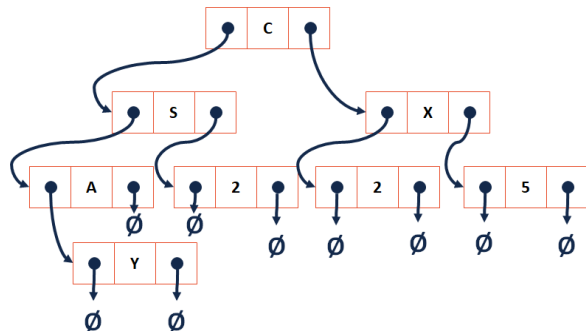
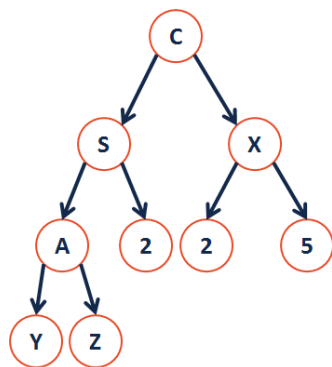


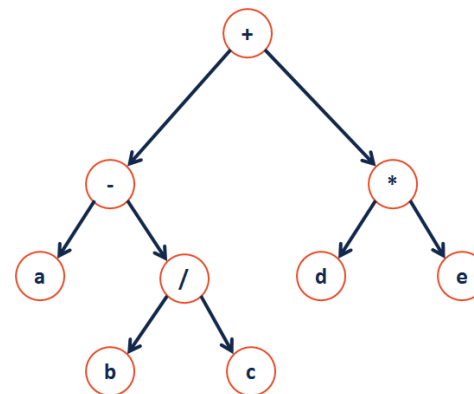
Trees are nothing new – they're fancy linked lists:



**Theorem:** If there are  $n$  data items in our representation of a binary tree, then there are \_\_\_\_\_ **nullptrs**.



Traversals:

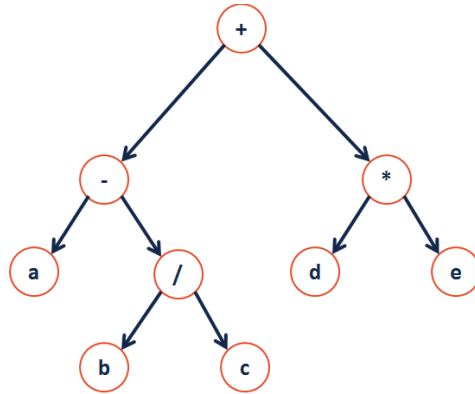


One Algorithm, Three Traversals:

BinaryTree.cpp	
50	void BinaryTree<T>::Order(TreeNode * cur) {
51	if (cur != nullptr) {
52	
53	
54	
55	
56	
57	}
58	}

## A Different Type of Traversal

Strategy:



```
BinaryTree.cpp
void BinaryTree<T>::levelOrder(TreeNode * root) {
}
}
```

Traversal vs. Search:

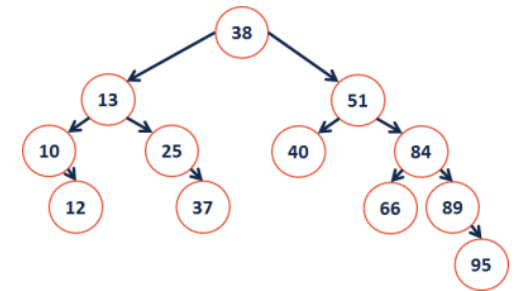
Breadth First Search:

Depth First Search:

## Dictionary ADT

```
Dictionary.h
1 #pragma once
2
3
4
5 class Dictionary {
6 public:
7
8
9
10
11
12
13
14 private:
15 // ...
16 };
```

## A Searchable Binary Tree?



```
BST.h
private:
```

## CS 225 – Things To Be Doing:

1. Exam 4 Friday in lecture
2. PotDs with workspaces to practice
3. lab\_trees starts today
4. Daily POTDs