CS 225

Data Structures

Feb. 16 – Tree Proof

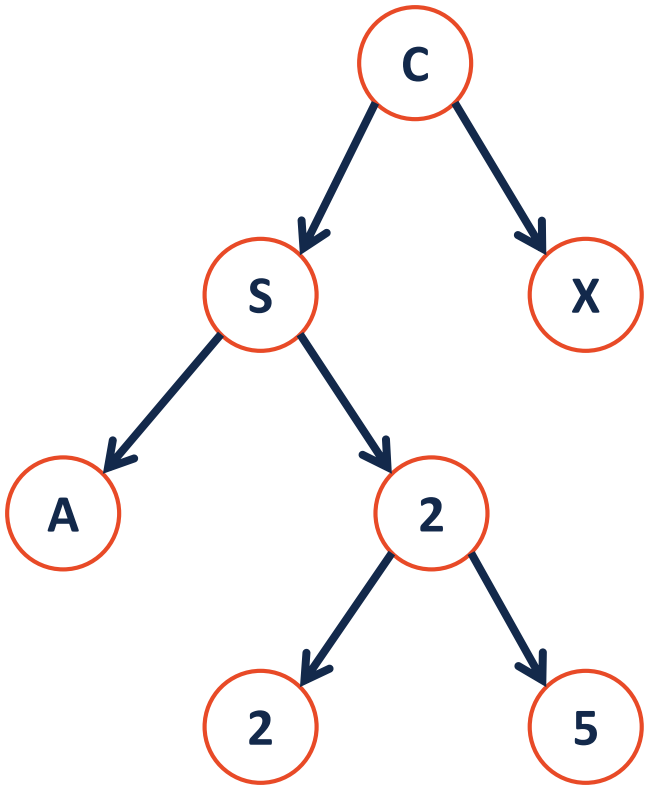
Wade Fagen-Ulmschneider

Tree Property: full

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

1.

2.



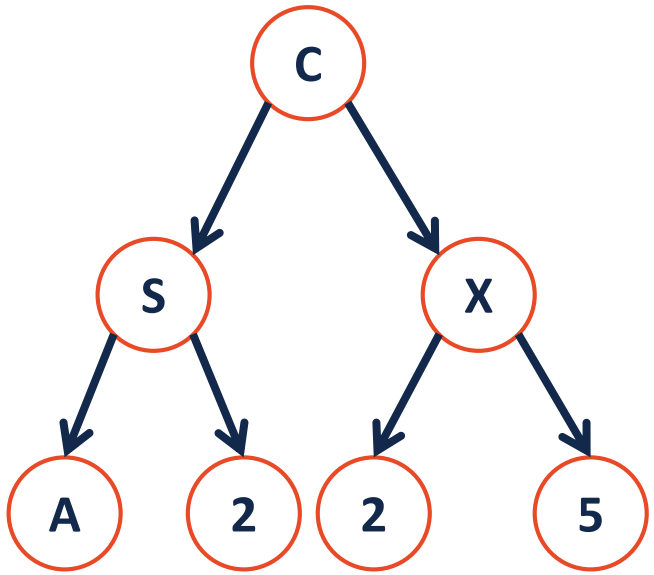
Tree Property: perfect

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.

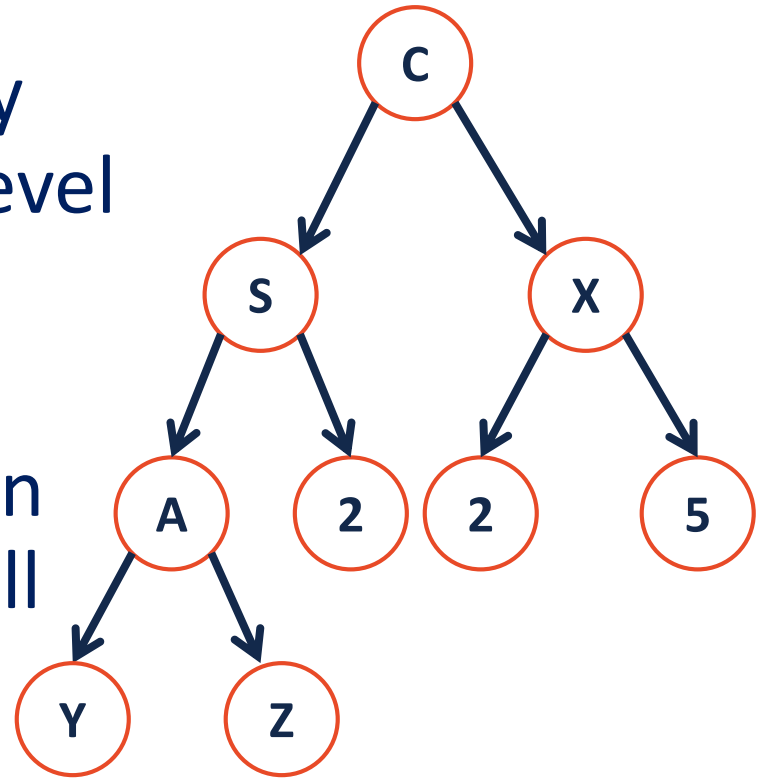
2.



Tree Property: complete

Conceptually: A perfect tree for every level except the last, where the last level if "pushed to the left".

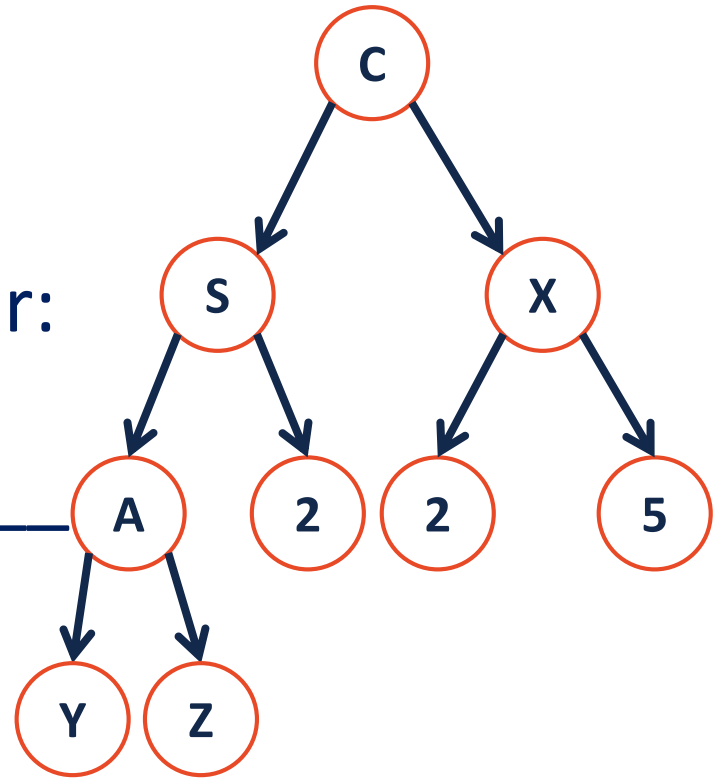
Slightly more formal: For all levels k in [0, h-1], k has 2^k nodes. For level h, all nodes are "pushed to the left".



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, $\mathbf{T_L}$, $\mathbf{T_R}$}** and either:



T_L is _____ and T_R is _____

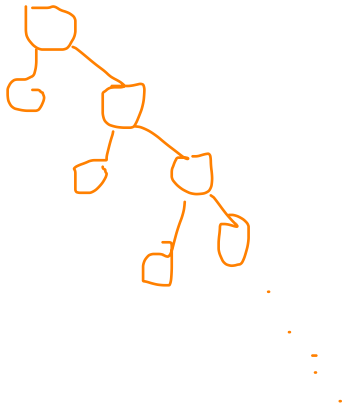
OR

T_L is _____ and T_R is _____

Tree Property: complete

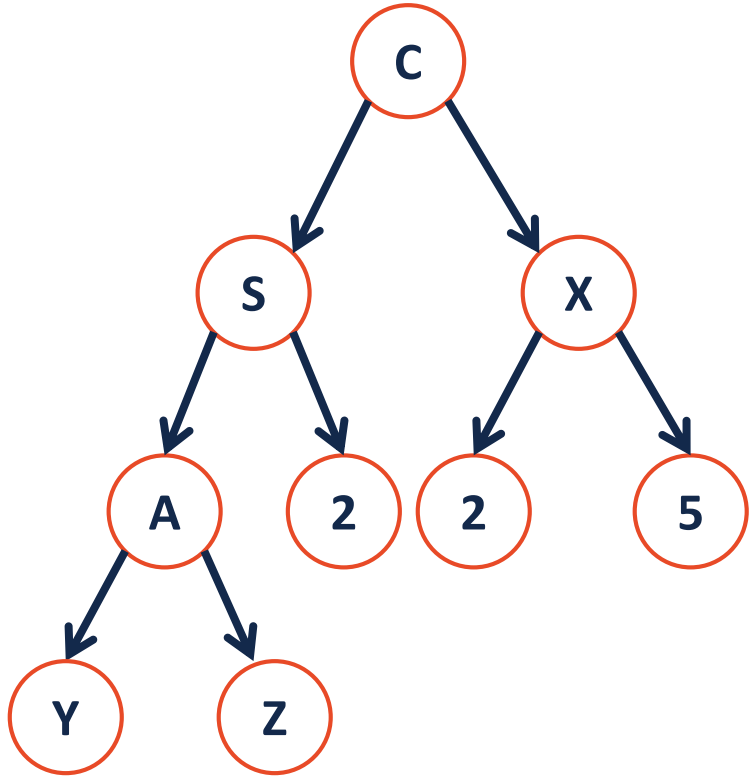
Is every **full** tree **complete**?

NO



If every **complete** tree **full**?

No



Tree ADT

insert, inserts an element to the tree.

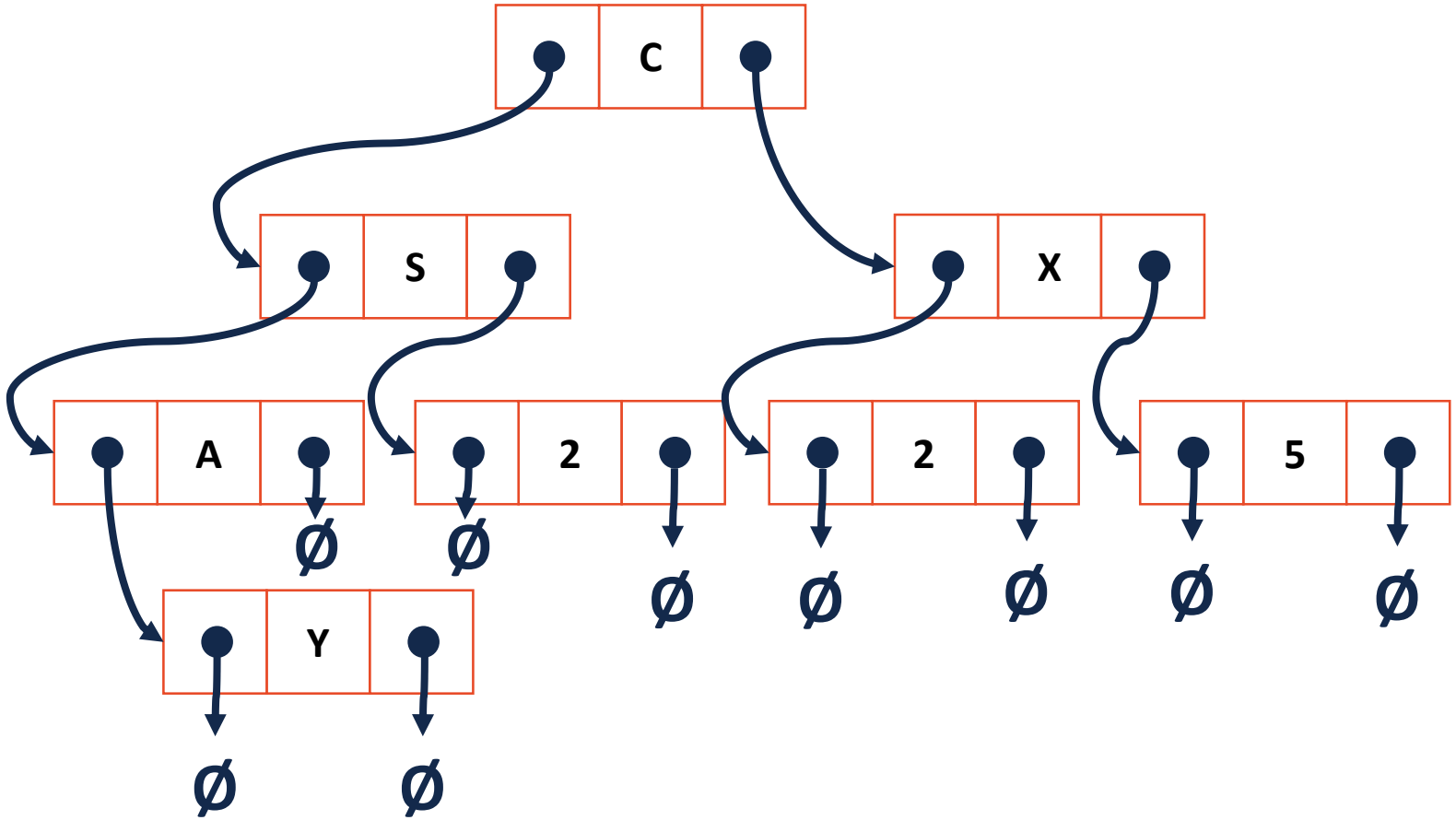
remove, removes an element from the tree.

traverse,

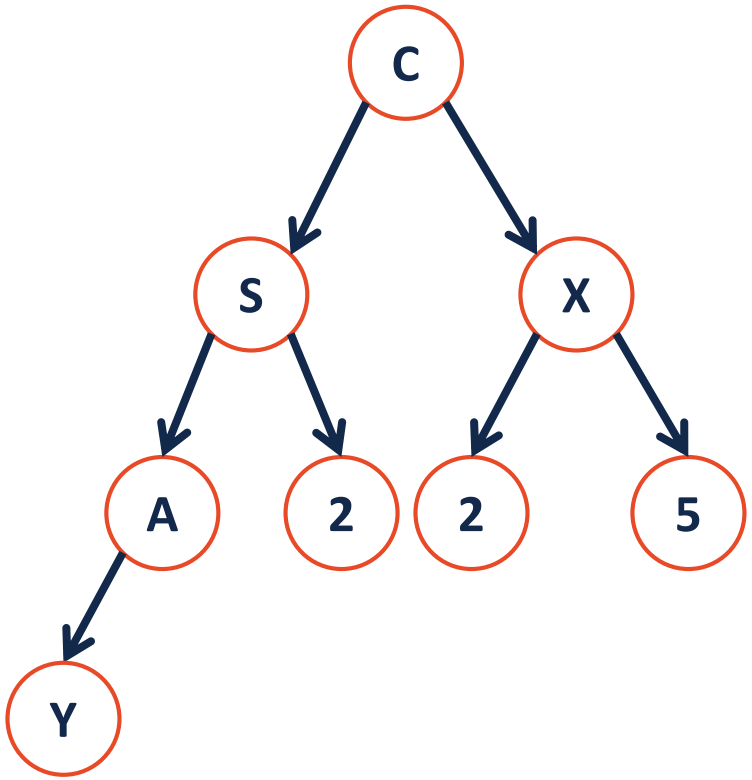
BinaryTree.h

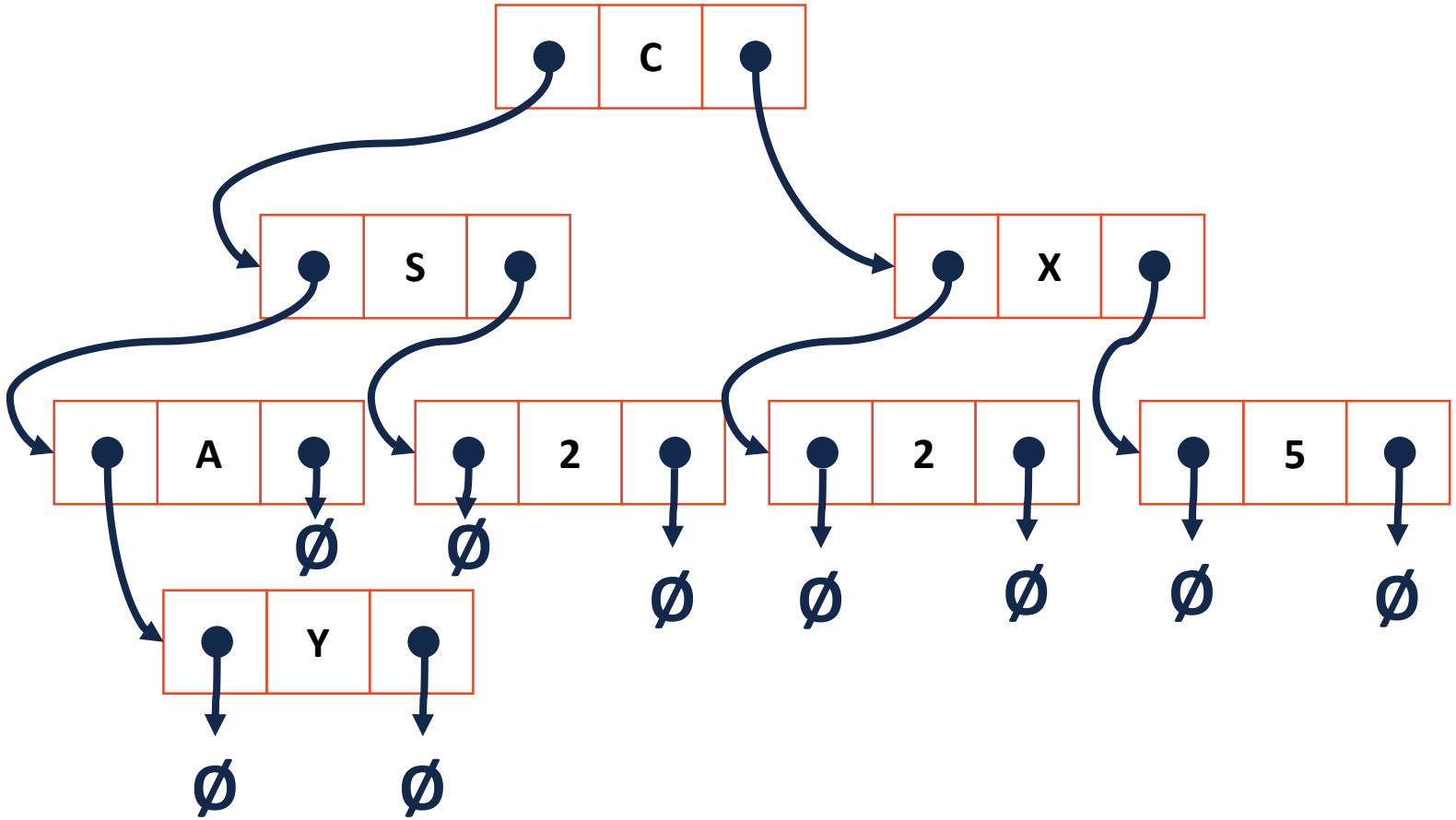
```
 1 #ifndef BINARYTREE_H
 2 #define BINARYTREE_H
 3
 4 template <class T>
 5 class BinaryTree {
 6   public:
 7     /* ... */
 8
 9   private:
10
11
12
13
14
15
16
17
18
19
20 };
21
22 #endif
```

Trees aren’t new:



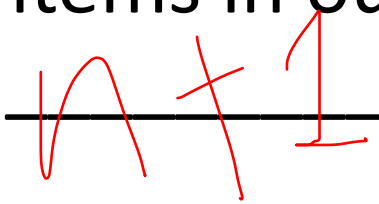
Trees aren’t new:





How many NULLs?

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



How many NULLs?

Base Cases:

n = 0:

n = 1:

n = 2:

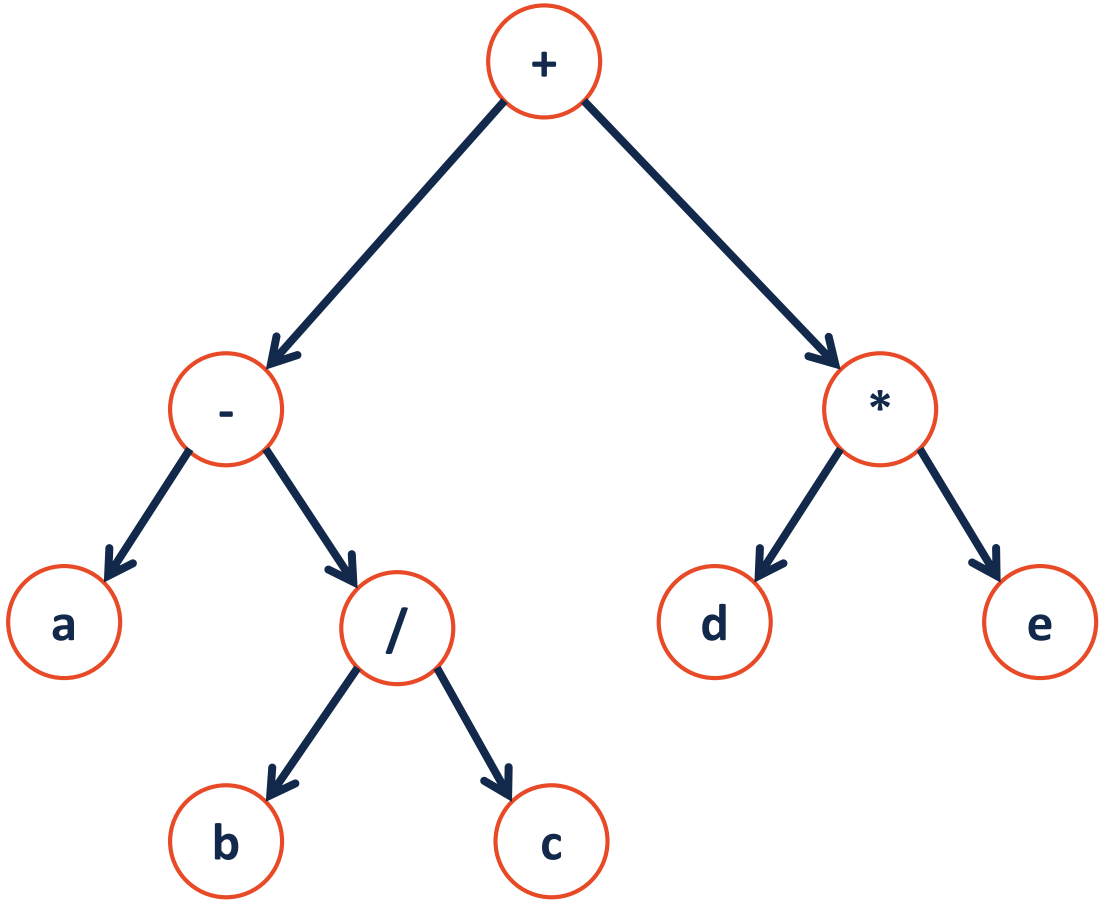
How many NULLs?

Induction Hypothesis:

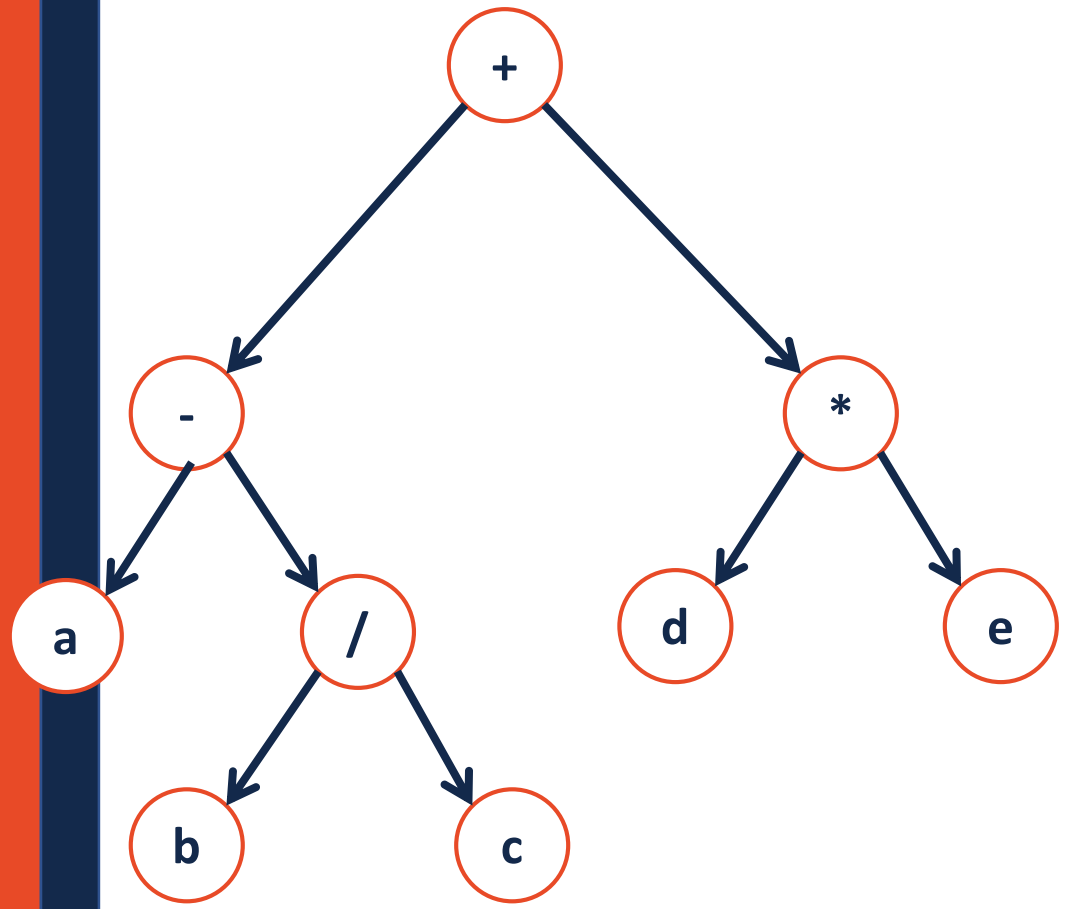
How many NULLs?

Consider an arbitrary tree **T** containing **n** data elements:

Traversals

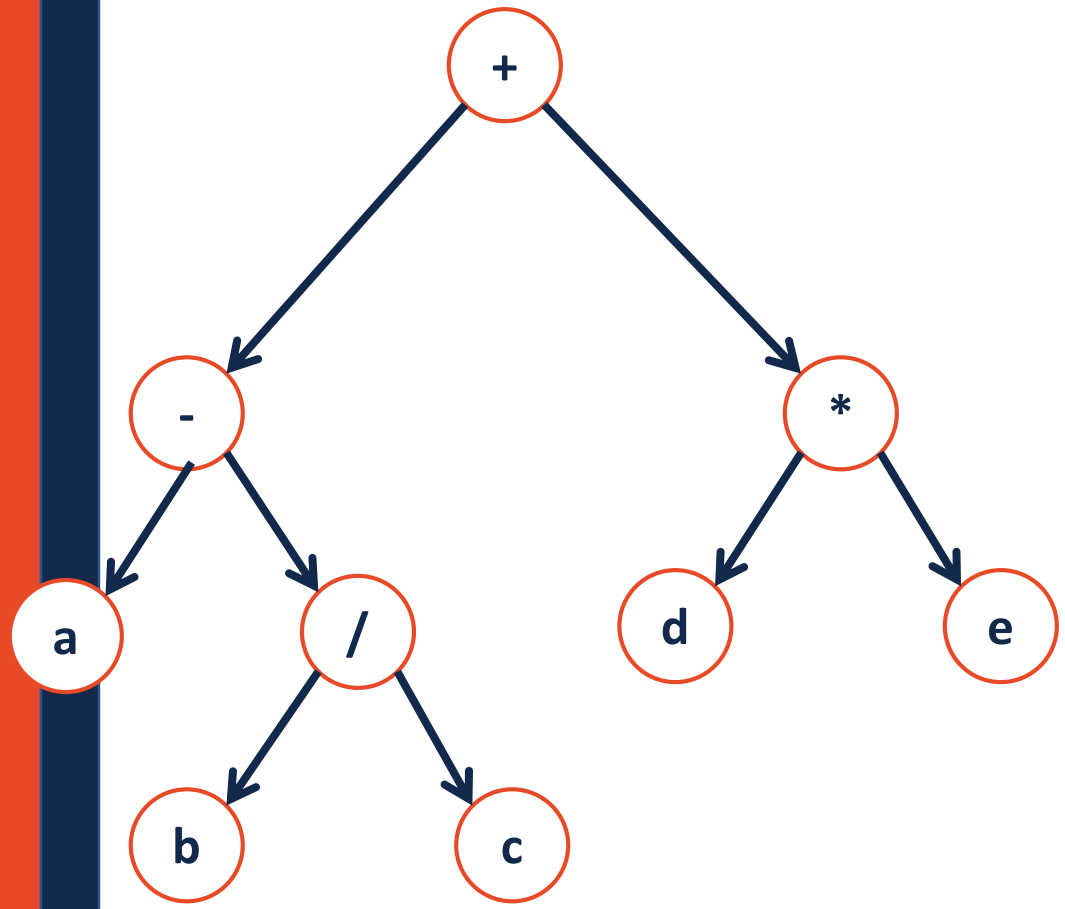


Traversals



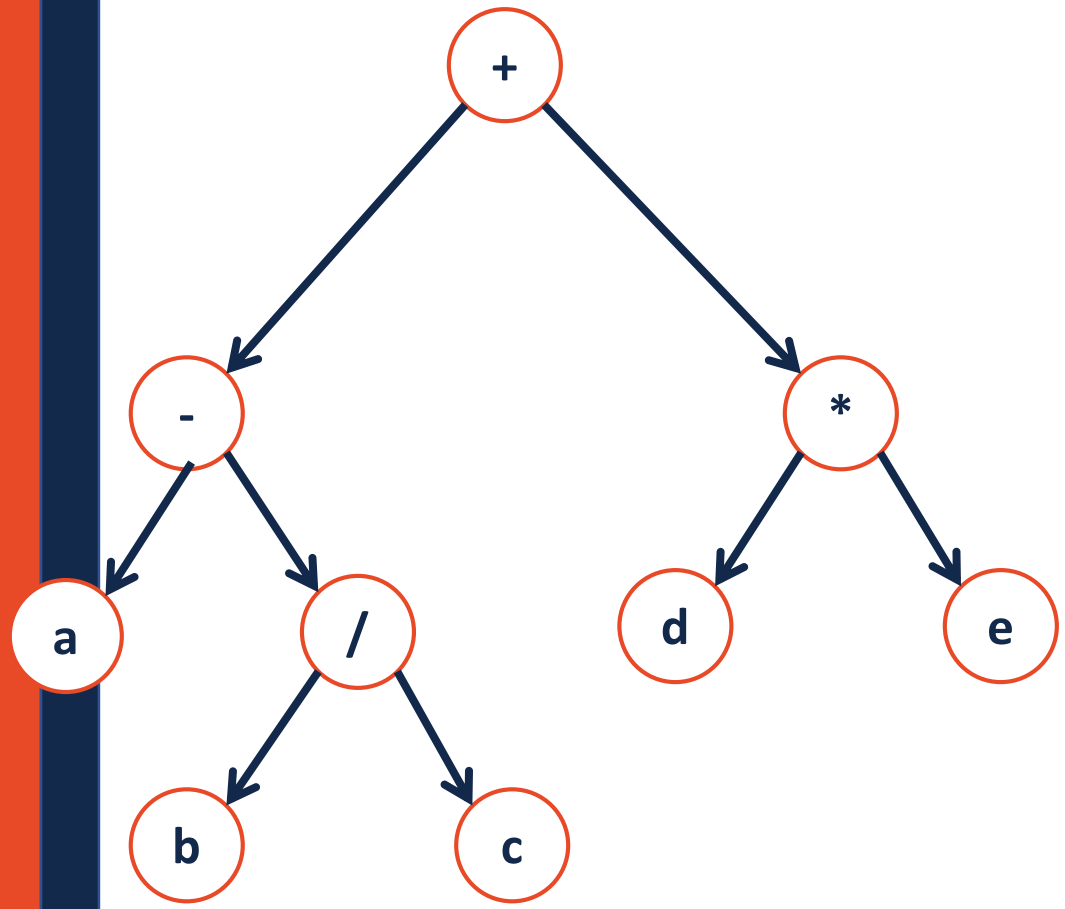

```
1  template<class T>
2  void BinaryTree<T>::__Order(TreeNode * root)
3  {
4     if (root != NULL) {
5
6        ____________________________;
7
8        ___Order(root->left);
9
10       ____________________________;
11
12       ___Order(root->right);
13
14       ____________________________;
15
16    }
17 }
```

Traversals




```
1  template<class T>
2  void BinaryTree<T>::__Order(TreeNode * root)
3  {
4     if (root != NULL) {
5
6        ____________________________;
7
8        ___Order(root->left);
9
10       ____________________________;
11
12       ___Order(root->right);
13
14       ____________________________;
15
16    }
17 }
```

Traversals




```
1  template<class T>
2  void BinaryTree<T>::__Order(TreeNode * root)
3  {
4     if (root != NULL) {
5
6        ____________________________;
7
8        ___Order(root->left);
9
10       ____________________________;
11
12       ___Order(root->right);
13
14       ____________________________;
15
16    }
17 }
```