Announcements

1. MP Stickers due tonight 11:59pm
   a. Need an extension? Fill out the extension request form on the webpage
2. Exam 1 Window starts today! (ends Wednesday)
   a. If you are sick or an emergency comes up, email cs225admin
3. MP Lists will be released soon

Join Code: **225**

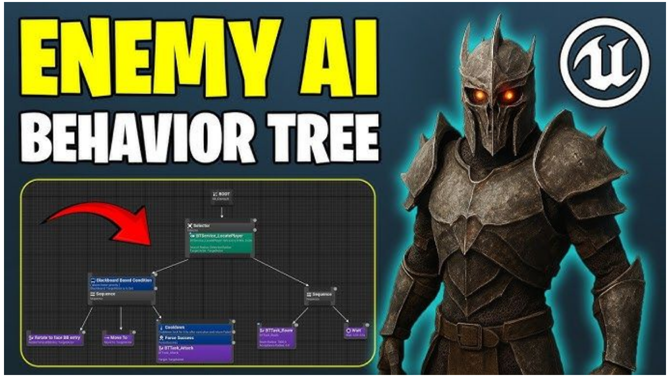
ILLINOIS

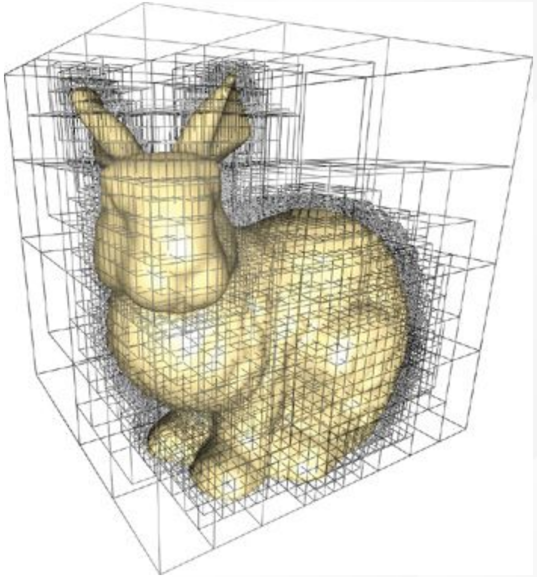
Tree Practice

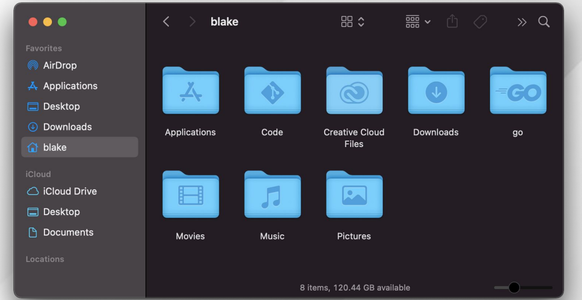
Learning Objectives

1. Define Different Parts of Trees
2. Classify different types of trees

Motivation


ENEMY AI
BEHAVIOR TREE

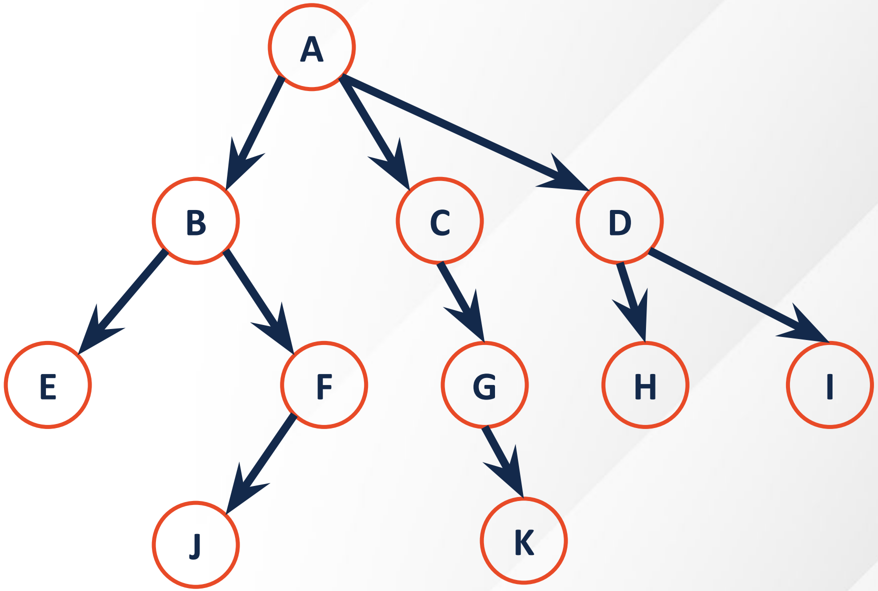



blake
Favorites
AirDrop
Applications
Desktop
Downloads
blake
iCloud
iCloud Drive
Desktop
Documents
Locations
Applications
Code
Creative Cloud Files
Downloads
go
Movies
Music
Pictures
8 items, 120.44 GB available

Tree Terminology

Node

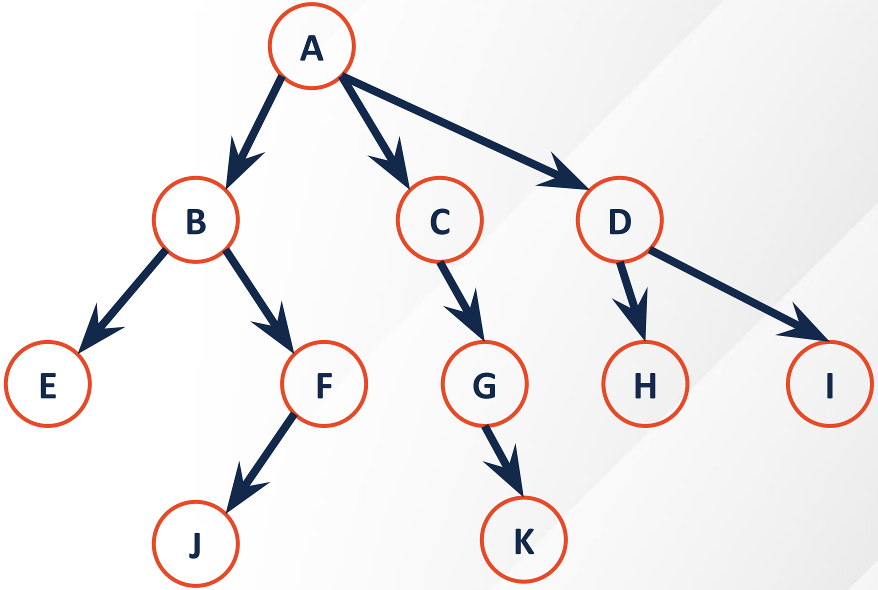
Edge



Tree Terminology

Types of Nodes

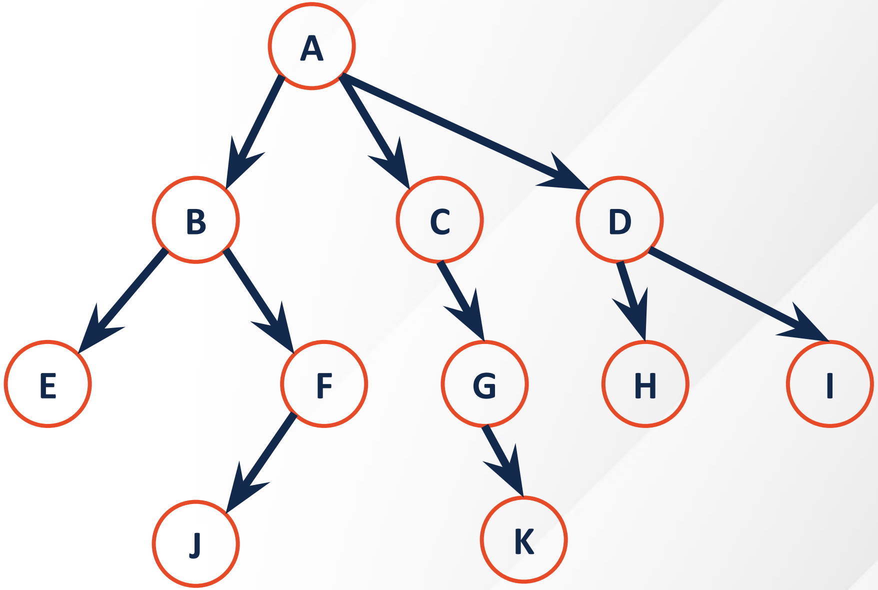
- Root
- Leaf
- Internal/Branch



Tree Terminology

Relationships:

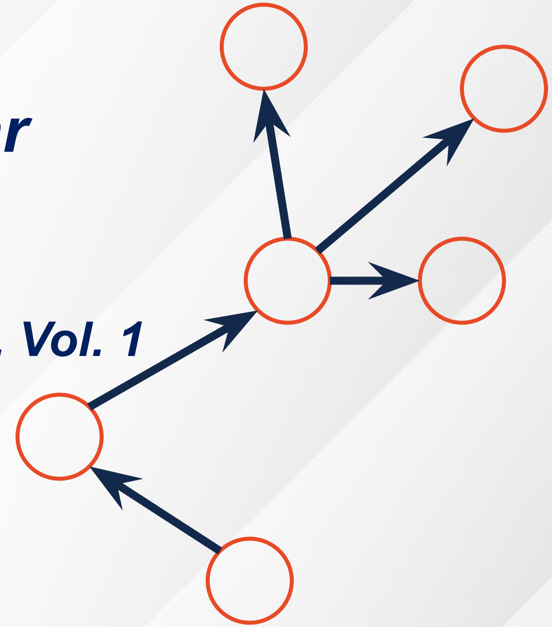
- Sibling
- Descendant
- Ancestor



Trees

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

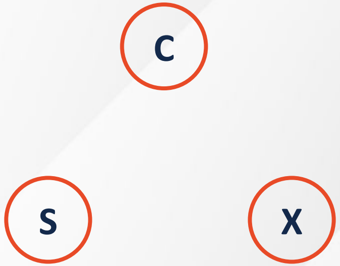
- Donald Knuth, The Art of Programming, Vol. 1



A tree is:

General Trees

How many unique trees can be made with 3 nodes?



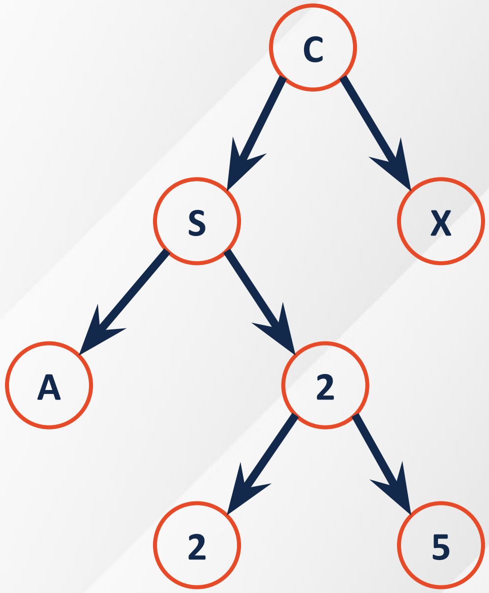
Binary Tree – Defined

A *binary tree* T is either:

-

OR

-



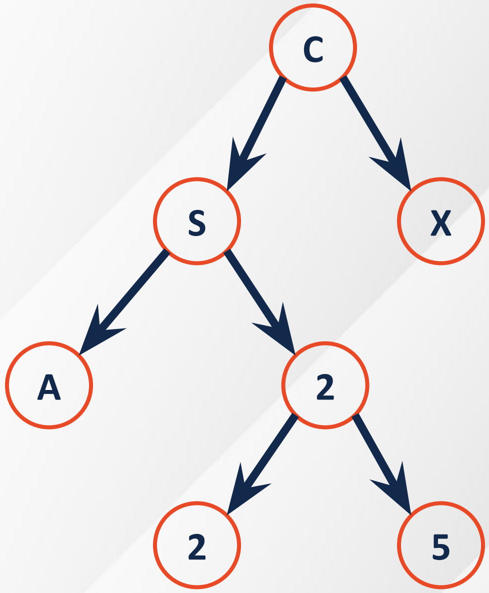
Binary Tree – Defined

A *binary tree* T is either:

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

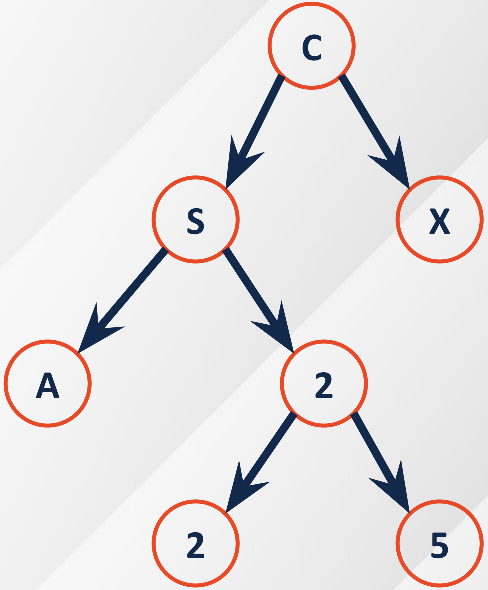
OR

$$T = \emptyset$$



Tree Property: height

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



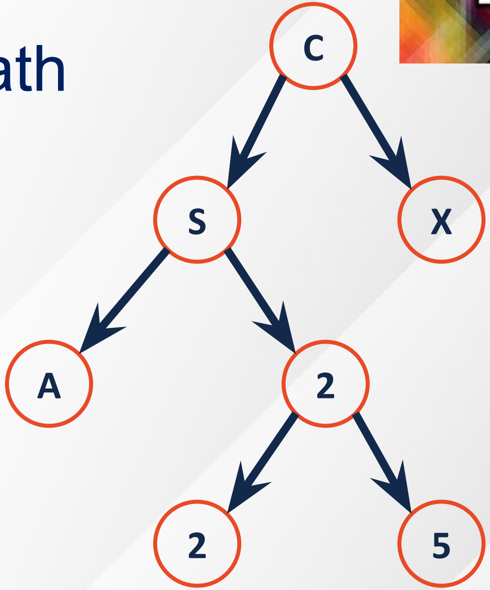
Given a binary tree T:

***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:

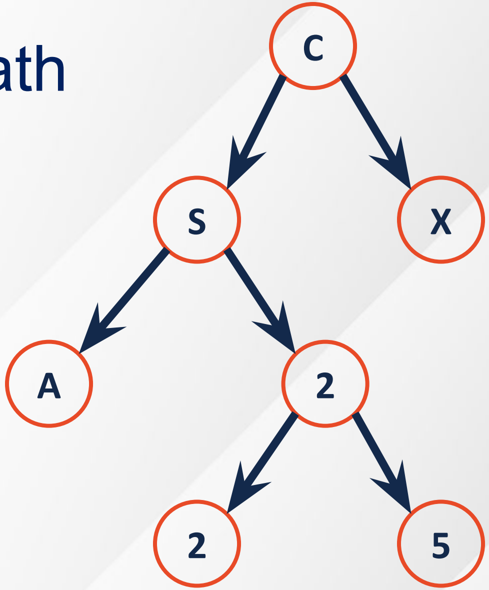
$$height(T) = max(height(T_L),\ height(T_R)) + 1$$

$$height(\emptyset) =$$

Tree Property: height

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

Given a binary tree T:



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

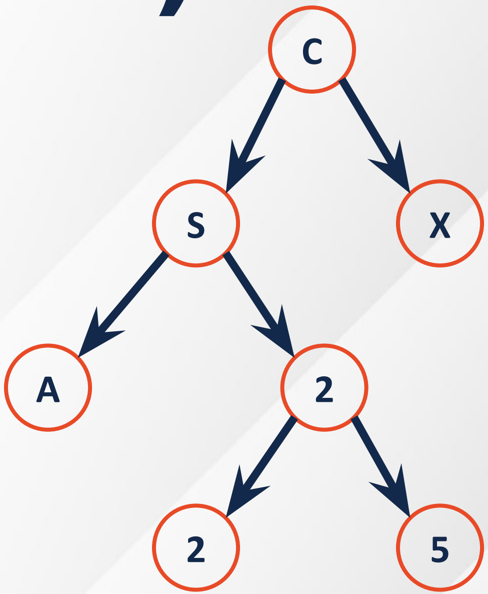
$$height(Ø) = -1$$

Tree Property: full (strict)

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

1.

2.



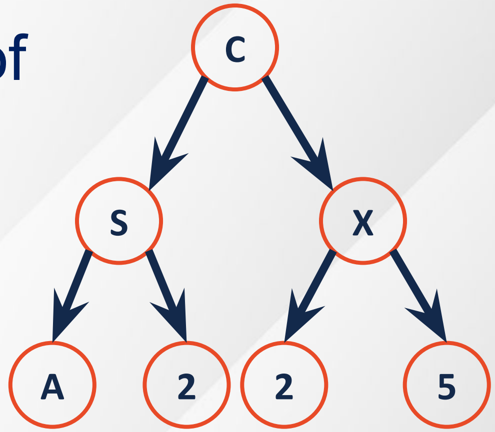
Tree Property: perfect

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.

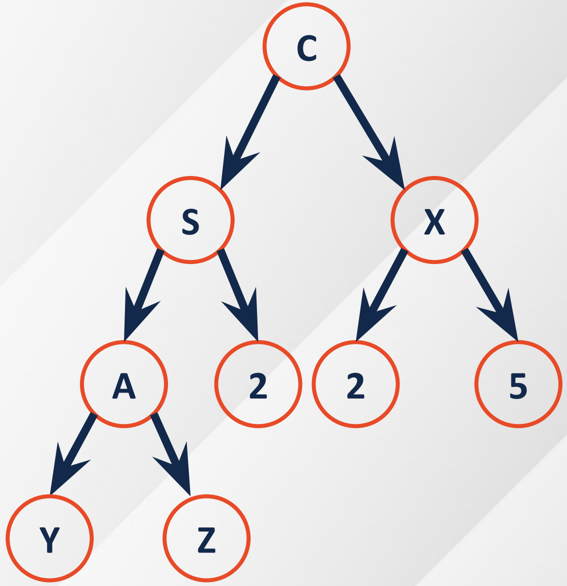
2.



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

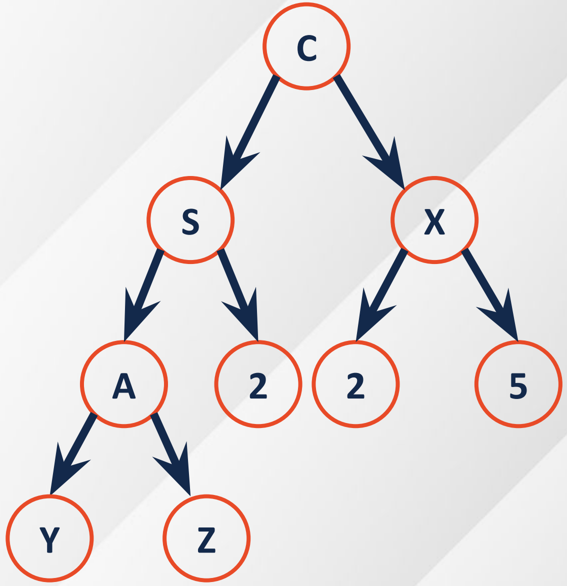
A **complete** tree ***C*** of height **h**, $\mathbf{C_h}$:

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

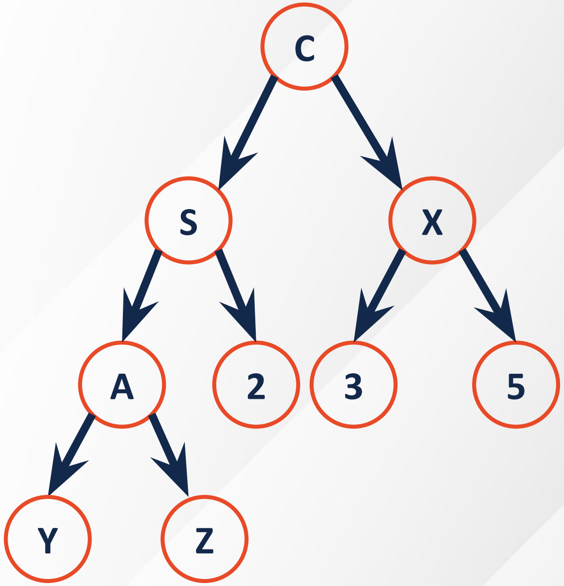
T_L is ______ and T_R is __________



Tree Property: complete

1:00

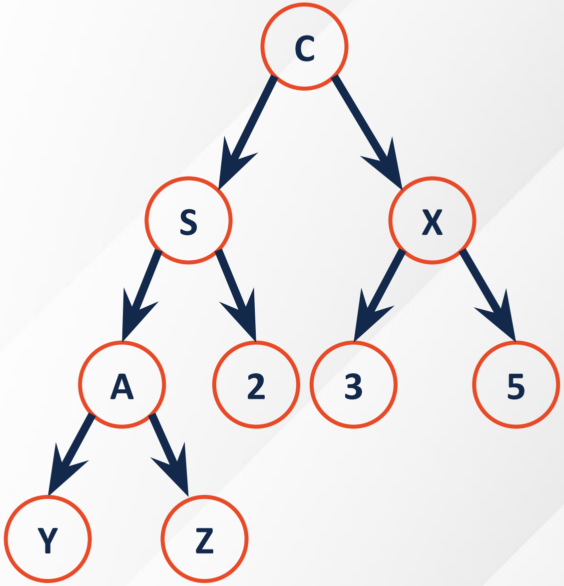
Is every **full** tree **complete**?



Tree Property: complete

1:00

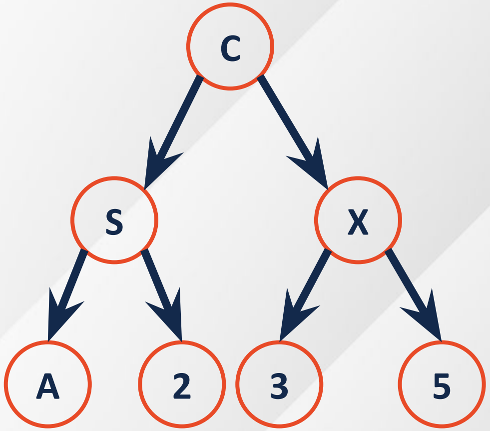
Is every **complete** tree **full**?



Perfect Trees

Given a perfect tree with height h:

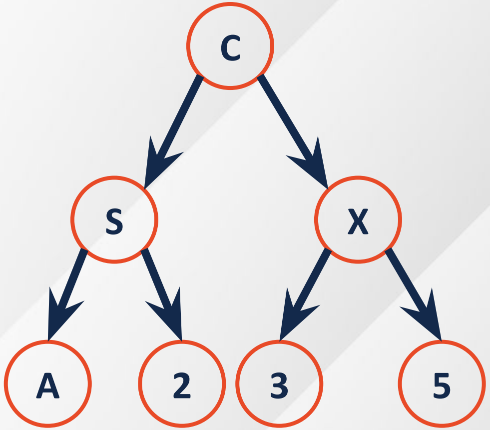
How many leaf nodes are there?



Perfect Trees

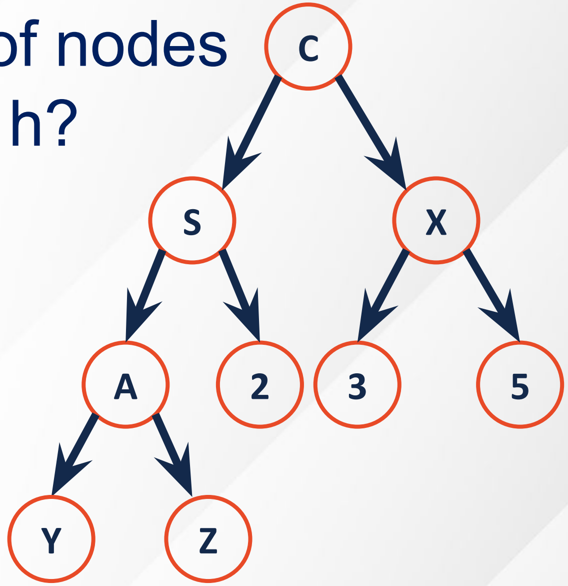
Given a perfect tree with height h:

How many nodes in total?



Complete Trees

What is the range of number of nodes in a complete tree with height h?



General Trees

If a tree has n_1 nodes with 1 child,

n_2 nodes with 2 children,

…

n_m nodes with m children,

then how many leaf nodes are there?

General Trees

Wasted Pointers in a binary tree?