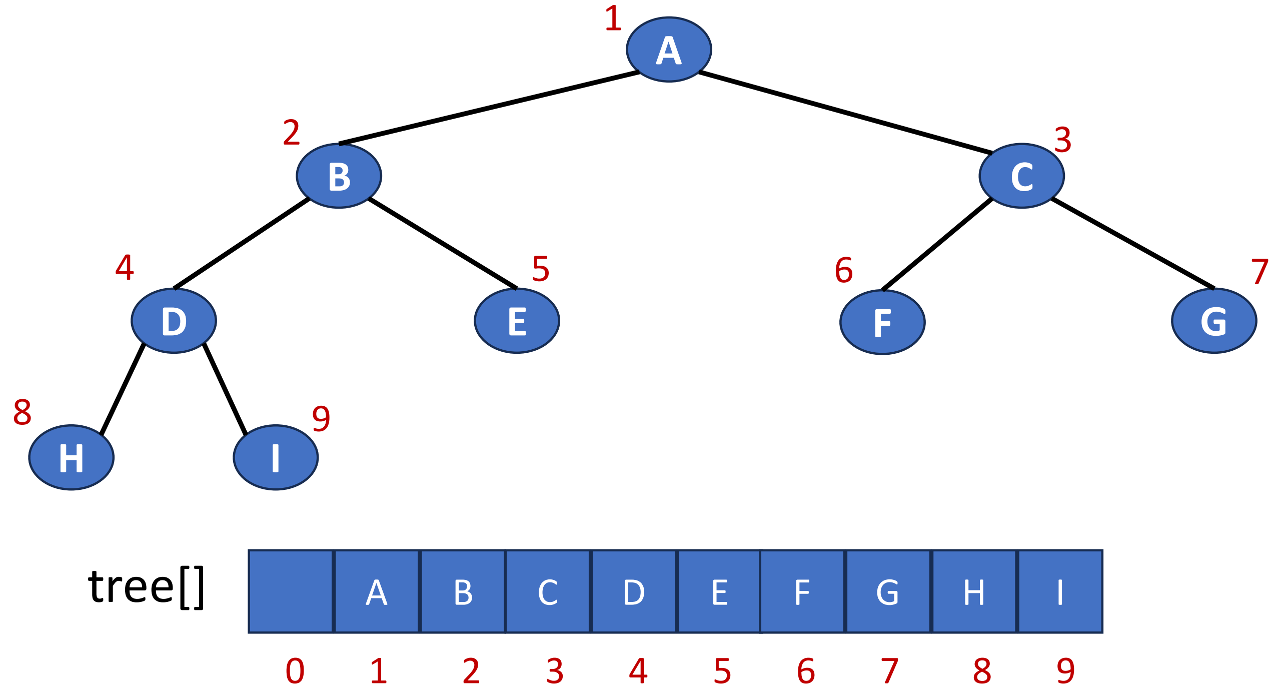
Group members: 1~3 people

Binary tree representations

- Array representation
- Linked representation

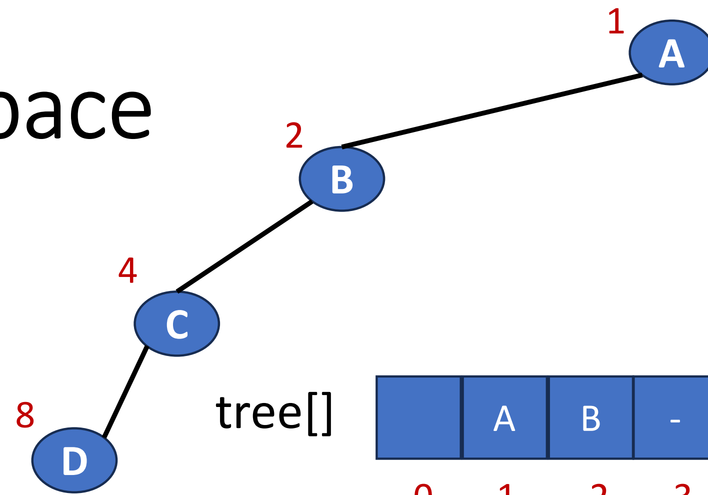
Array representation

- Number the nodes using the numbering scheme for a full binary tree. The node that is numbered *i* is stored in tree[*i*].



May have unutilized space

- Left-skewed binary tree

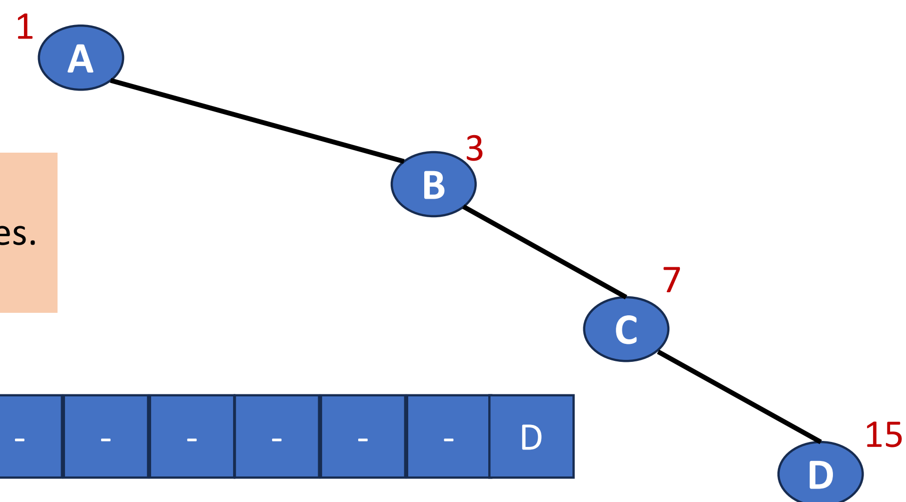


tree[]

	A	B	-	C	-	-	-	D
0	1	2	3	4	5	6	7	8

- Right-skewed binary tree

Worst case
A skewed tree of depth *k* requires 2^k-1 spaces.
But only *k* will be used.



	A	-	B	-	-	-	C	-	-	-	-	-	-	-	D
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Linked representation

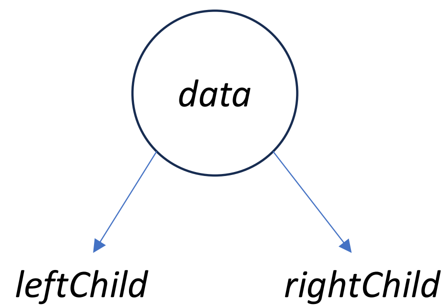
- Each binary tree node is represented as an object whose data type is TreeNode.

- The space required by an n node binary tree is
  n * (space required by one node).

Node representations

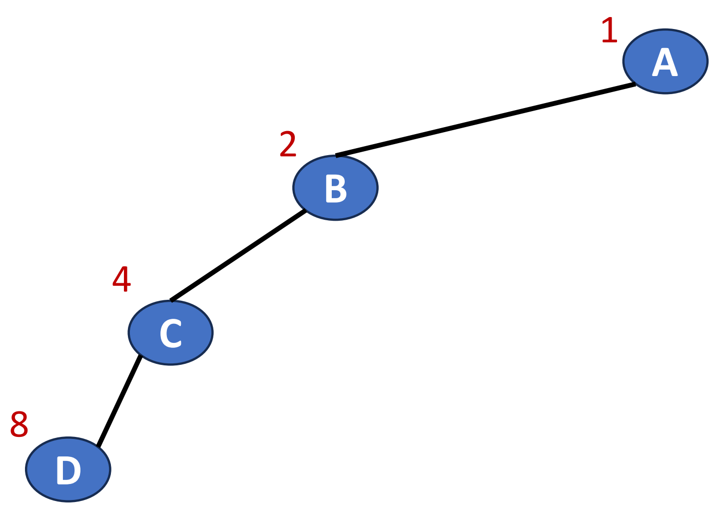
```c
typedef struct node *treePointer;
typedef struct node{
        char data;
        treePointer leftChild, rightChild;
        };
```

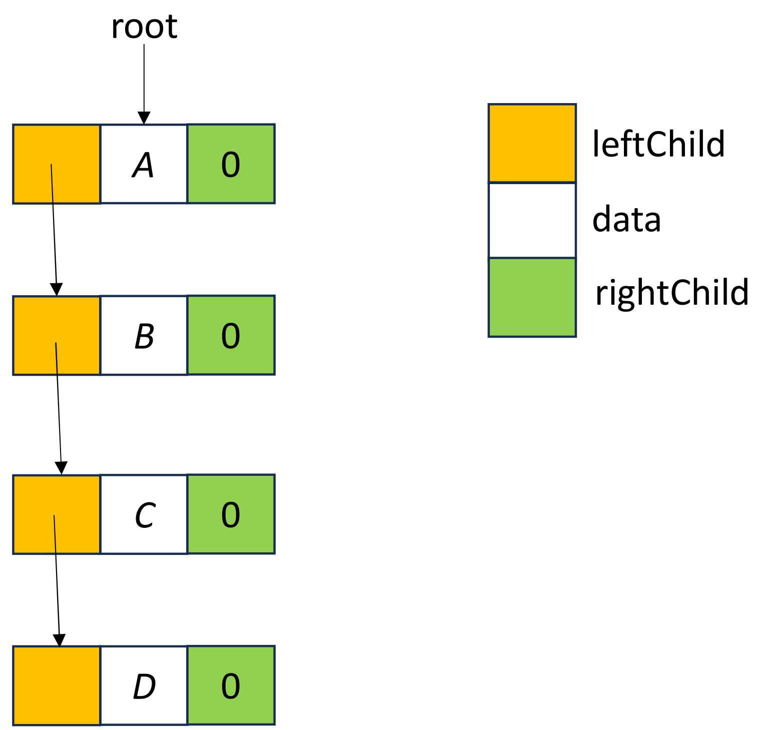
leftChild	*data*	*rightChild*



Linked representation for binary tree

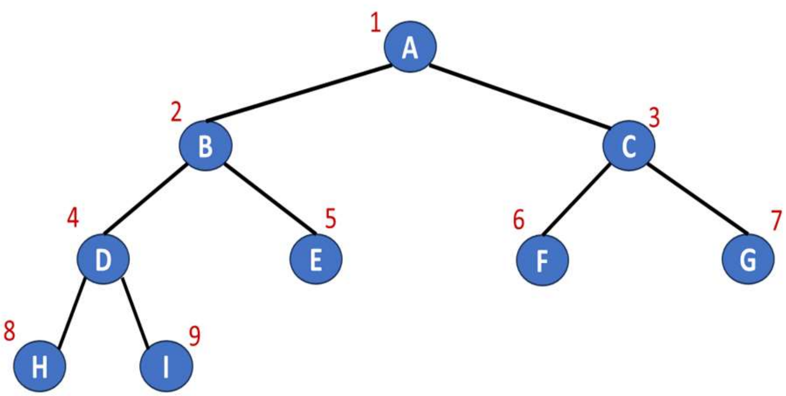
- Skewed binary tree

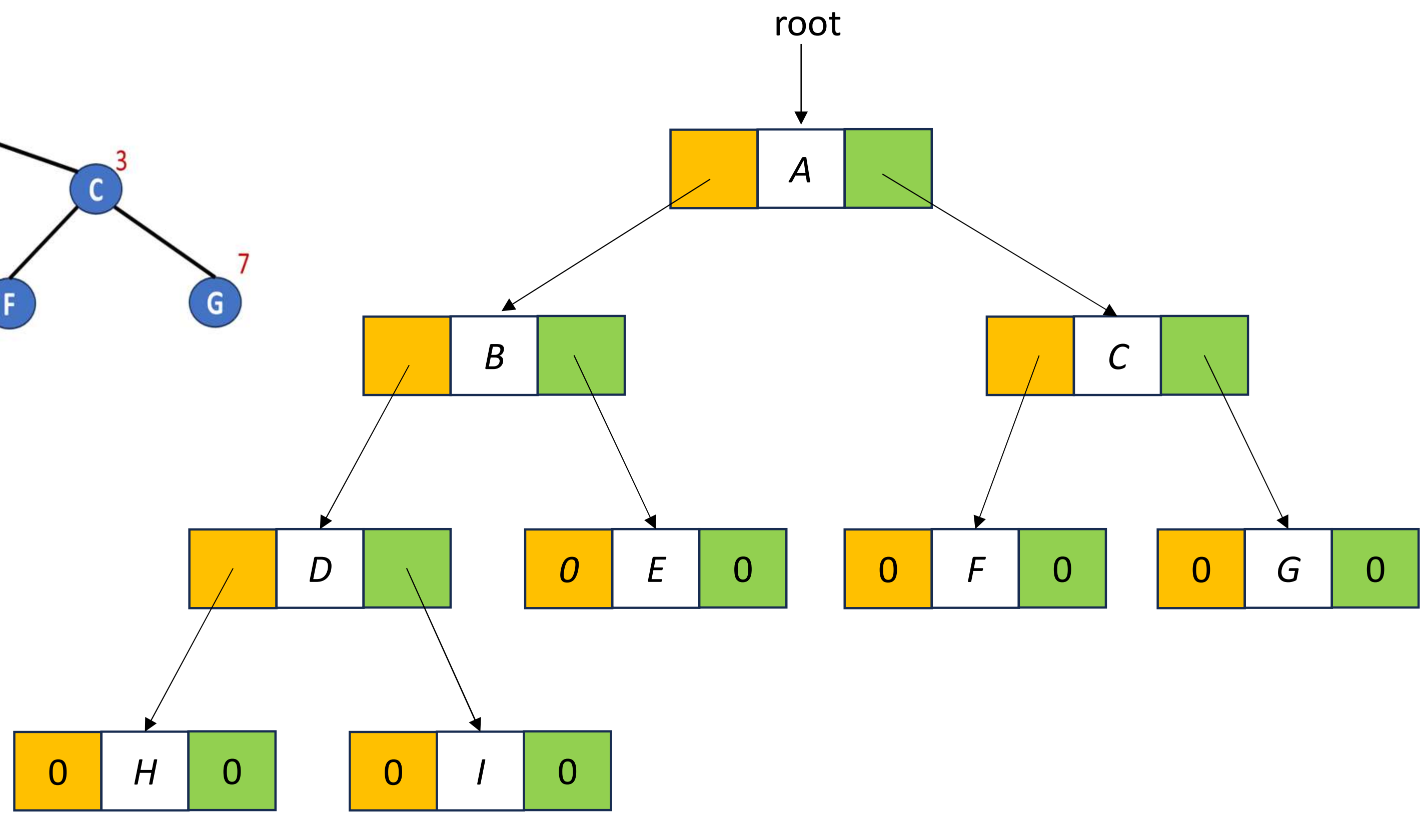




Linked representation for binary tree

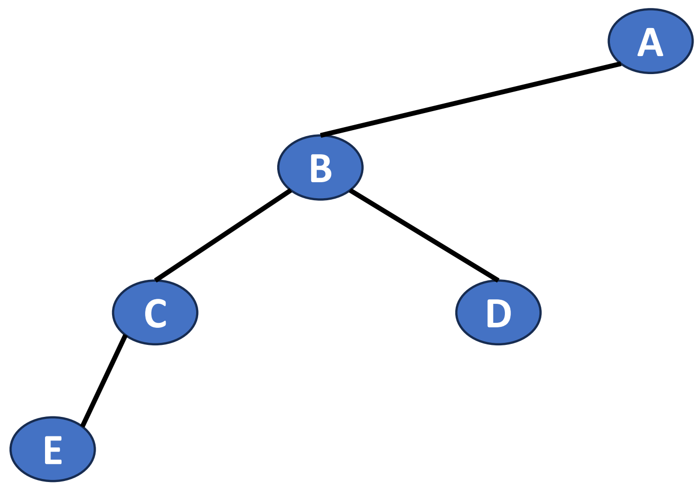
- Complete tree





Exercise

- Draw the internal memory representation of the binary tree
  1. Using sequential (array) representation
  2. Using linked representation



Exercise

- Draw the internal memory representation of the binary tree
  1. Using sequential (array) representation
  2. Using linked representation