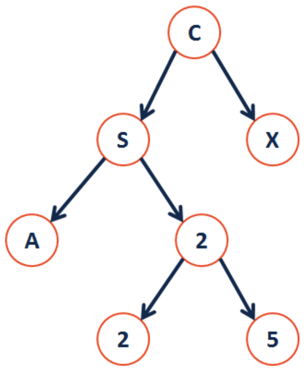
#13: Trees

February 16, 2022 · *G Carl Evans*

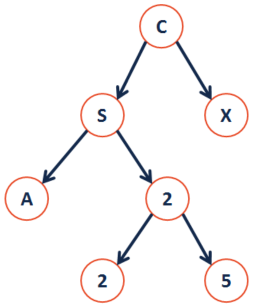
Definition: Binary Tree

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

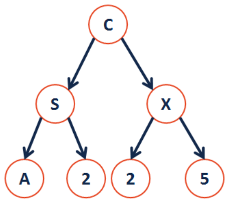


The height of a tree **T** is:

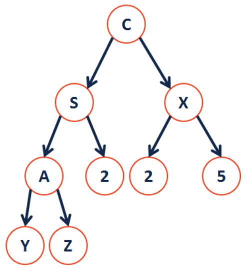
Tree Property: Full



Tree Property: Perfect



Tree Property: Complete



Towards a Tree Implementation – Tree ADT:

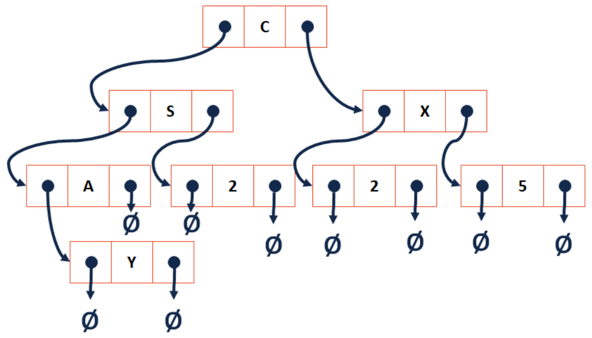
ADT Functionality (English Description)	**Function Call**

Tree Class

BinaryTree.h

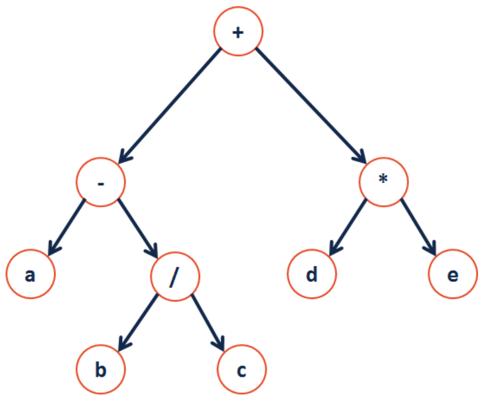
```
#pragma once

template <typename T>
class BinaryTree {
  public:
    /* ... */
  private:




};
```

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 _____________ **nullptr**s.

Traversals:



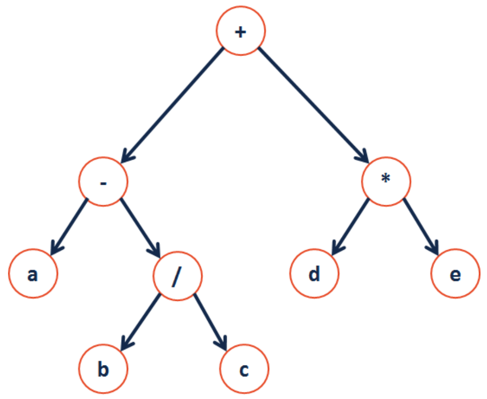
One Algorithm, Three Traversals:

BinaryTree.cpp

```
void BinaryTree<T>::____Order(TreeNode * cur) {
  if (cur != nullptr) {





  }
}
```

A Different Type of Traversal

Strategy:



Traversal vs. Search:

Breadth First Search:

Depth First Search:

CS 225 – Things To Be Doing:

1. mp_list due Sunday.
2. lab_inheritance starts today
3. exam 1 ongoing.
4. Daily POTDs