

Lecture 13 - February 25

General Trees

Linear vs. Non-Linear Structures

General Trees: Terminology

Generic TreeNode in Java

Announcements/Reminders

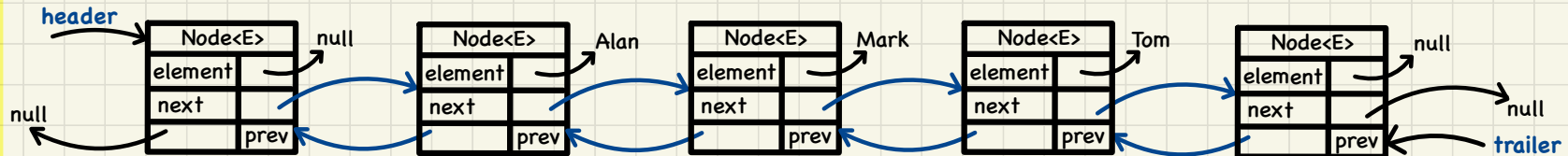
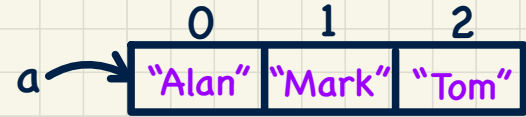
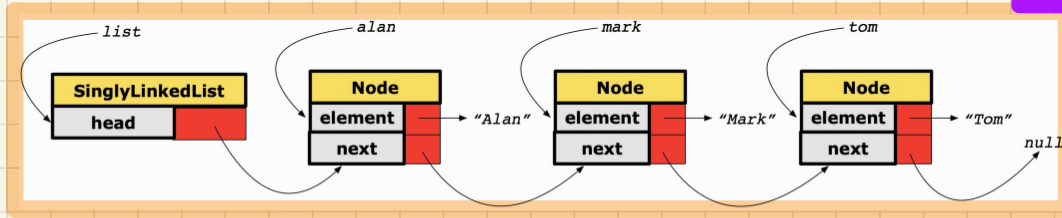
- Survey on **Makeup Lecture** for ProgTest1
- **Assignment 3** (on linked Trees) to be released
- ~~**ProgTest2** (on linked Trees) to be released~~
- This week's office hour: 3pm, Wed

WT

Running Time: Arrays vs. SLL vs. DLL

see earlier discussion

DATA STRUCTURE	ARRAY	SINGLY-LINKED LIST	DOUBLY-LINKED LIST
OPERATION			
size			
first/last element			$O(1)$
element at index i	$O(1)$	$O(n)$	$O(n)$
remove last element		$O(1)$	$O(1)$
add/remove first element, add last element			$O(1)$
add/remove i^{th} element	given reference to $(i-1)^{\text{th}}$ element not given	$O(n)$	$O(1)$

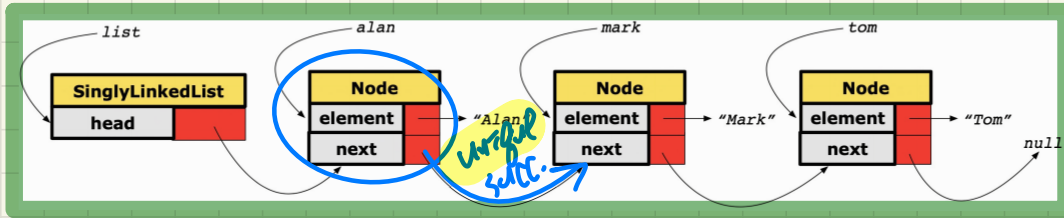


Trees

- a.
 - 1. General Trees
 - 2. Binary Trees (BTs)
- b.
 - 3. Binary Search Trees (BSTs)
- c.
 - 4. Balanced BSTs
 - 5. Priority Queues
 - 6. Heap
 - 7. Heap Sort

Linear vs. Non-Linear Structures

SLLs

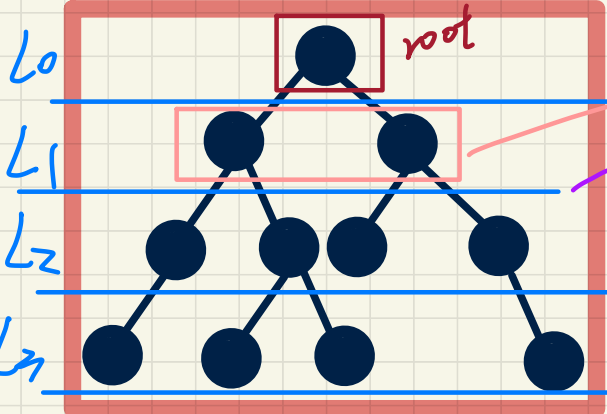
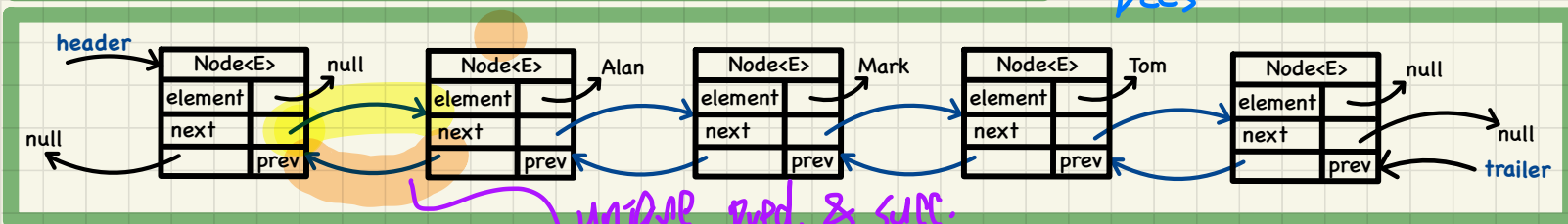


unique prod.
 $i-1$
arrays

unique & ser.
 i
 $i+1$

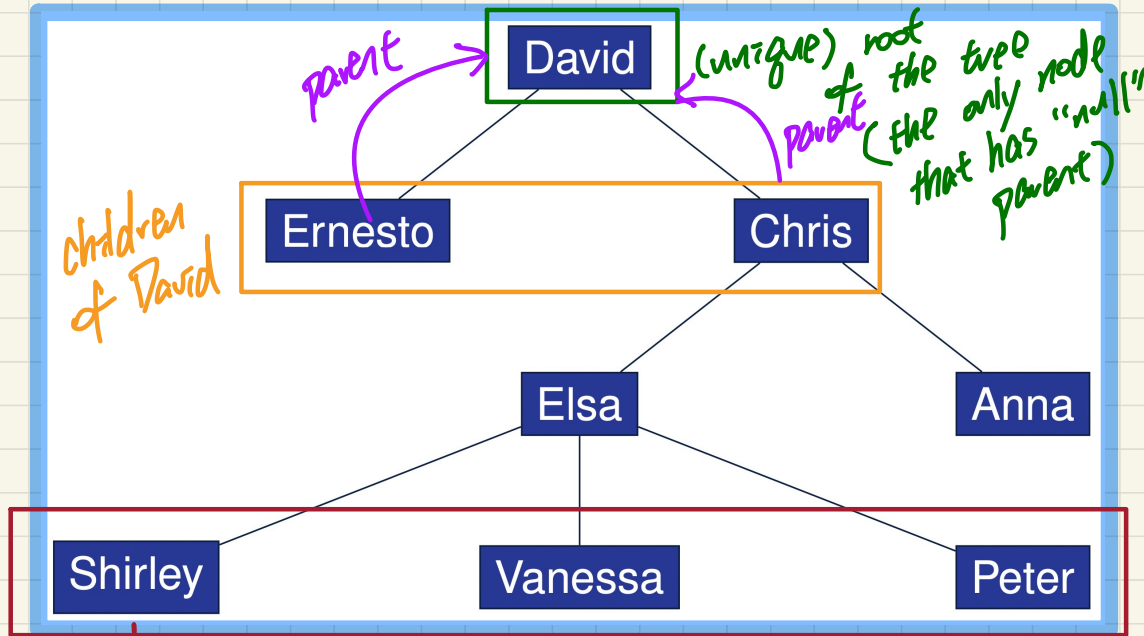


DLLs



multiple "successors" of the root
(which can be stored in a linear DS)
hierarchical structure (levels)

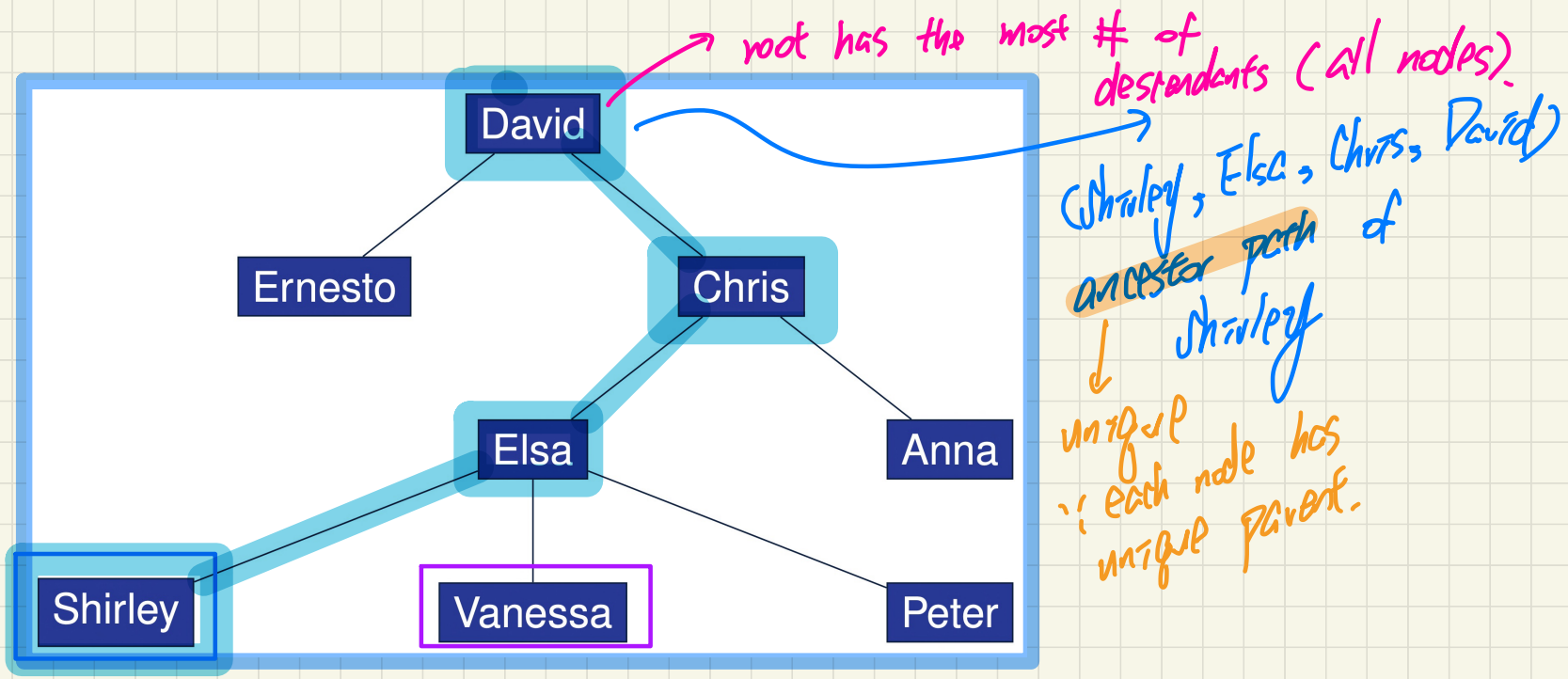
General Trees: Terminology (1)



- root
- parent
- children
- ancestors
- descendants
- siblings

↓
nodes sharing the same parent

↓
siblings (at the same level)

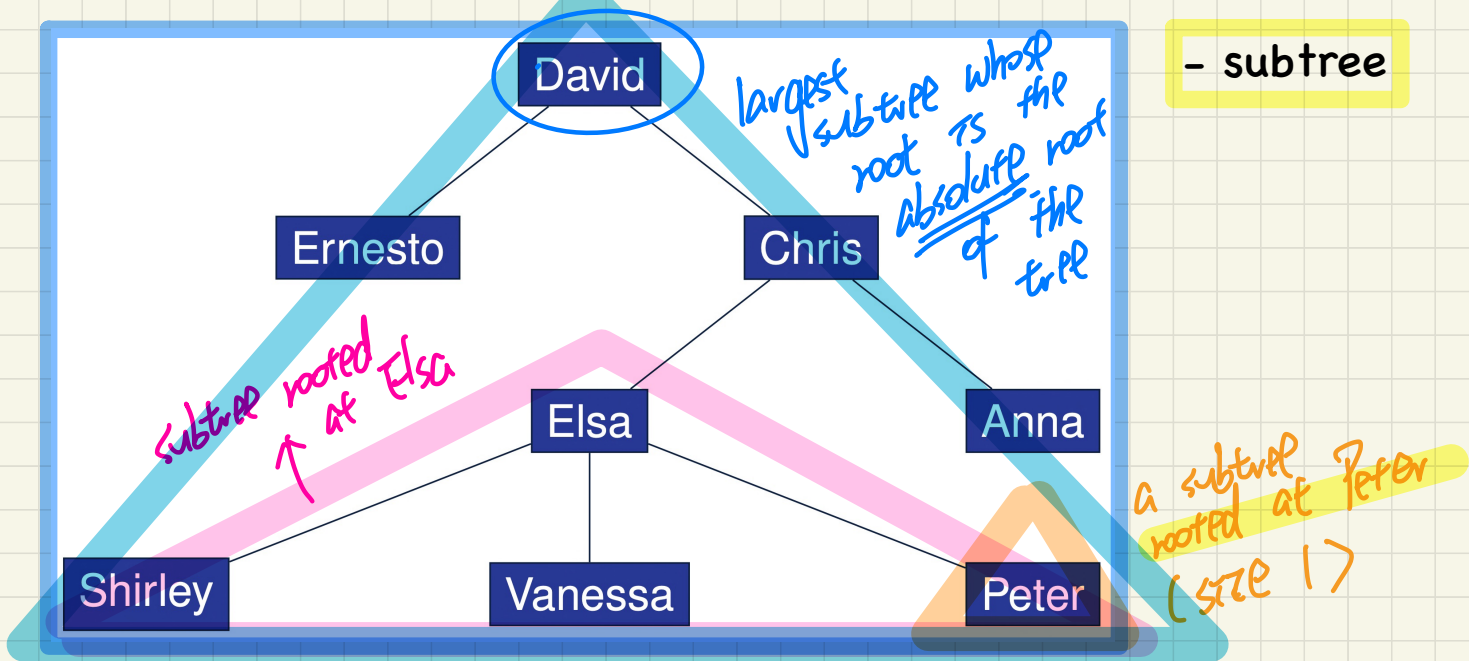


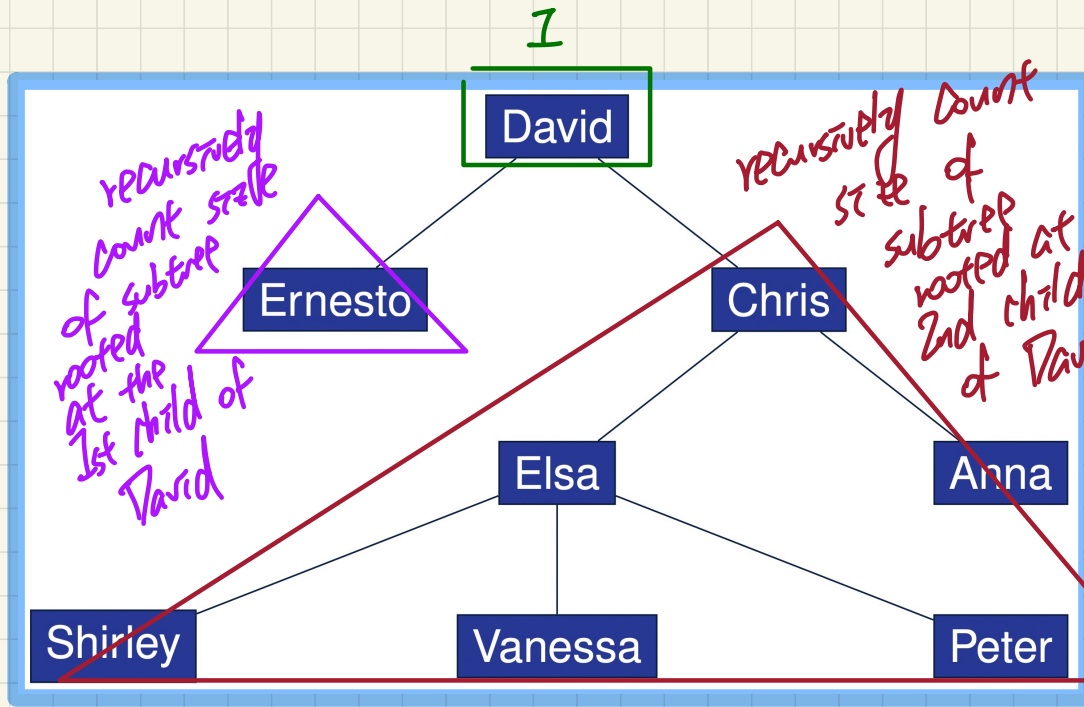
Ancestors of Shirley: S, Elsa, C, D

Descendants of Vanessa: Vanessa

A node is both its ancestor and descendant.

General Trees: Terminology (2)





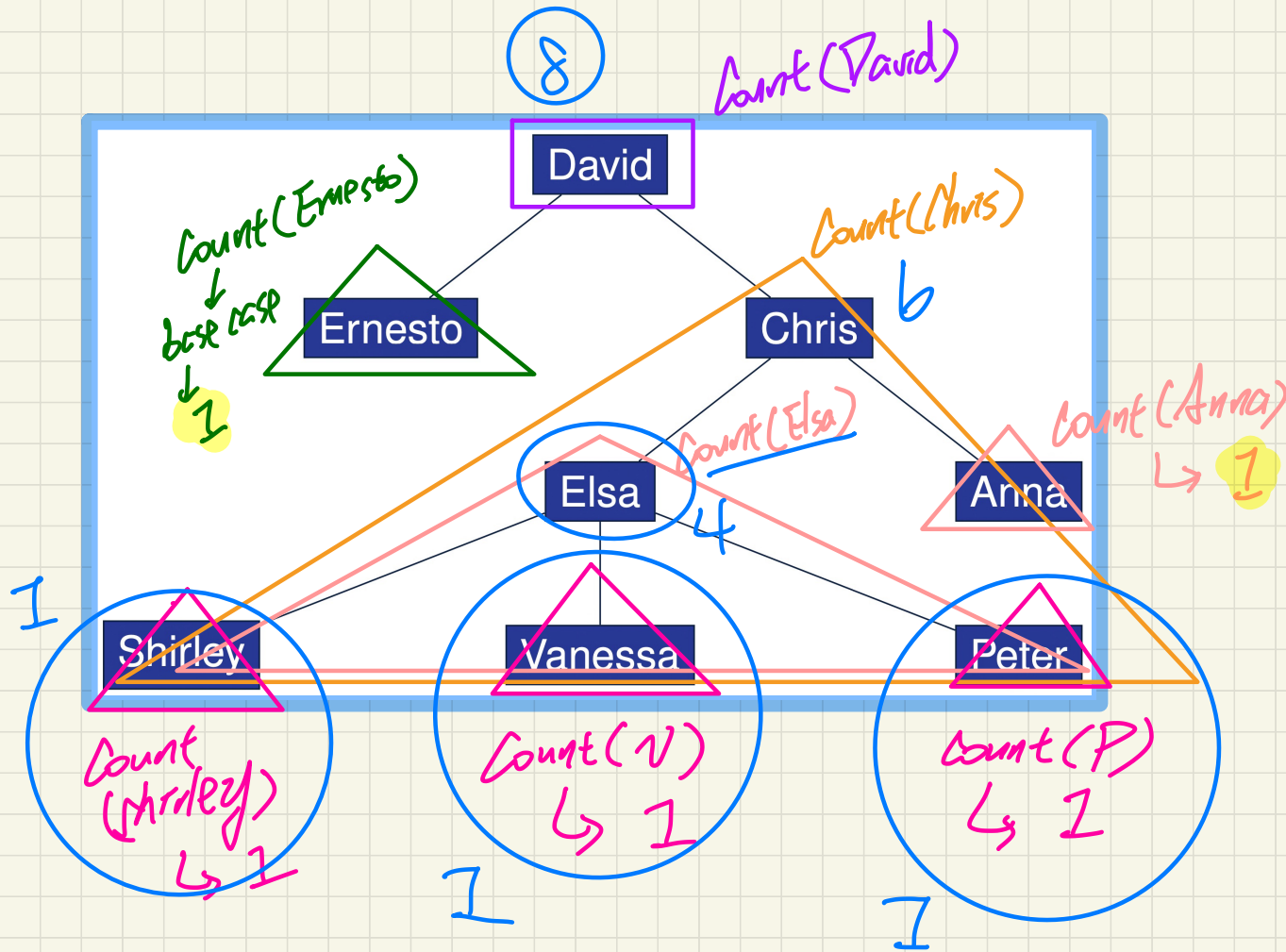
Problem

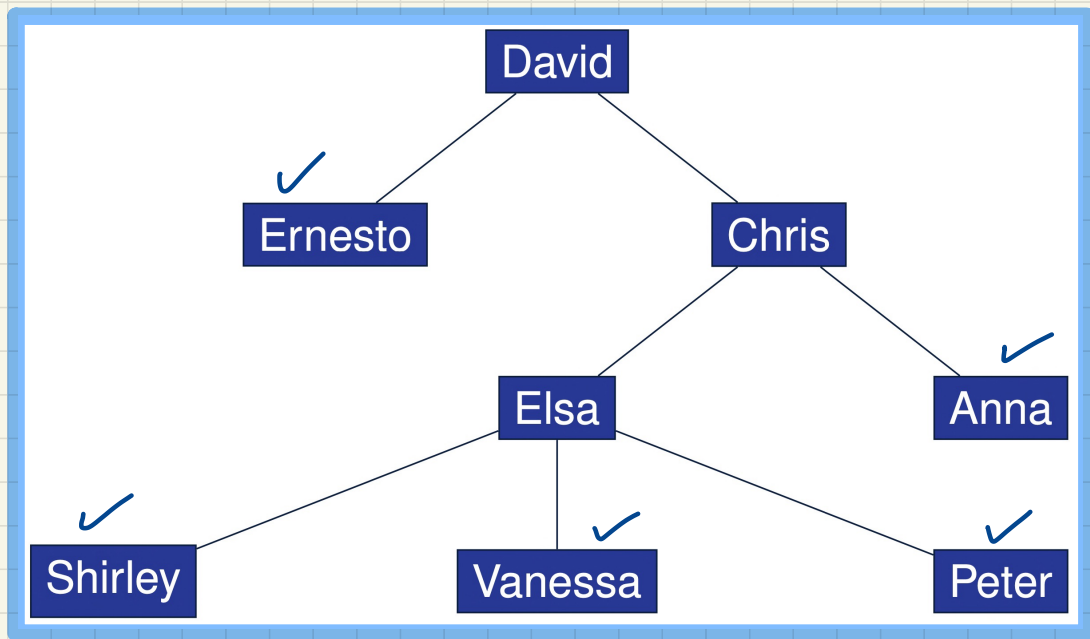
Count the # of nodes in a tree rooted at 1.

Count (David)

↓
root of some subtree.

- subtrees are the subproblems for you to make recursive calls on.



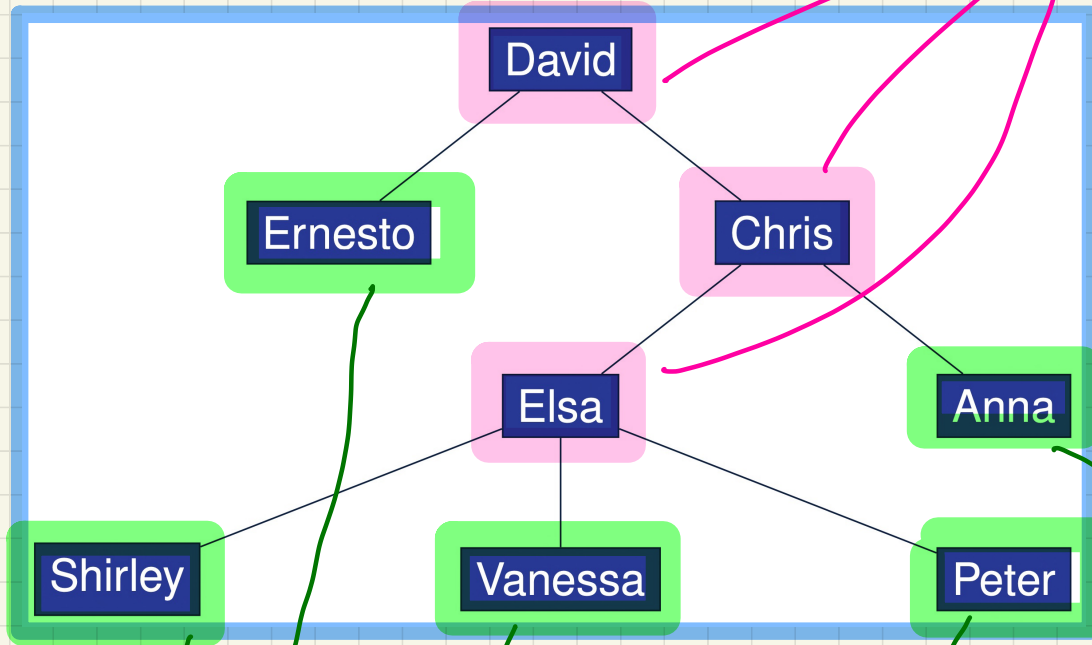


How many subtrees
in total?

$f = \#$ nodes
in the tree.

size	# subtrees
1	5
:	
:	
:	
f	① rooted at David

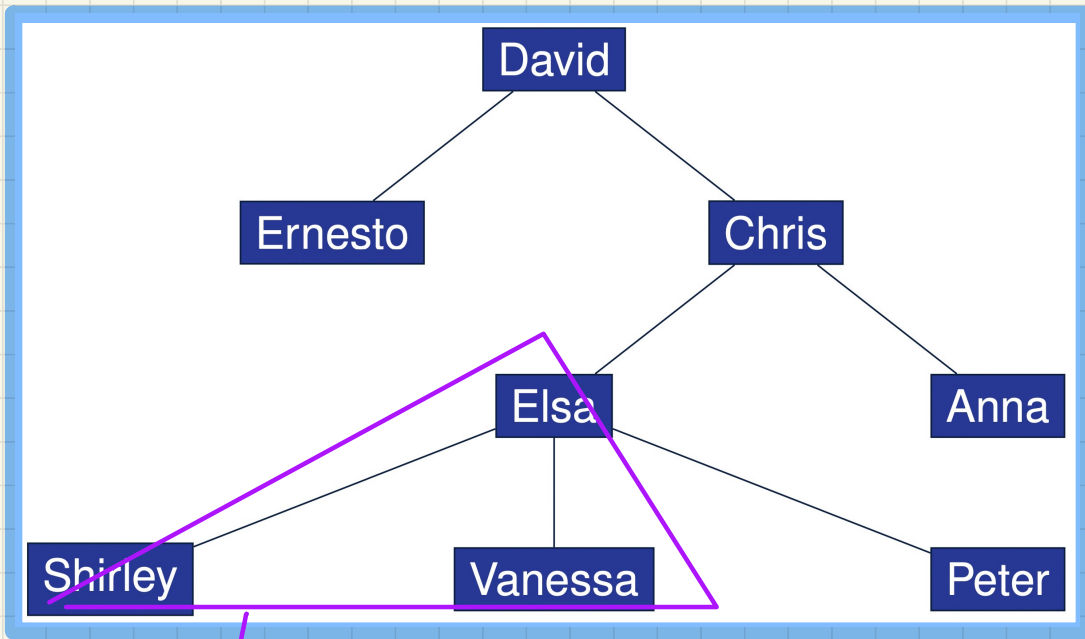
General Trees: Terminology (3)



internal nodes
→ RECURSIVE CASES

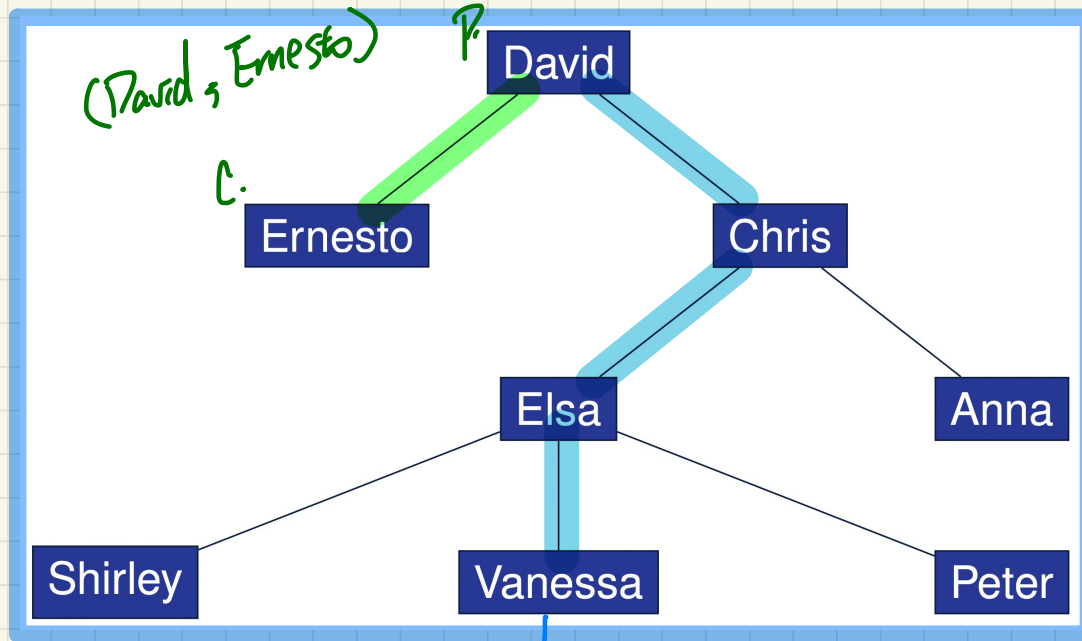
- external nodes
- internal nodes

external nodes (leaves)
→ base cases.



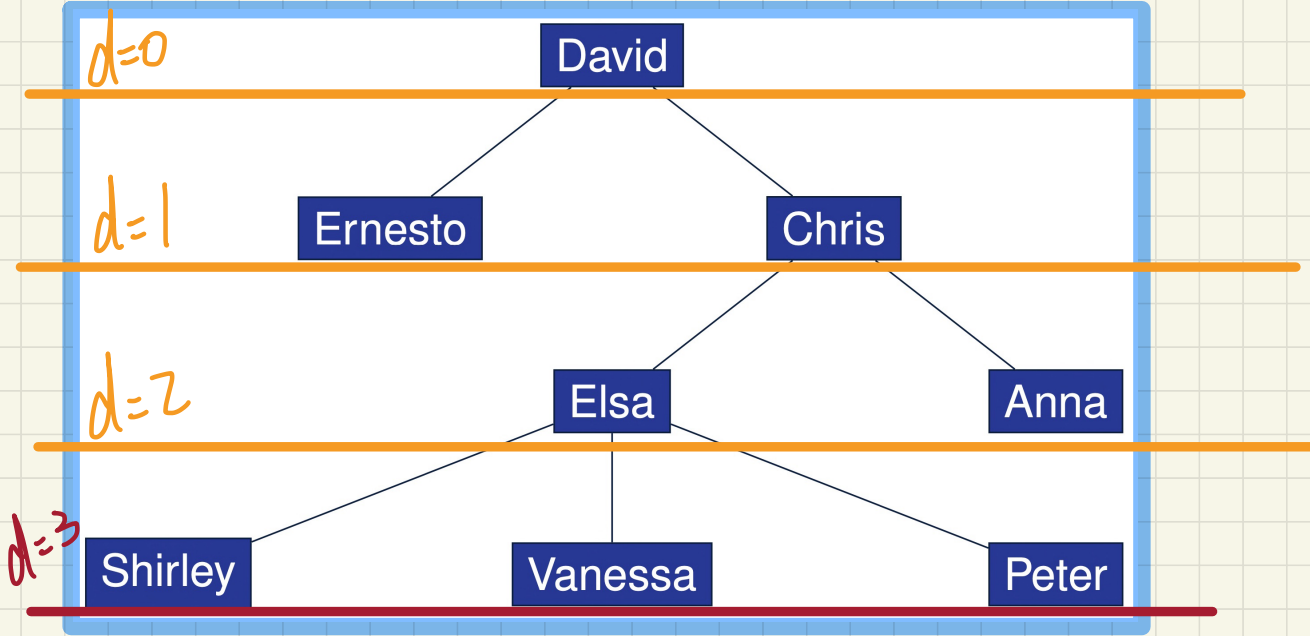
↓
a subtree
rooted at Elsa?
No ∵ Peter not included

General Trees: Terminology (4)



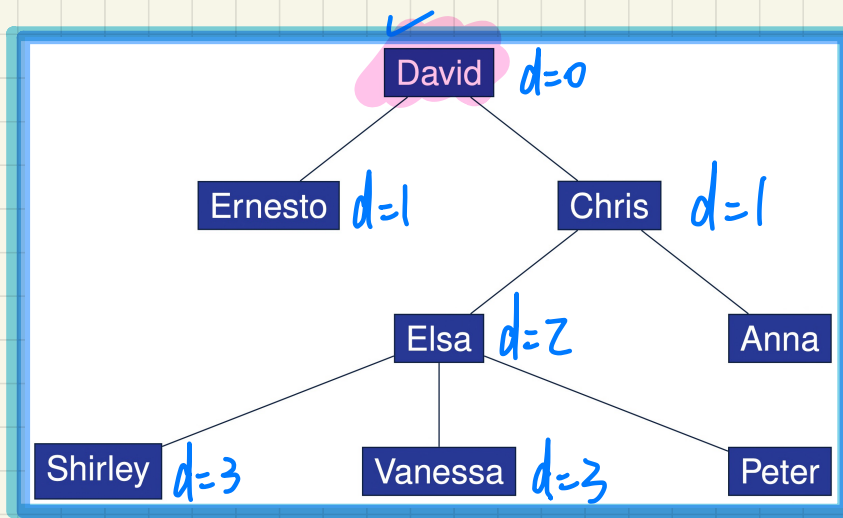
- edge
- path
- depth
- height

path
(V, E, C, D) → edges in this path



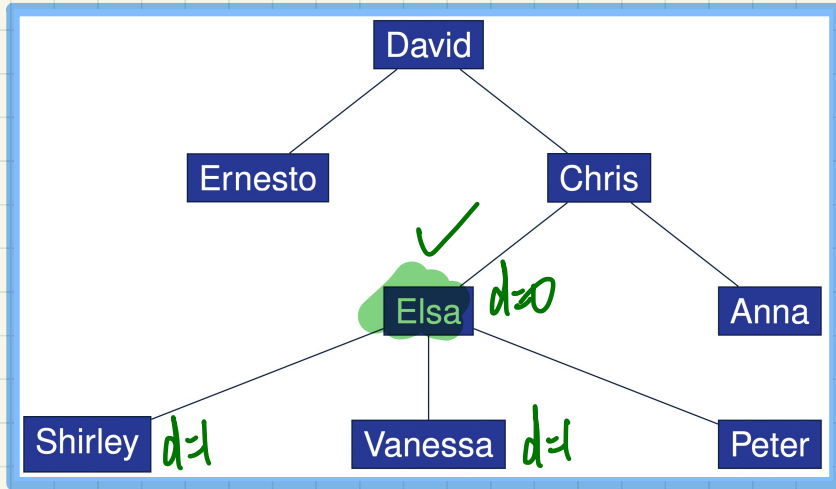
depth of a node : # of edges (parent links) from node to root.

height of a ^(sub-)tree : max depth of descendants



Height of tree rooted at David:

max d value : $\boxed{3}$

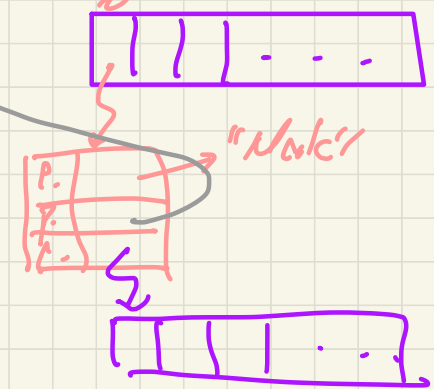
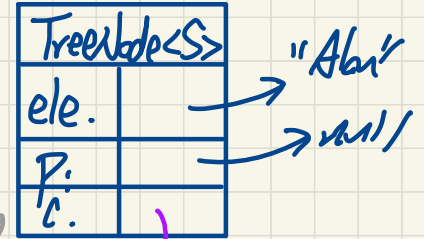


Height of tree rooted at Elsa:

max d value : $\boxed{1}$

Generic, General Tree Nodes

```
public class TreeNode<E> {  
    private E element; /* data object */  
    private TreeNode<E> parent; /* unique parent node */  
    private TreeNode<E>[] children; /* list of child nodes */  
  
    private final int MAX_NUM_CHILDREN = 10; /* fixed max */  
    private int noc; /* number of child nodes */  
  
    public TreeNode(E element) {  
        this.element = element;  
        this.parent = null;  
        this.children = (TreeNode<E>[])  
            Array.newInstance(this.getClass(), MAX_NUM_CHILDREN);  
        this.noc = 0;  
    }  
    public E getElement() { ... }  
    public TreeNode<E> getParent() { ... }  
    public TreeNode<E>[] getChildren() { ... }  
  
    public void setElement(E element) { ... }  
    public void setParent(TreeNode<E> parent) { ... }  
    public void addChild(TreeNode<E> child) { ... }  
    public void removeChildAt(int i) { ... }  
}
```



Compare:

+ prev ref.
+ next ref.
in a DLN.

