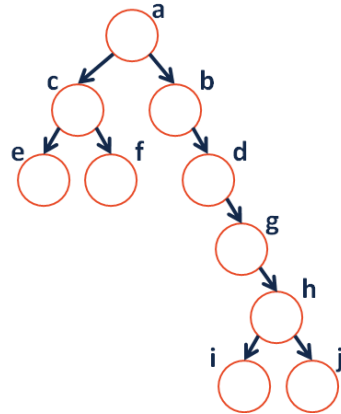
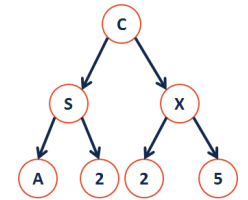


We will primarily talk about **binary trees**:

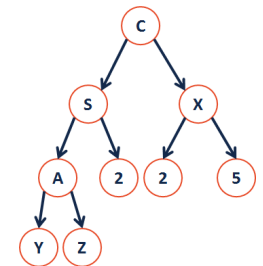
- How many parents does each vertex have?
- Which vertex has the fewest **children**?
- Which vertex has the most **ancestors**?
- Which vertex has the most **descendants**?
- List all the vertices in b's left **subtree**.
- List all the **leaves** in the tree.



Tree Property: Perfect

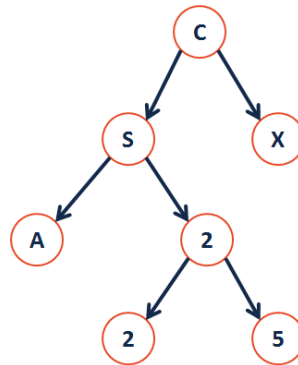


Tree Property: Complete



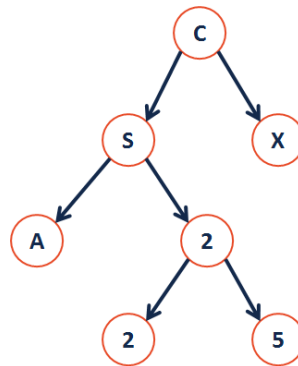
Definition: Binary Tree

A *binary tree T* is:



The height of a tree **T** is:

Tree Property: Full



Towards a Tree Implementation – Tree ADT:

ADT Functionality (English Description)	Function Call

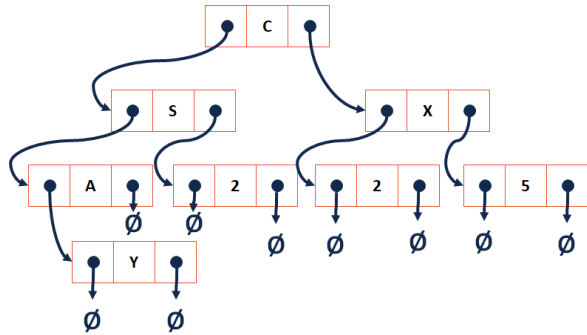
Tree Class

```

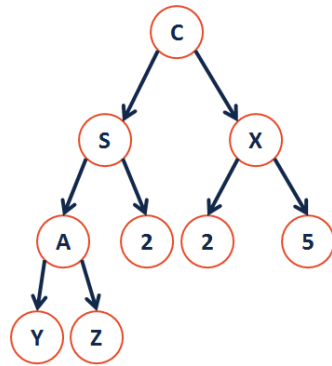
BinaryTree.h
1 #pragma once
2
3 template <typename T>
4 class BinaryTree {
5     public:
6         /* ... */
7     private:
8
9
10
11
12 };

```

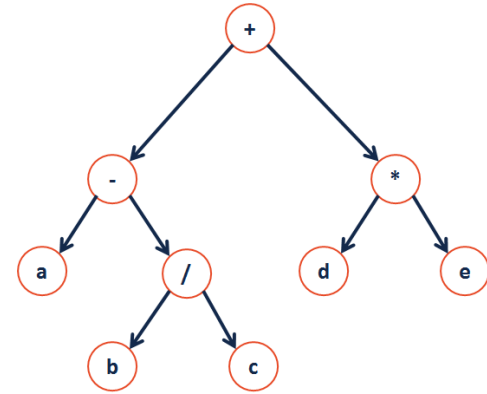
Trees are nothing new – they're fancy linked lists:



Theorem: If there are n data items in our representation of a binary tree, then there are _____ NULL pointers.



Traversals:



CS 225 – Things To Be Doing:

1. Programming Exam A is on-going (ends on Monday!)
2. MP3 extra credit deadline is Monday!
3. lab_quacks due Sunday
4. Daily POTDs