Data Structure

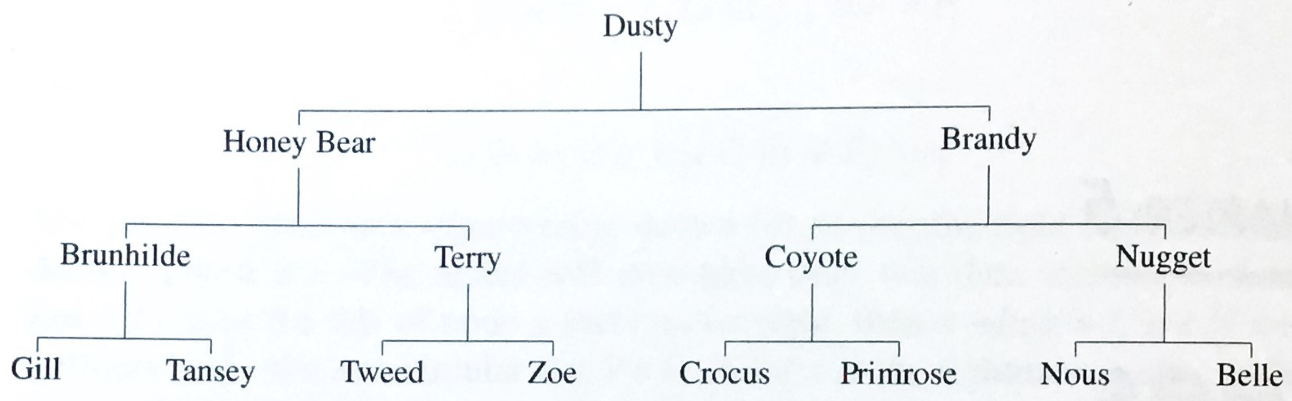
Trees

Shin Hong

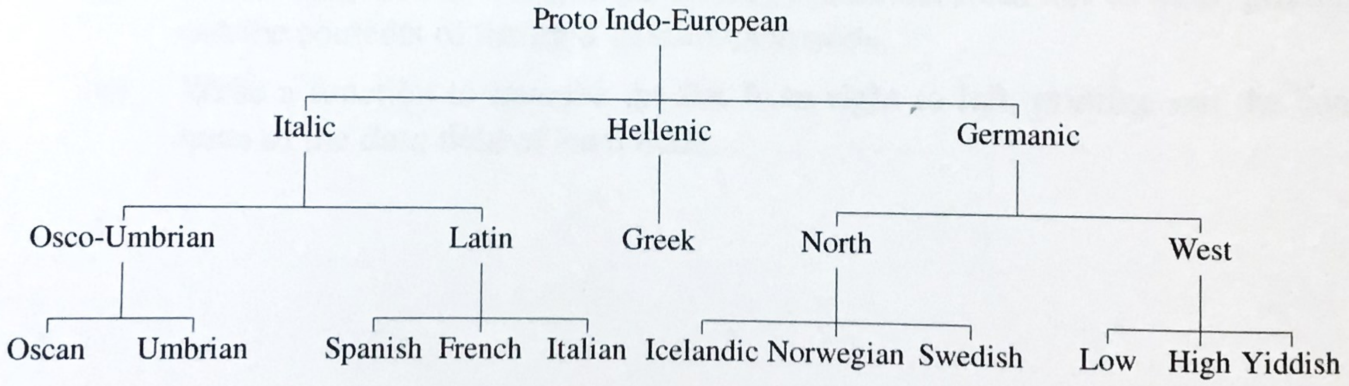
28 Apr 2023

DS&A. Chapter 7. Trees

Motivation



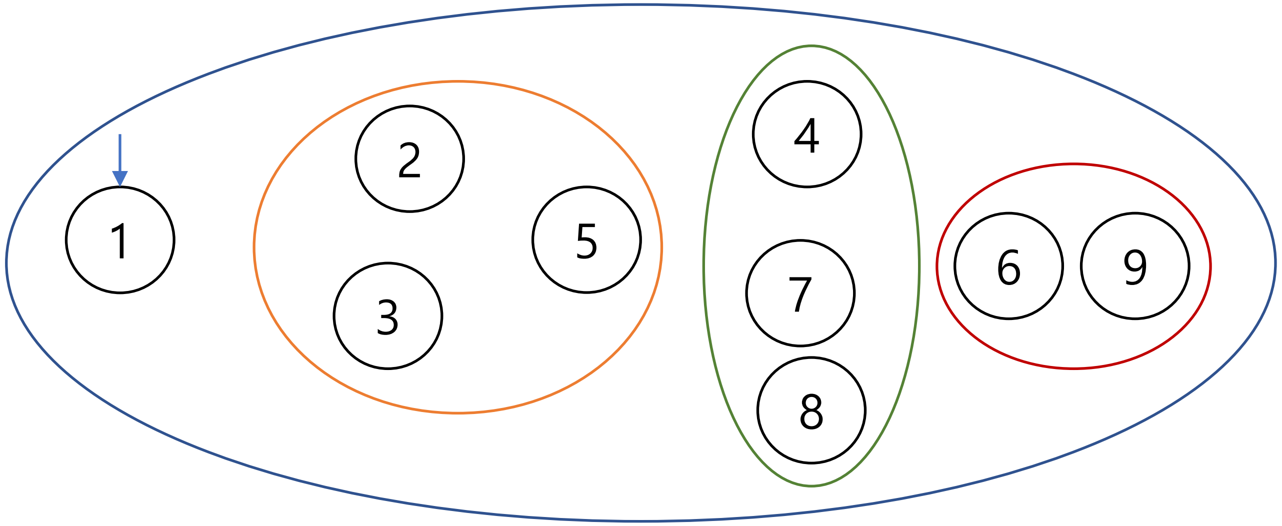
(a) Pedigree



(b) Lineal

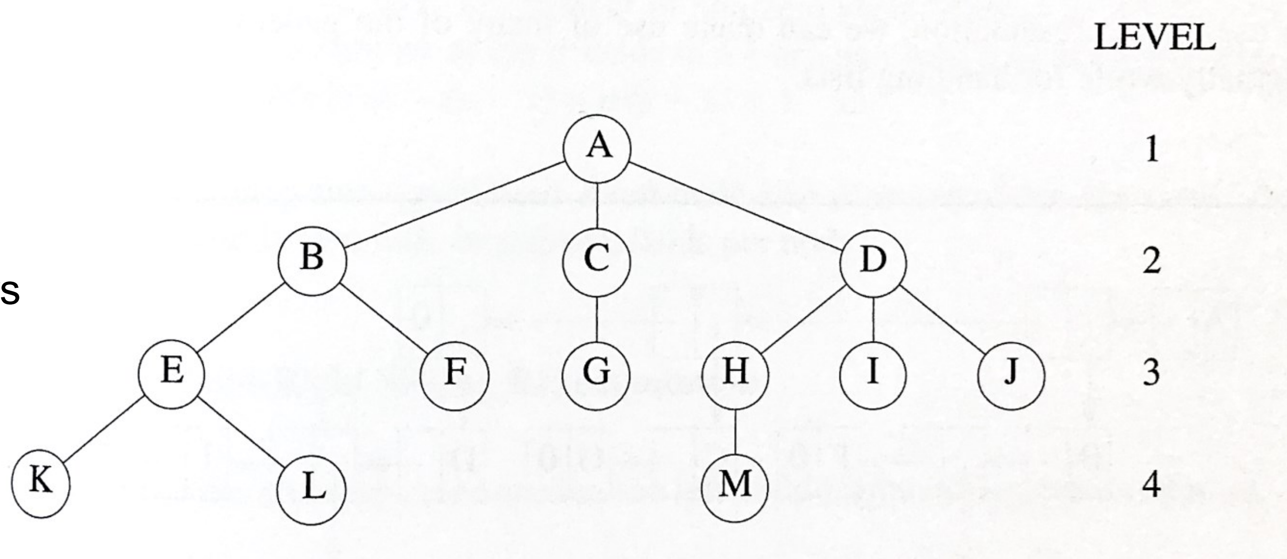
Tree

- A tree is a finite set of one or more nodes such that:
  - there exists a specifically designated node called the *root*, and
  - the remaining nodes are partitioned into disjoint sets T_1, T_2, ⋯, T_n, where each of these sets is a tree (subtree)



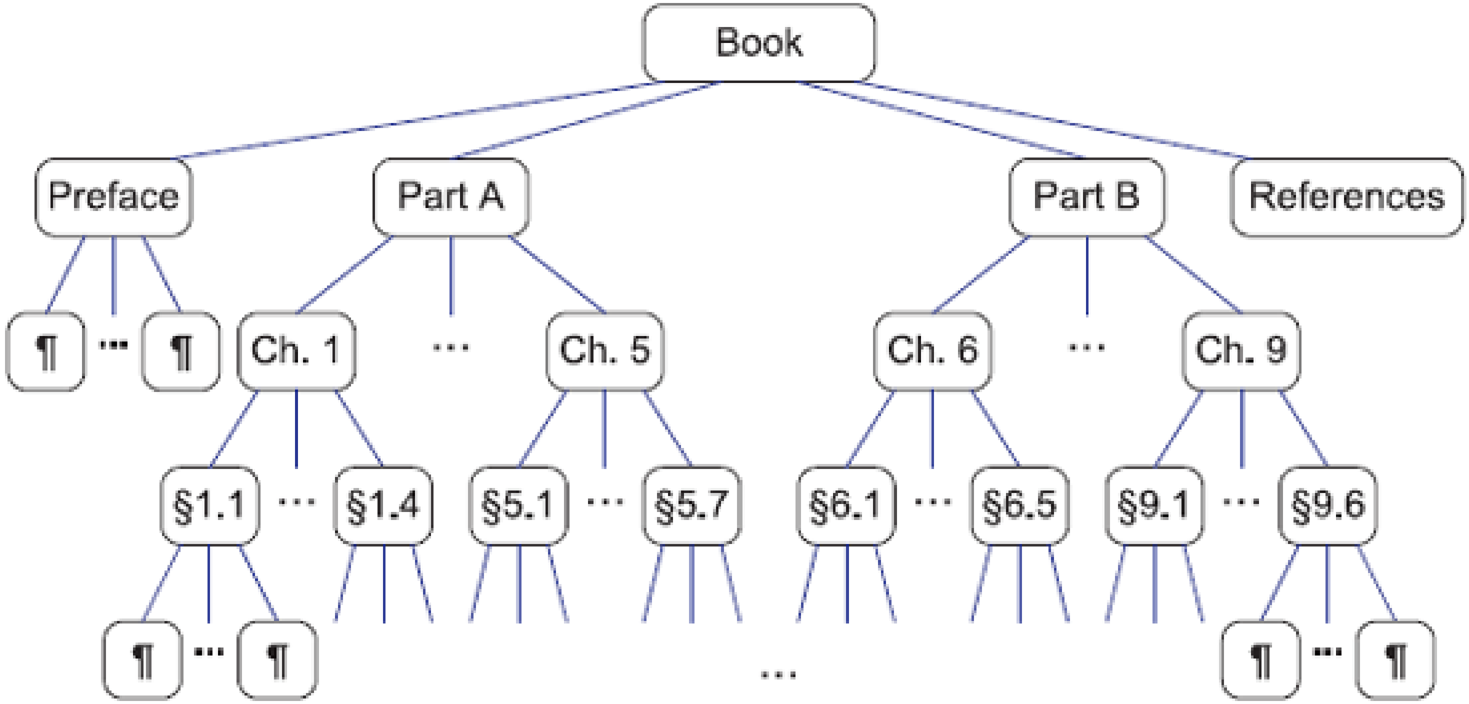
Terminologies

- Node: the item of information
- Branch (edge): links between two nodes (a parent and a child)
- Degree of a node: the number of subtrees
  - Degree of a tree
- Leaf (terminal, external) node: node with degree zero
  - non-terminal (internal) nodes
- Children, Parent, Siblings, Ancestors
- Level of a node
  - depth of a node is the number of the ancestors
- Height of a tree



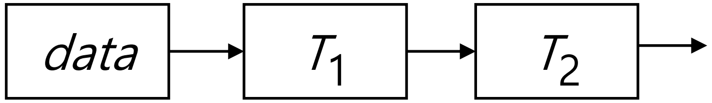
Ordered Tree

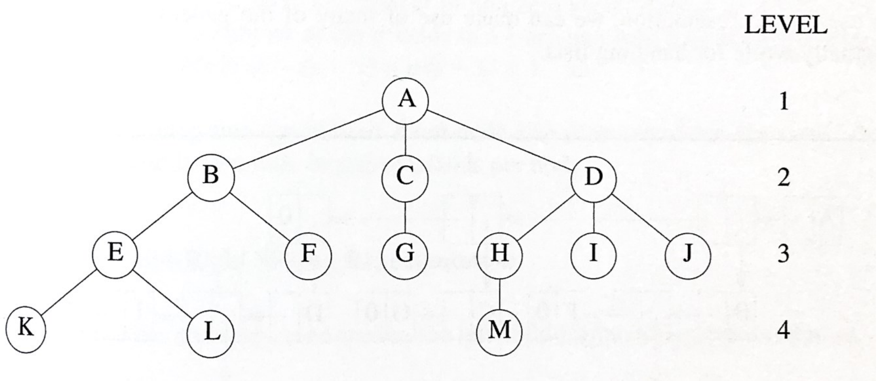
- A tree is ordered if there is a linear ordering defined for children of each node
  - an ordering determines how the tree is used

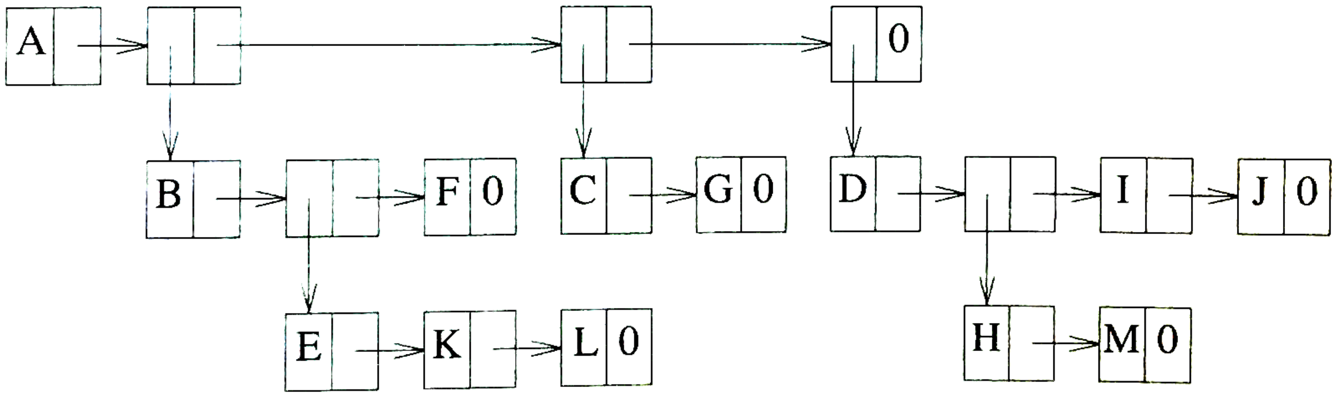


Tree Representation

- List representation
  - *Data, or* (*Data* (T_1, T_2, ... , T_N))
  - E.g., $(A\,(B\,(E\,(K,L),F),C\,(G),D\,(H\,(M),I,J)))$



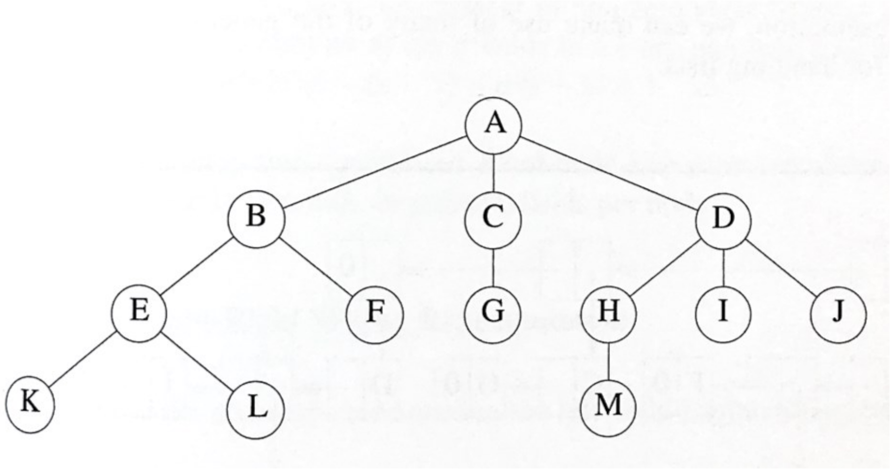


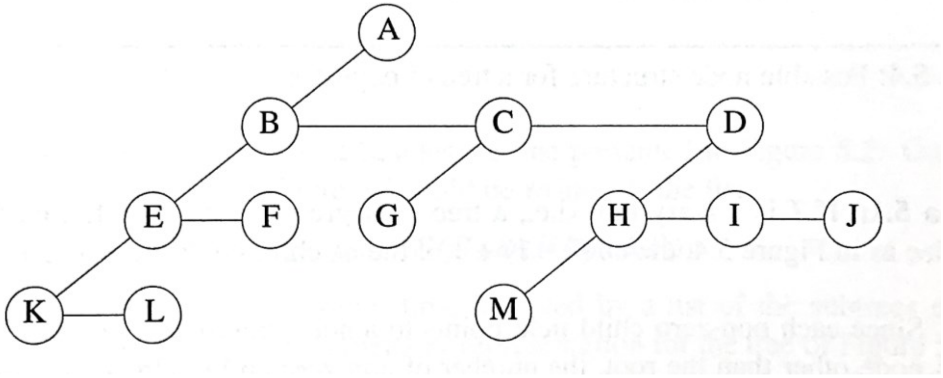


Tree Representation

- Left child-right sibling representation

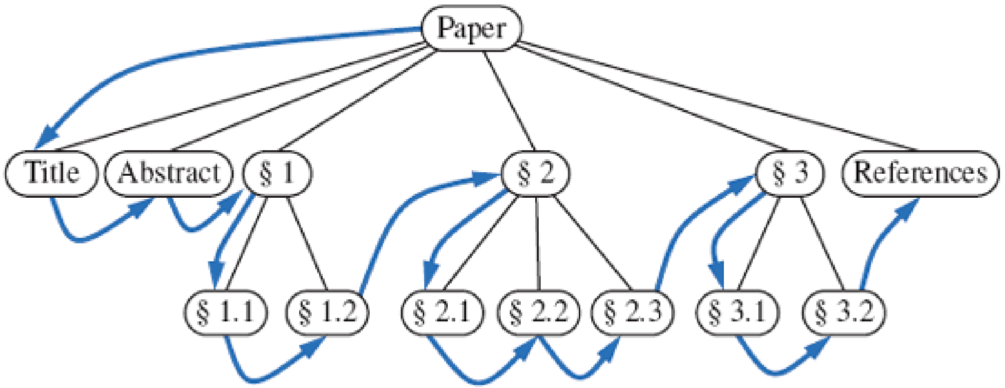
data	
child	*sibling*

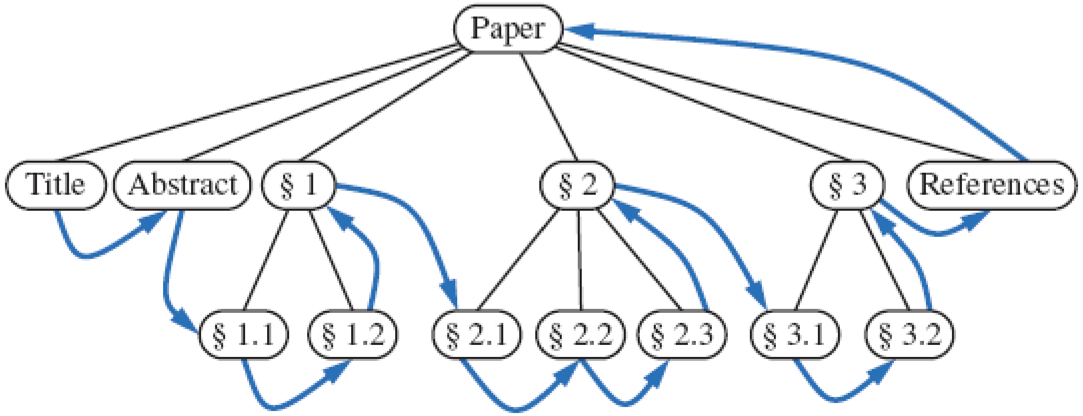




Tree Traversal

- A traversal of a tree is a systematic way of accessing (visiting) all nodes
- preorder traversal: visit the root node first, and then visit the sub-trees recursively
- postorder traversal: recursively visit the sub-tree first, and then visit the root node





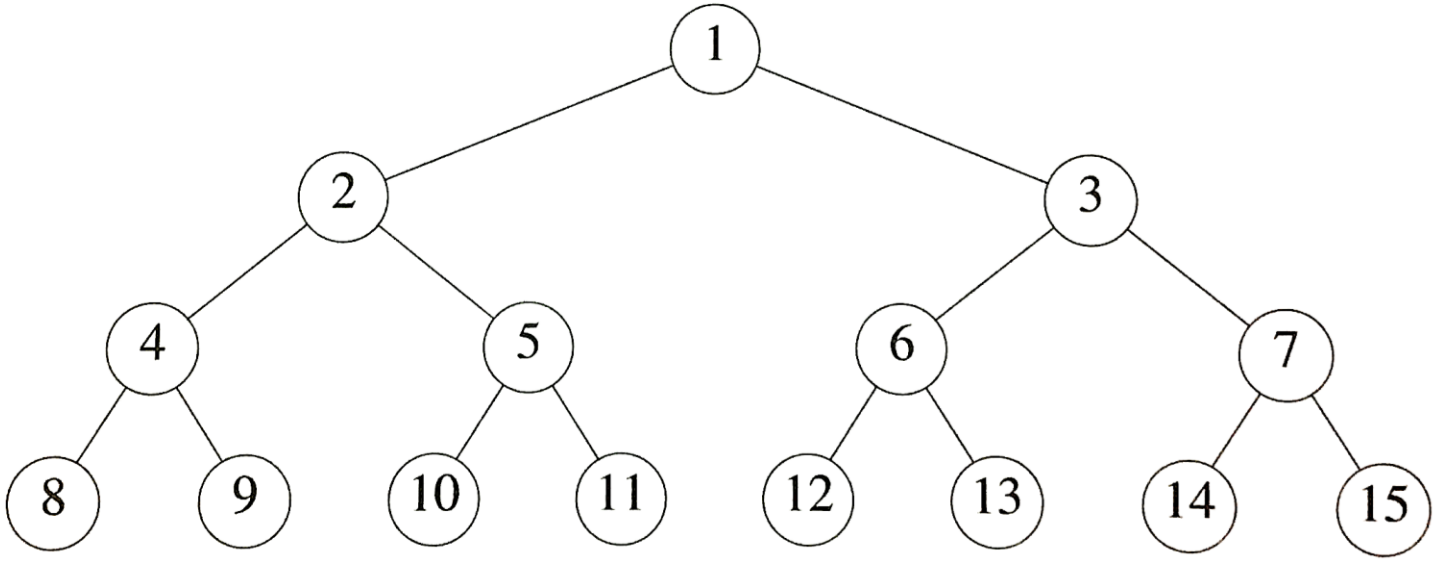
Binary Tree

- A binary tree is a finite set of nodes that is either empty or consists of a root and two disjoint binary trees
  - A binary tree is a tree with a degree 2
  - each node may have a left child and a right child

- The definition of binary tree differs from the standard notion of a tree
  - no tree with zero node, but there’s an empty binary tree
  - no ordering in children in a tree, but a binary has

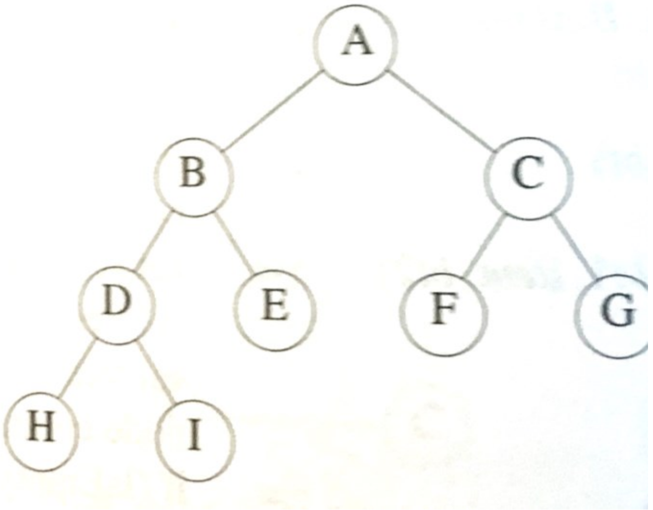
Terminologies (1/2)

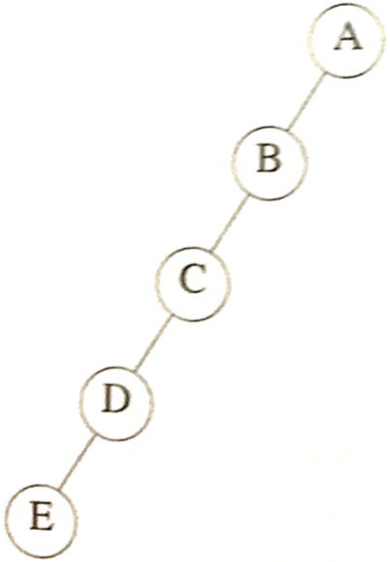
- A **full binary tree** of depth k is a binary tree of depth k having $2^k - 1$ nodes



Terminologies (2/2)

- A binary tree with n nodes and depth k is **complete** if and only if its nodes correspond to the nodes numbered from 1 to n in the full binary tree of depth k
- The hiehgt of a complete binary tree with n nodes is $\lceil \log_2(n+1) \rceil$
- A tree is called skewed if nodes are skewed at left or right subtrees

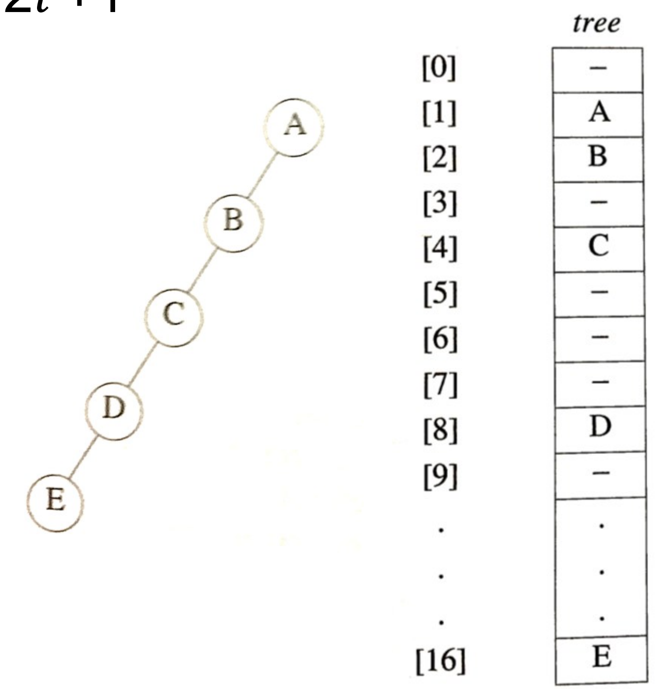




Representation: with array

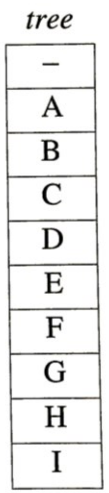
- If a complete binary with n nodes is represented sequentially, then for any n ode with index i, the following properties hold
  1. parent(i) is at index floor $\lfloor i/2 \rfloor$ except the root node (i is 1)
  2. left_child(i) is at 2i
  3. right_child(i) is at 2i +1

⌋ except the root node (i is 1)



	tree
[0]	–
[1]	A
[2]	B
[3]	–
[4]	C
[5]	–
[6]	–
[7]	–
[8]	D
[9]	–
.	.
.	.
.	.
[16]	E

(a) Tree of Figure 5.10(a)

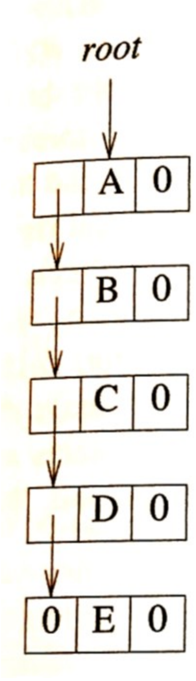


tree
–
A
B
C
D
E
F
G
H
I

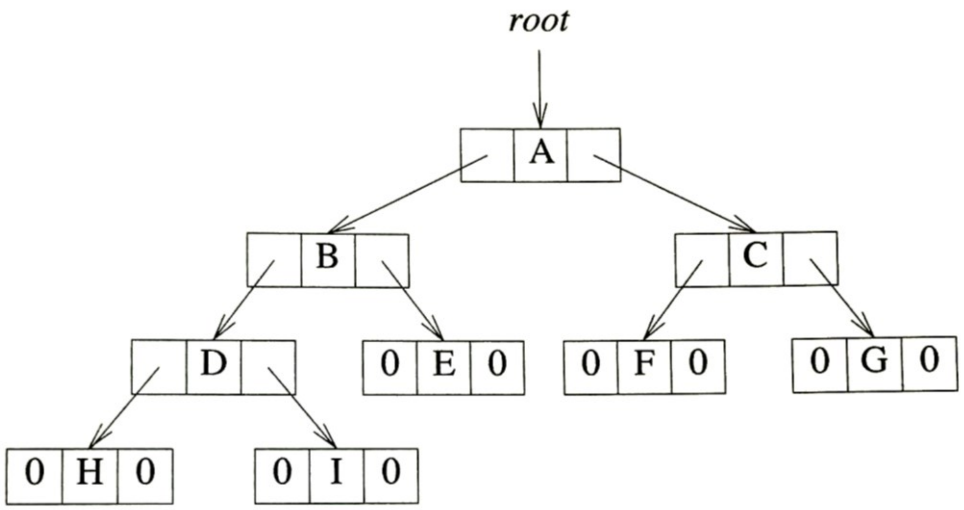
(b) Tree of Figure 5.10(b)

Representation: linked list

```
struct tree {
   int data ;
   struct tree * left ;
   struct tree * right ;
}
```



root
A 0
B 0
C 0
D 0
0 E 0



root
A
B
C
D
0 E 0
0 F 0
0 G 0
0 H 0
0 I 0