

EECS 204002
Data Structures 資料結構
Prof. REN-SONG TSAY 蔡仁松 教授
NTHU

CH. 5 TREES

2018/9/16 © Ren-Song Tsay, NTHU, Taiwan 1

5.1

Introduction

2018/9/16 © Ren-Song Tsay, NTHU, Taiwan 2

Tree Structure

- Data in a tree structure are organized in a **hierarchical** manner.

```

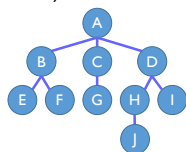
graph TD
    Dusty --> HoneyBear[Honey Bear]
    Dusty --> Brandy
    HoneyBear --> Brunnhilde
    HoneyBear --> Terry
    Brunnhilde --> Gill
    Brunnhilde --> Tansey
    Terry --> Tweed
    Terry --> Zoe
    Brandy --> Coyote
    Brandy --> Nugget
    Coyote --> Crocus
    Coyote --> Primrose
    Nugget --> Nous
    Nugget --> Belle
  
```

Pedigree Chart

Tree Definition

A tree is a finite set of one or more nodes

- There is one root
- The remaining nodes can be partitioned into n disjoint sets $T_1, T_2, \dots, T_n (n \geq 0)$
- Each subset T_i is a tree (also called subtrees of the root)

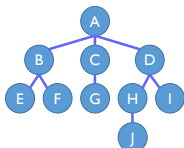


4

5.1.1

Terminology

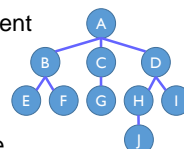
- **Degree of a node**
 - The number of subtrees
 - Eg : $\text{deg}(A) = 3; \text{deg}(C) = 1$
- **Leaf or Terminal nodes**
 - The node whose degree is 0
 - Eg : $E \setminus F \setminus G \setminus J \setminus I$
- **Nonterminals**
- **Degree of a tree**
 - The maximum degree of the nodes in the tree
 - Eg : Deg. of the tree = 3



5

Terminology

- **Parent / Children**
- **Sibling**
 - Children of the same parent
 - Eg : $E \setminus F$ are siblings
- **Ancestors**
 - All nodes along the path from the root to that node
 - Eg : ancestor of $J \Rightarrow H \setminus D \setminus A$
- **Descendants**
 - All nodes in the subtrees



6

Terminology

- **Level of a node**
 - Level(root) = 1
 - Level(node) = $\ell + 1$ if level of n's parent is ℓ
 - Ex : level(G) = 3
- **Height or depth of a tree**
 - Maximum level of any node in the tree
 - Eg : Height of the tree = 4

level

... 1

... 2

... 3

... 4

7

5.1.2

List Representation

- Each tree node holds a **data field** and **several link fields** pointing to subtrees
- However, the degree of each node might vary. For a tree of **degree k** , allocate k link fields for each node.

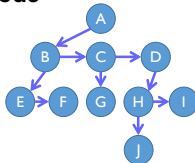
8

List Representation

- Disadvantage: Waste memory space!
 - If T is a tree of degree k with n nodes.
 - The total # of link fields are $n \cdot k$
 - The total # of used link fields are $n - 1$
 - For each node (except **root**), there is only one link pointing to it.
 - The # of zero link fields are $n \cdot k - (n - 1)$

Left Child-Right Sibling Representation

- Each node has exactly **two link fields**
 - Left link (child): points to **leftmost child node**
 - Right link (sibling): points to **closest sibling node**



Left Child-Right Sibling Representation

- Rotate clockwise **45°** Degree-2 tree
Binary Tree

