ILLINOIS

Tree Definitions

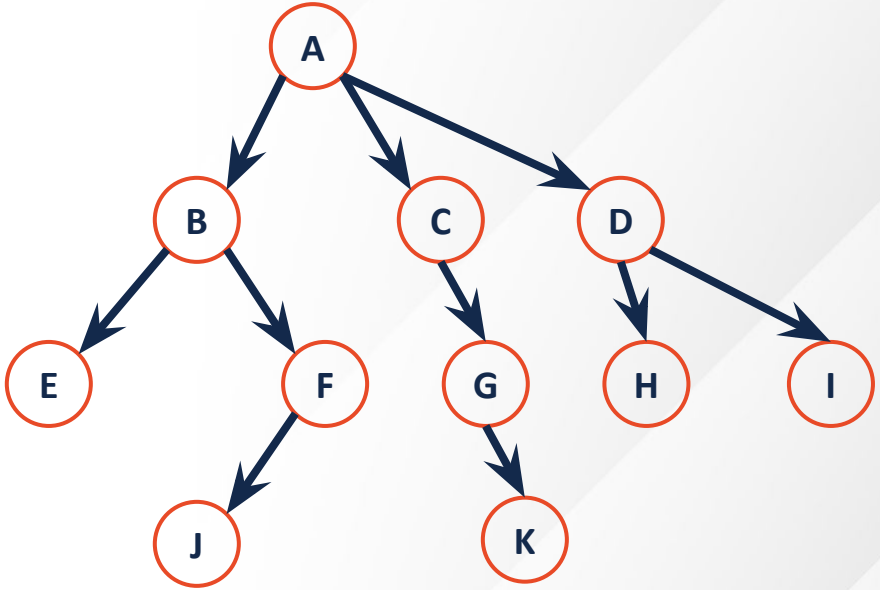
Learning Objectives

1. Classify Different Parts of Trees
2. Differentiate between different types of trees

Tree Terminology

Node

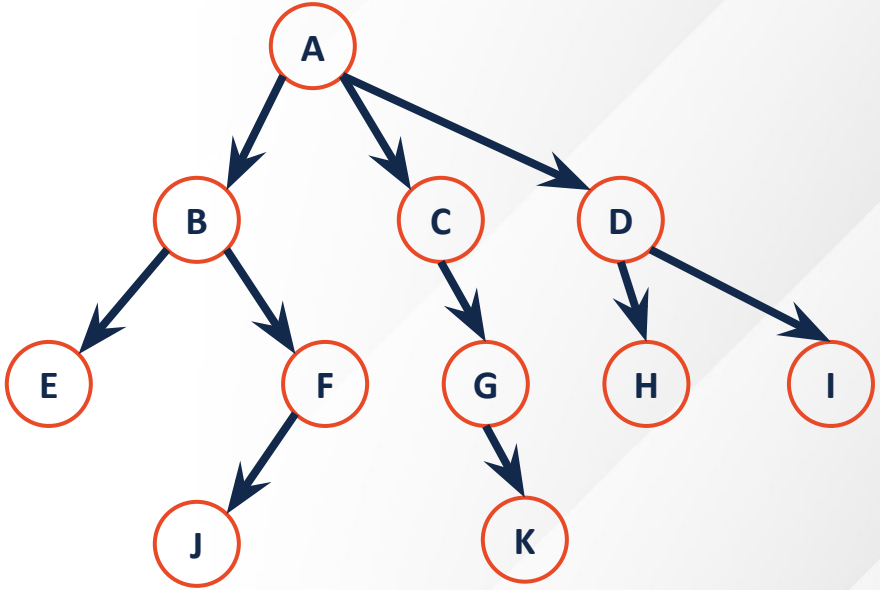
Edge



Tree Terminology

Types of Nodes

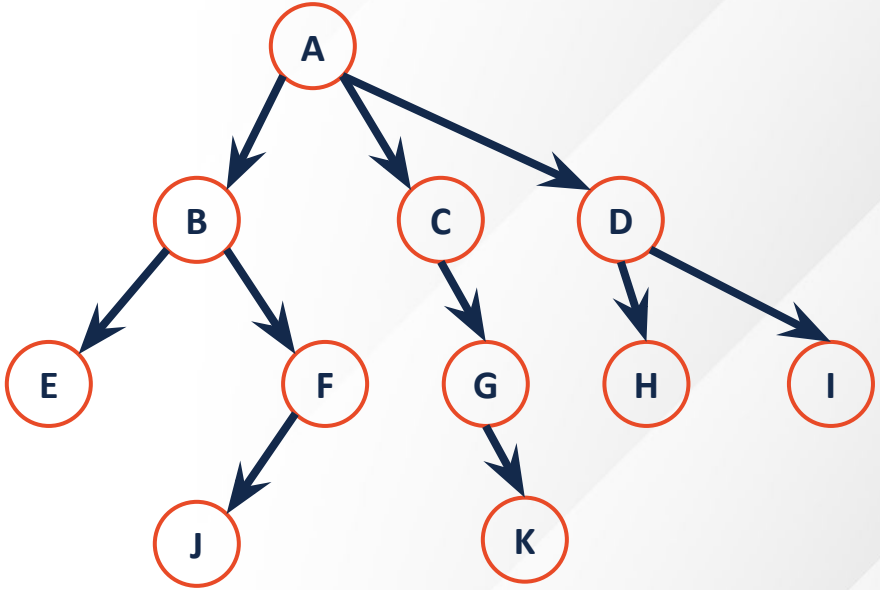
- Root
- Leaf
- Internal/Branch



Tree Terminology

Relationships:

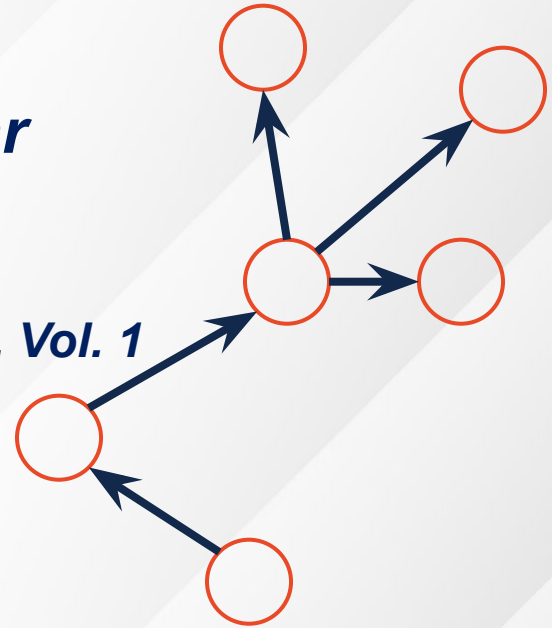
- Sibling
- Descendant
- Ancestor



Trees

"The most important non-linear data structure in computer science."

- Donald Knuth, The Art of Programming, Vol. 1



A tree is:

-
-

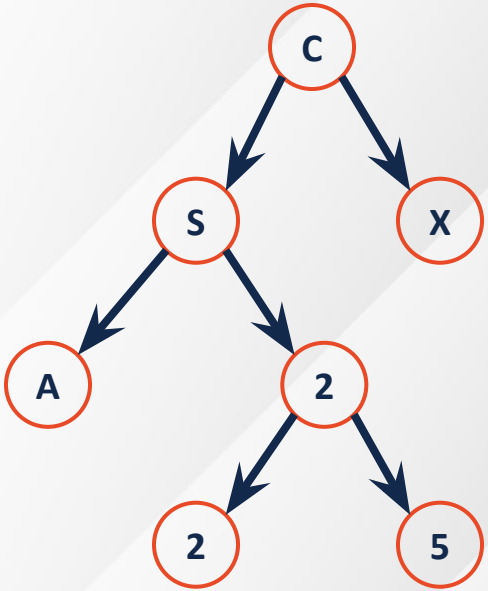
Binary Tree – Defined

A *binary tree* T is either:

-

OR

-



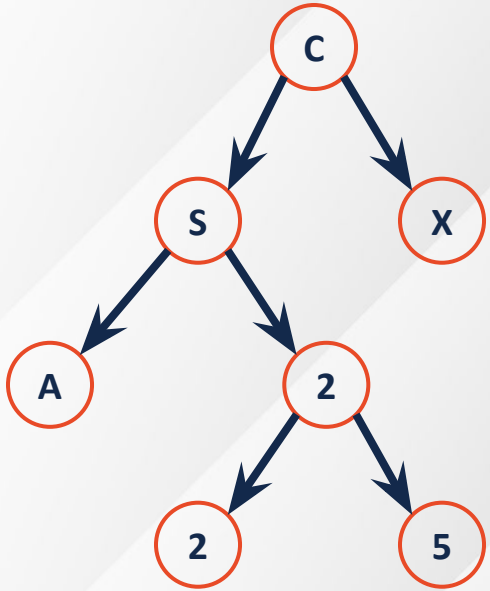
Binary Tree – Defined

A *binary tree* T is either:

$$T = (r, T_L, T_R)$$

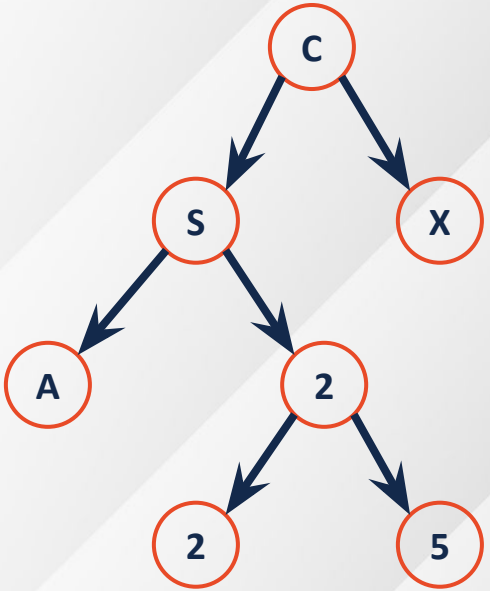
OR

$$T = \emptyset$$



Tree Property: height

***height(T)*: length of the longest path from the root to a leaf**



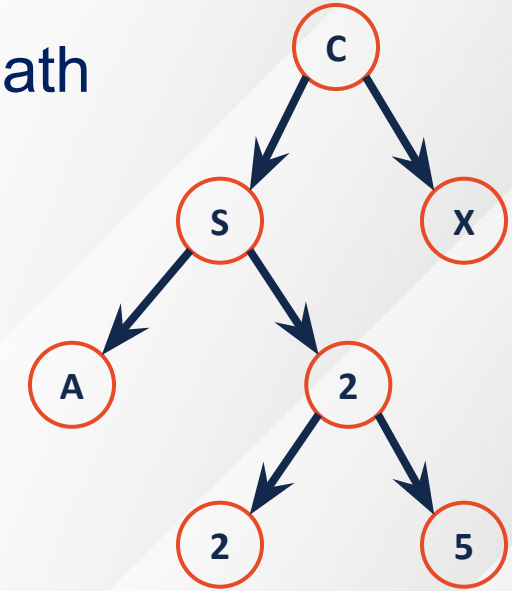
Given a binary tree T:

***height(T)* =**

***height(Ø)* =**

Tree Property: height

***height(T)*: length of the longest path from the root to a leaf**



Given a binary tree T:

$$height(T) = max(height(T_L),\ height(T_R)) + 1$$

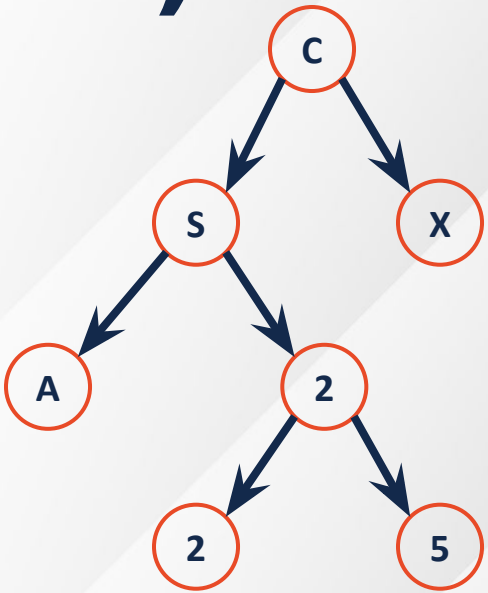
$$height(\emptyset) = -1$$

Tree Property: full (strict)

A tree ***F*** is **full** if and only if:

1.

2.



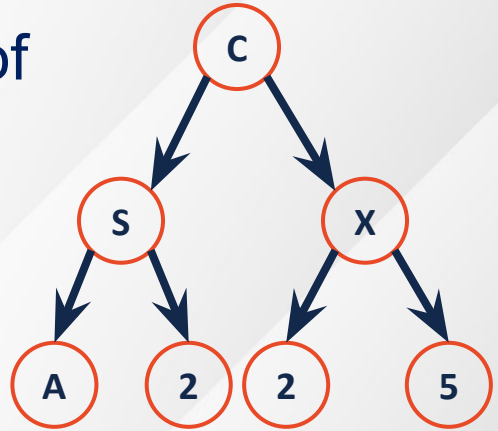
Tree Property: perfect

A **perfect** tree ***P*** is defined in terms of the tree's height.

Let $\mathbf{P_h}$ be a perfect tree of height **h**, and:

1.

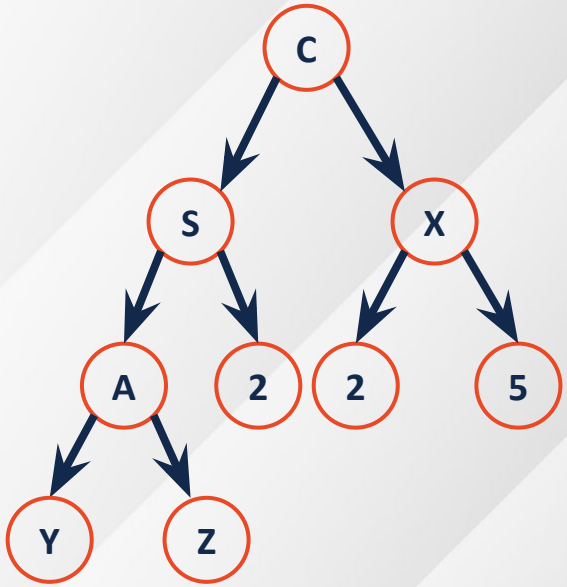
2.



Tree Property: complete

Conceptually: A perfect tree for every level except the last, where the last level if "pushed to the left".

Slightly more formal: For all levels k in [0, h-1], k has 2^k nodes. For level h, all nodes are "pushed to the left".



Tree Property: complete

A **complete** tree ***C*** of height **h**, $\mathbf{C_h}$:

1. $\mathbf{C_{-1}} = \{\}$
2. $\mathbf{C_h}$ *(where h>0)* **= {r, T_L, T_R}** and either:

T_L is _____ and T_R is _____

OR

T_L is _____ and T_R is _____