## 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 has been released!
    a. EC for Part 1 early submission

Join Code: **225**

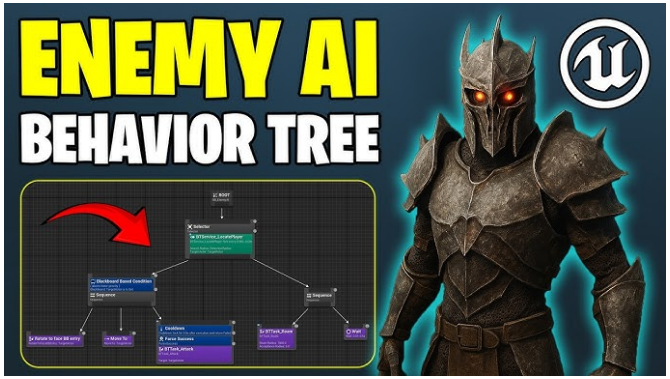
ILLINOIS

# Tree Introduction

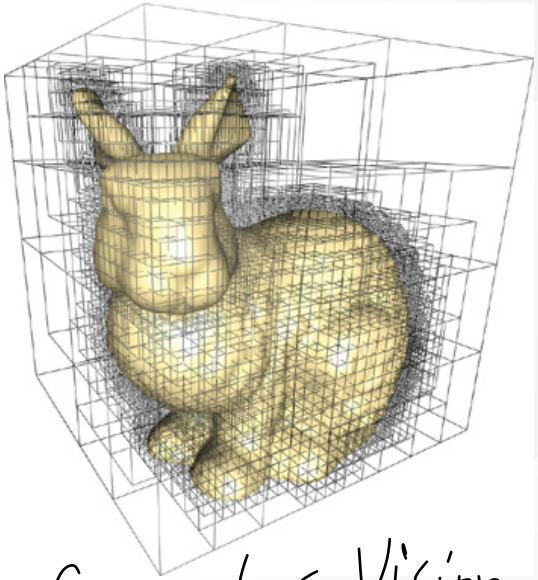
## Learning Objectives

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

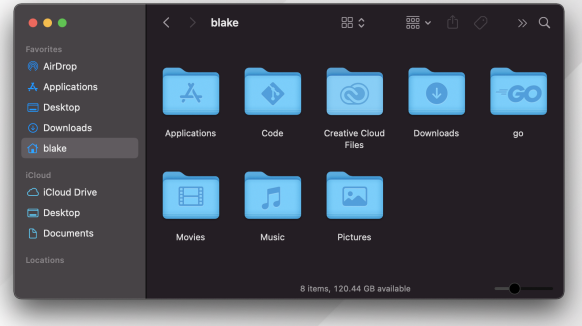
# Motivation




Decision/Behavior Trees



Computer Vision 3D Reconstruction

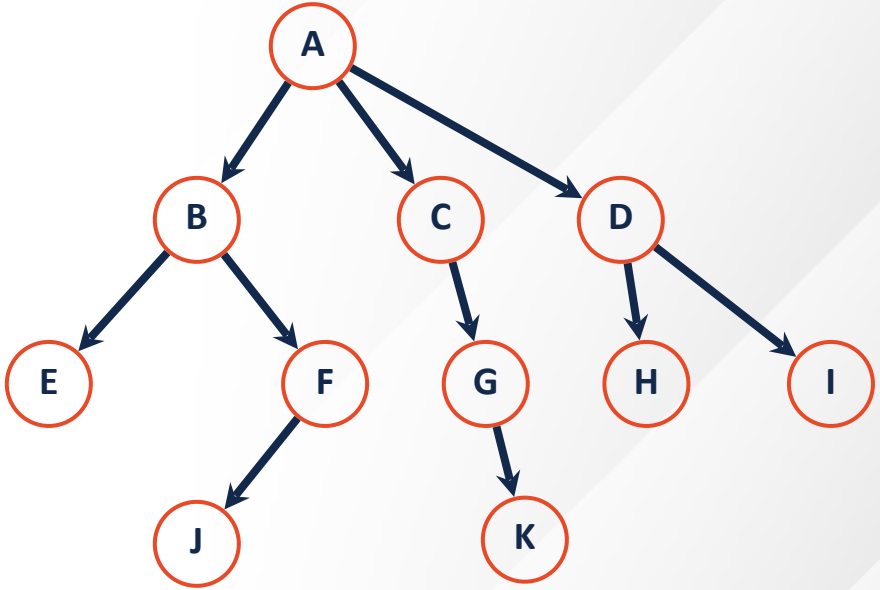



File Structure

# Tree Terminology

Node

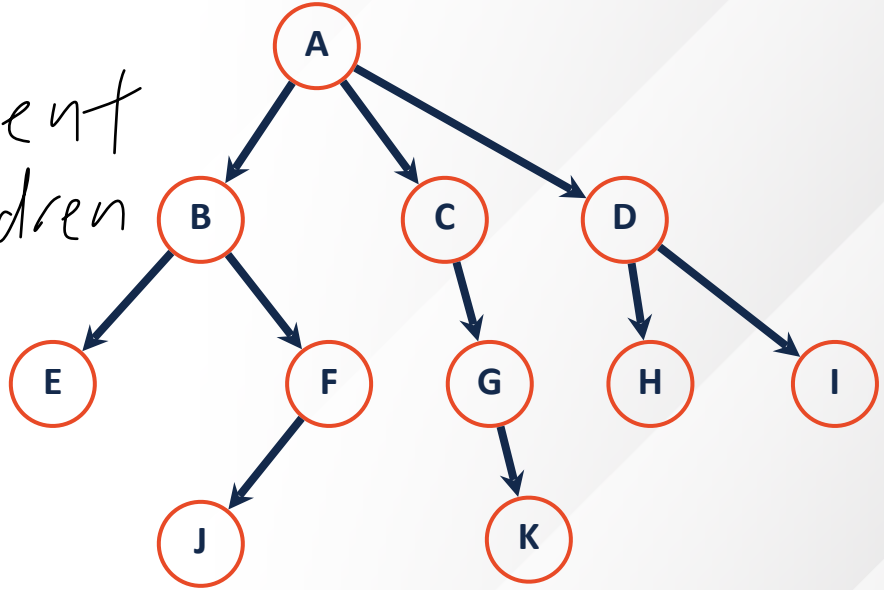
Edge



# Tree Terminology

Types of Nodes

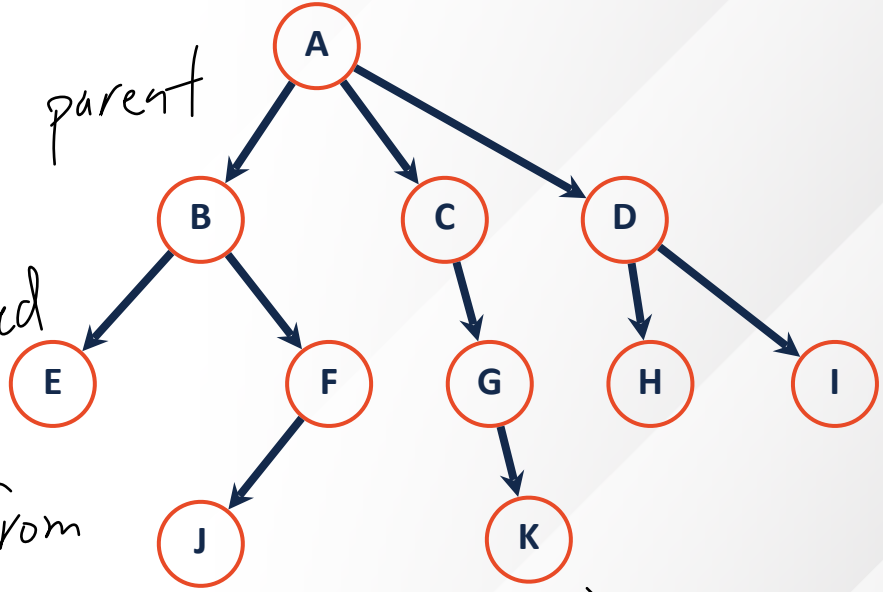
- Root – No Parent
- Leaf – No children
- Internal/Branch



# Tree Terminology

Relationships:

- Sibling — Shared parent
- Descendant
- Ancestor
- Neighbor — connected by an edge




Given a path from p to q,

if p comes before q, p is an ancestor of q

if p comes after q, p is a descendant

# Trees

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

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

**A tree is:**

- Null OR
- Rooted and children are disjoint OR
  - Acyclic Undirected Graph

# General Trees

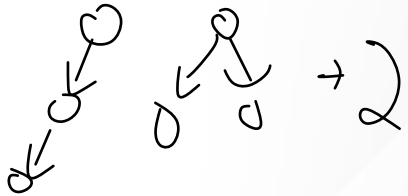
2:00

How many unique trees can be made with 3 nodes?

C

S

X

1. structure  → 2

2. Permutations, 3!

Total = 2 * 3! = 12

Join Code: **225**

# Binary Tree – Defined

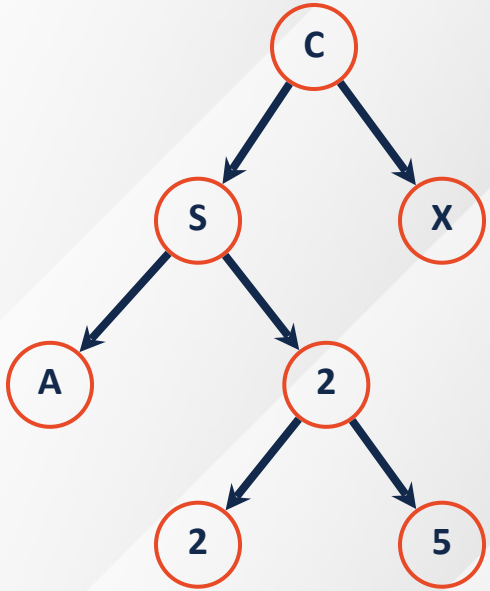
**A *binary tree* T is either:**

- Null

**OR**

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

where $T_L$ & $T_R$ are binary trees



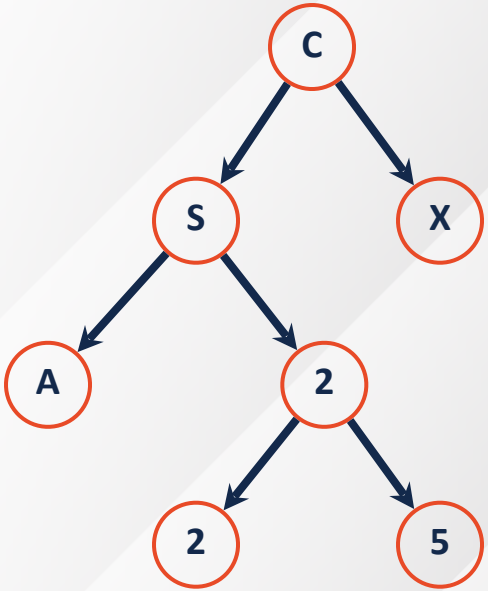
# Binary Tree – Defined

A *binary tree* T is either:

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

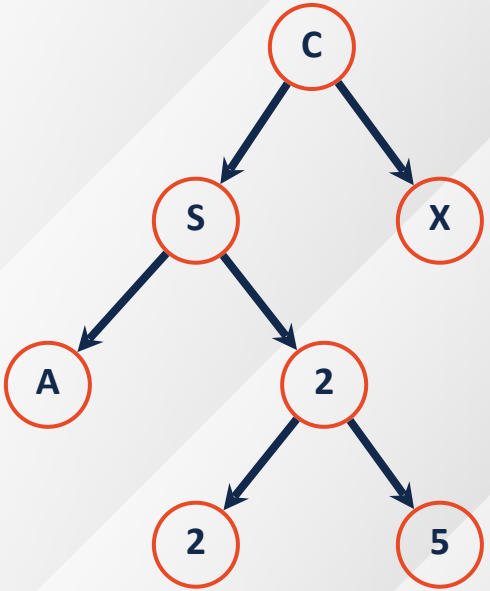
OR

$$T = \emptyset$$



# Tree Property: height

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




**Given a binary tree T:**

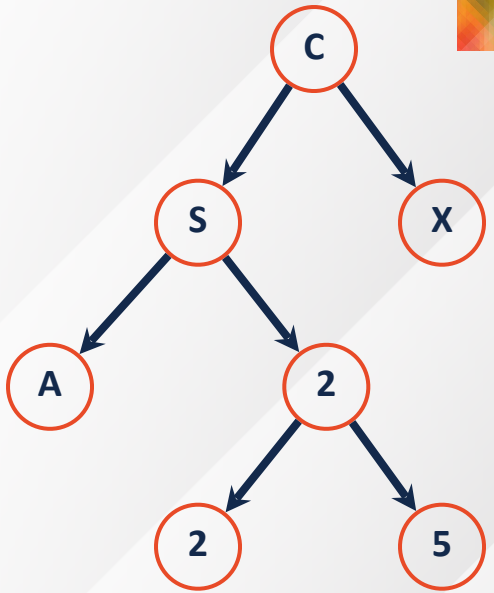
***height(T)* =** $\max(height(T_L), height(T_R)) + 1$

***height(Ø)* =** $-1$

# Tree Property: height

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

**Given a binary tree T:**



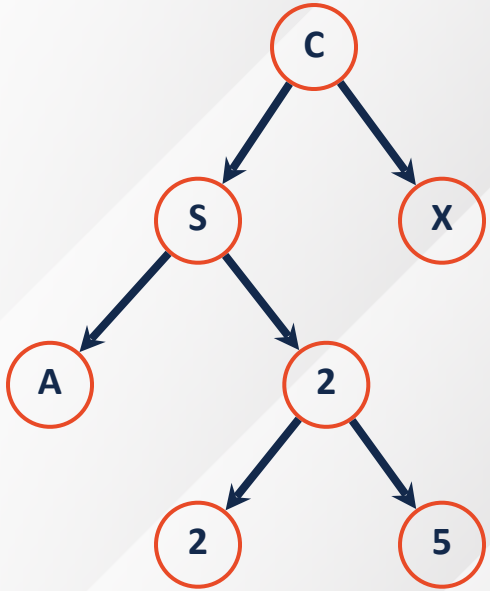

$$height(T) = max(height(T_L), height(T_R)) + 1$$

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

Join Code: **225**

# Tree Property: height

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




**Given a binary tree T:**

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

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

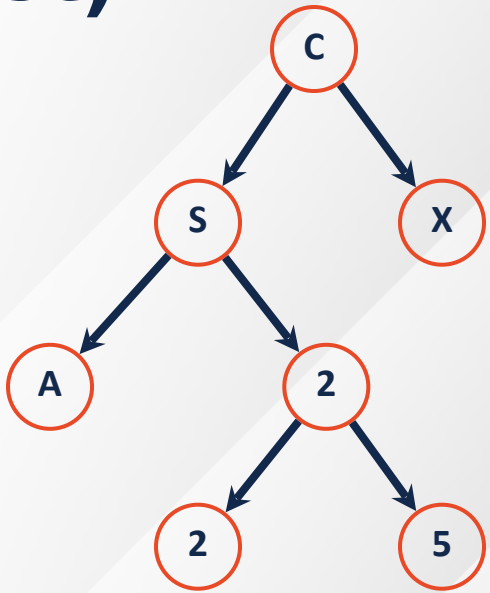
# Tree Property: full (strict)

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

1. $F = \{\}$
2. $F = \{r, T_L, T_R\}$

where $T_L$ & $T_R$ are empty

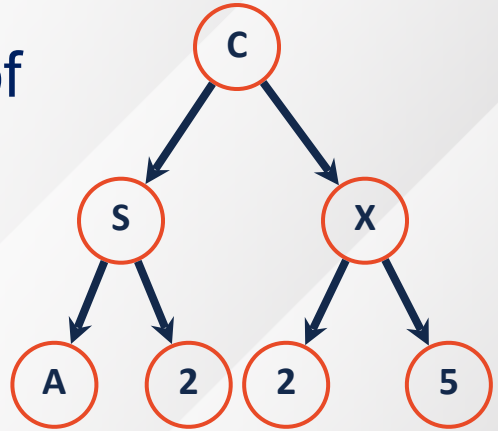
or $T_L$ & $T_R$ are not empty




Every node has 0 or 2 children

# Tree Property: perfect

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



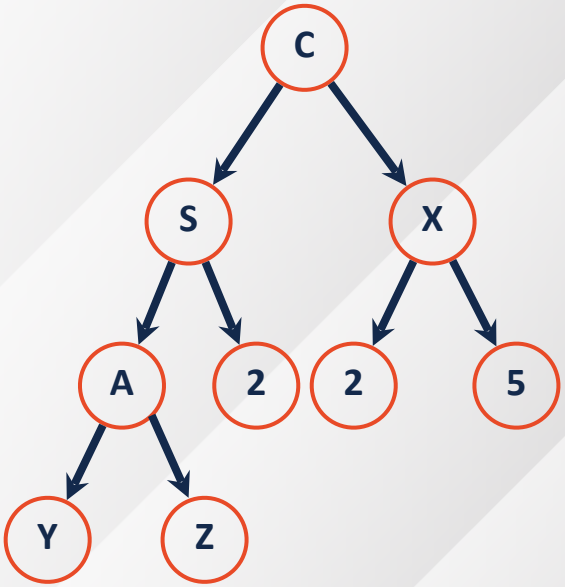

Let $\mathbf{P_h}$ be a perfect tree of height **h**, and:

1. $P_{-1} = \{\}$
2. $P_h = \{r, T_L, T_R\}$ where $T_L$ & $T_R$ are $P_{h-1}$

# 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

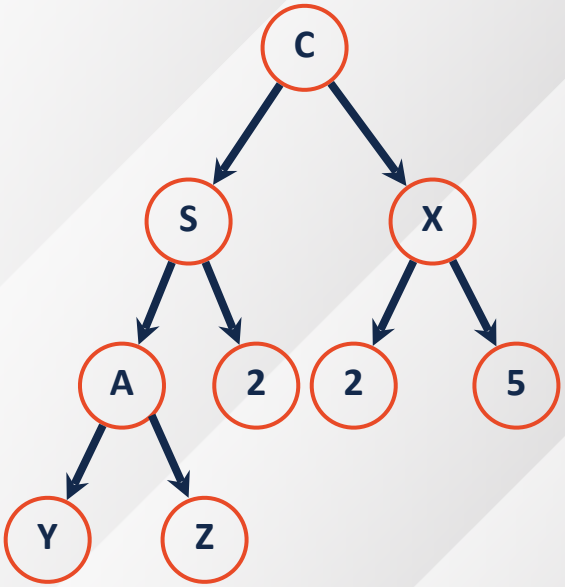
A **complete** tree ***C*** of height **h**, $\mathbf{C_h}$:

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

$T_L$ is $C_{h-1}$ and $T_R$ is $P_{h-2}$

**OR**

$T_L$ is $P_{h-1}$ and $T_R$ is $C_{h-1}$

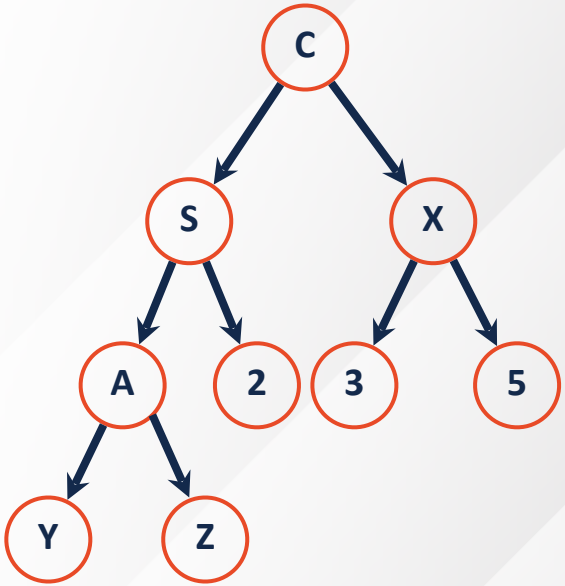


# Tree Property: complete

1:00

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

No



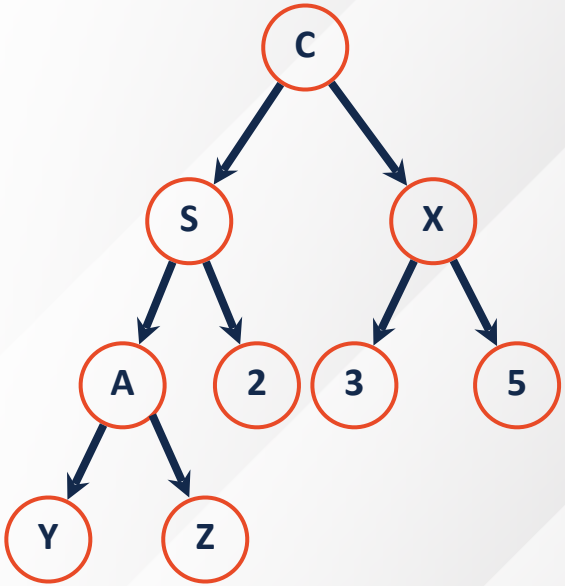

Join Code: **225**

# Tree Property: complete

1:00

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

No



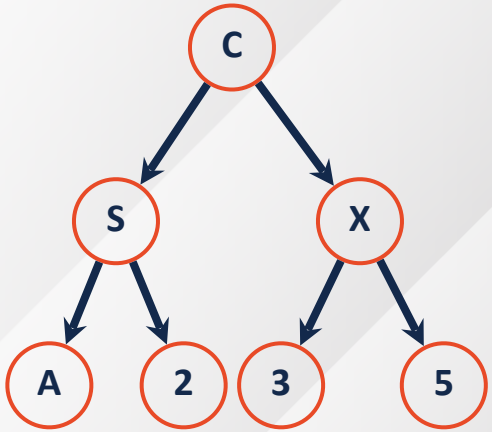

Join Code: **225**

# Perfect Trees




Given a perfect tree with height h:

How many leaf nodes are there?



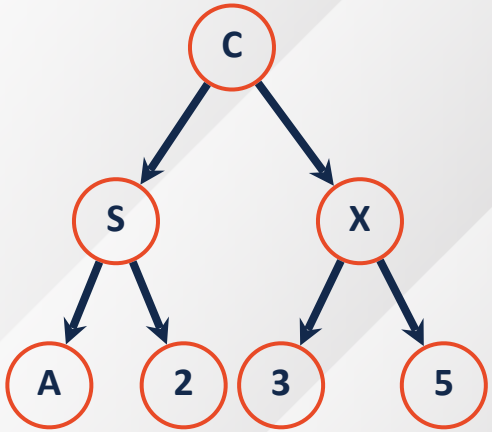

Join Code: **225**

# Perfect Trees

2:00

Given a perfect tree with height h:

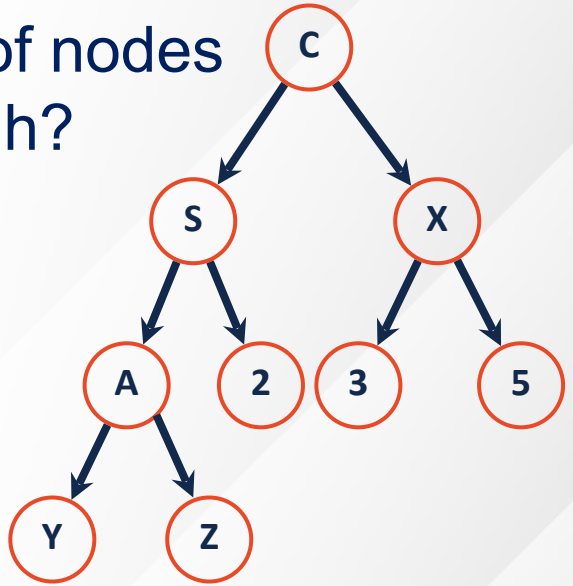
How many nodes in total?




Join Code: **225**

# 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?