



CS 225

Data Structures

Oct. 4 – Tree Proof

Tree Property: full

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

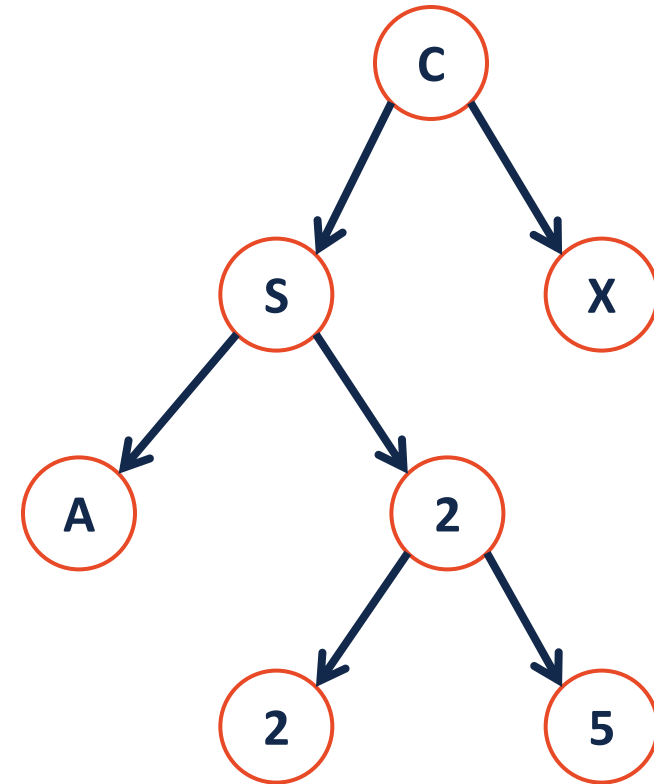
1. $F = \{\}$ *or*

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

where either:

T_L and T_R are empty *or*

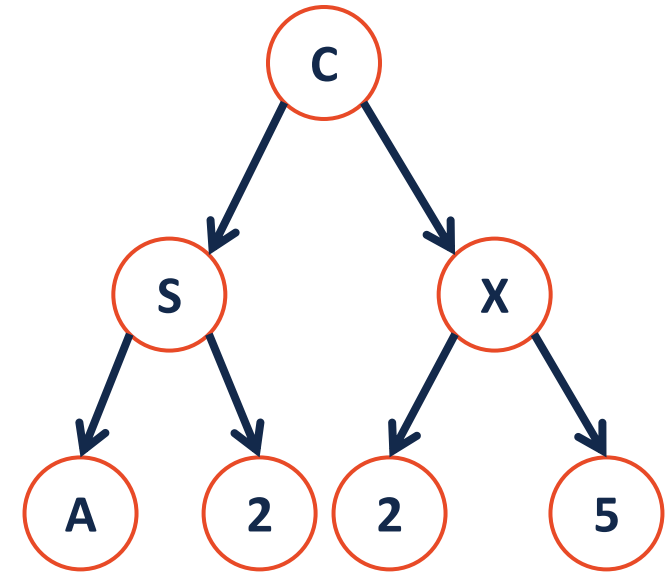
T_L and T_R are not empty



Tree Property: perfect

A **perfect** tree **P** for height **h** is:

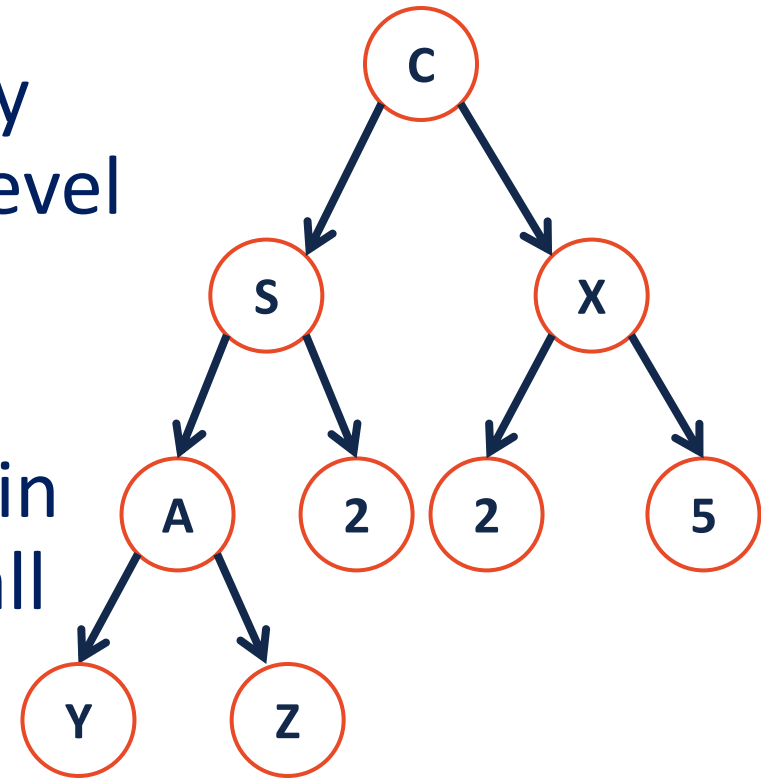
1. $P_{-1} = \{\}$ *or*
2. $P_h = \{r, T_L, T_R\}$ where:
 T_L and T_R are P_{h-1}



Tree Property: complete

Conceptually: A perfect tree for every level except the last, where the last level is “pushed to the left”.

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



Tree Property: complete

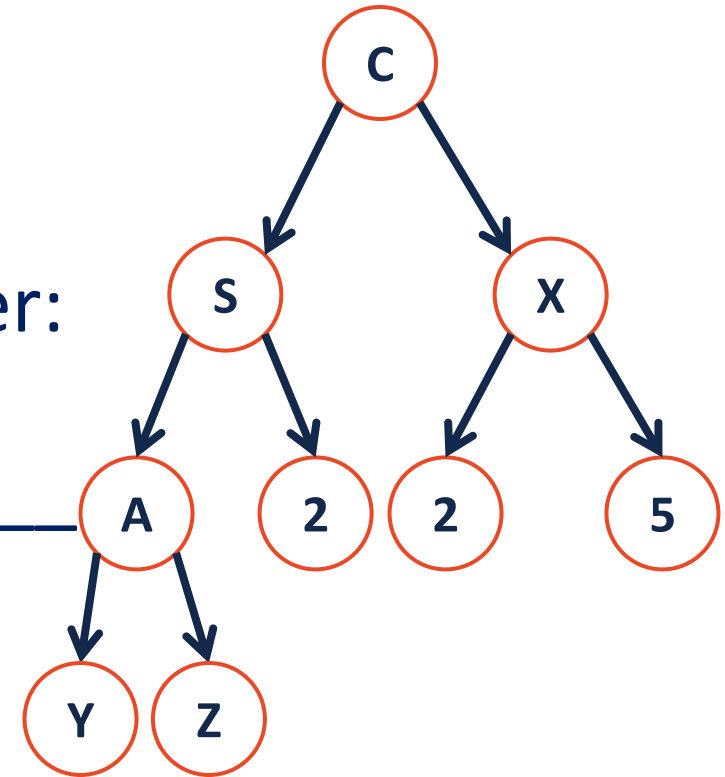
A **complete** tree C of height h , C_h :

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

T_L is _____ and T_R is _____

OR

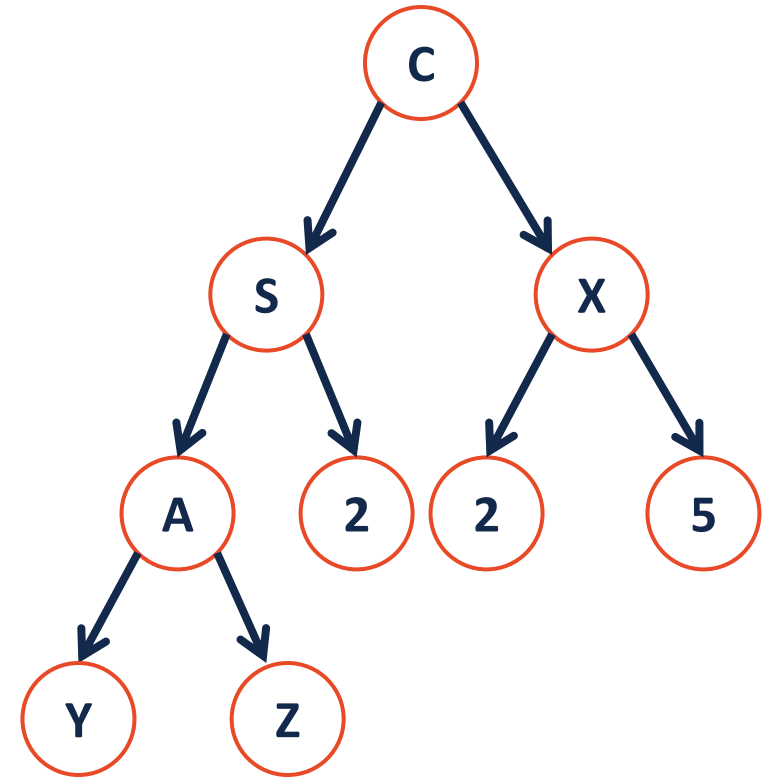
T_L is _____ and T_R is _____



Tree Property: complete

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

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



Tree ADT

insert, inserts an element to the tree.

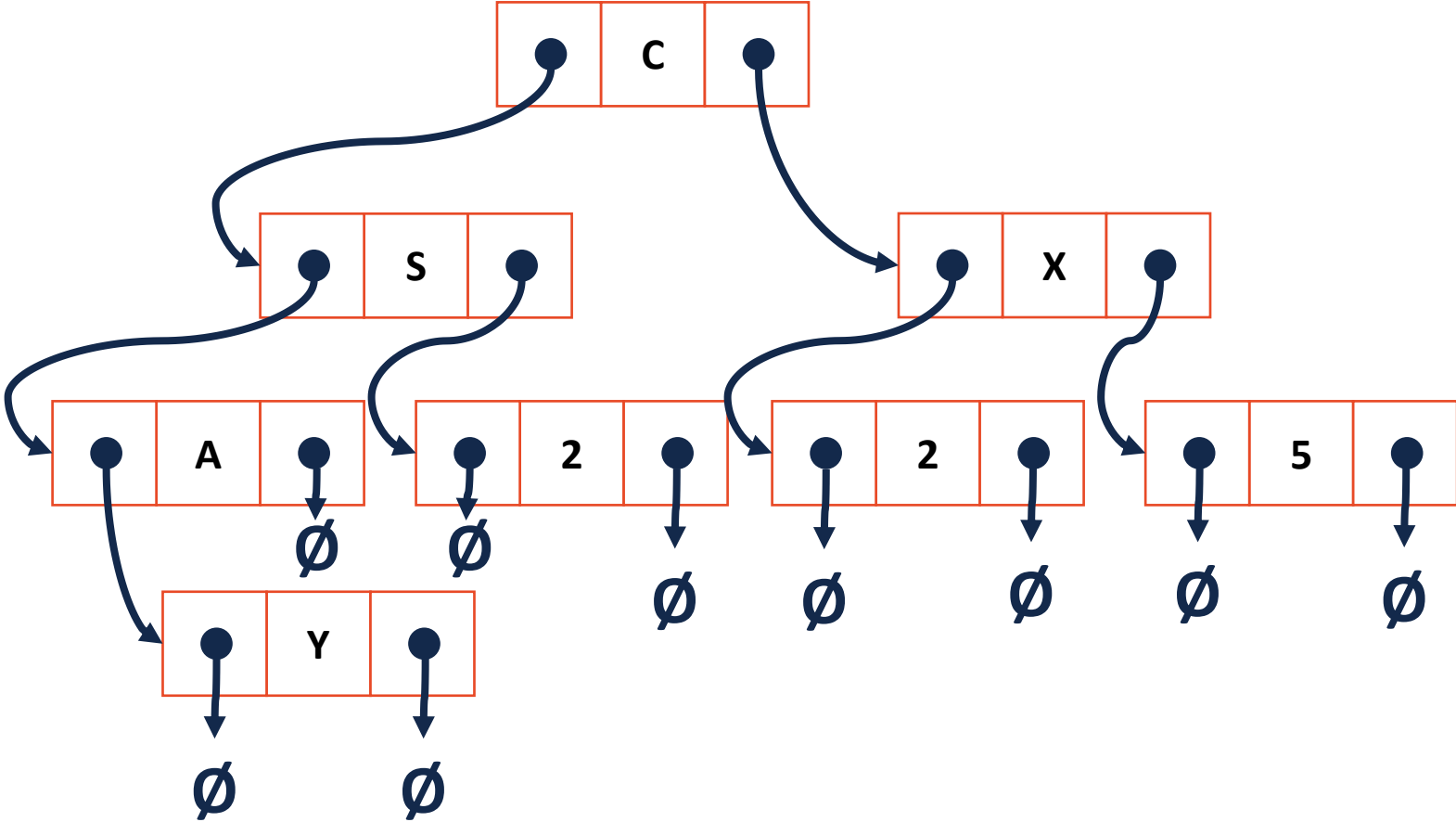
remove, removes an element from the tree.

traverse,

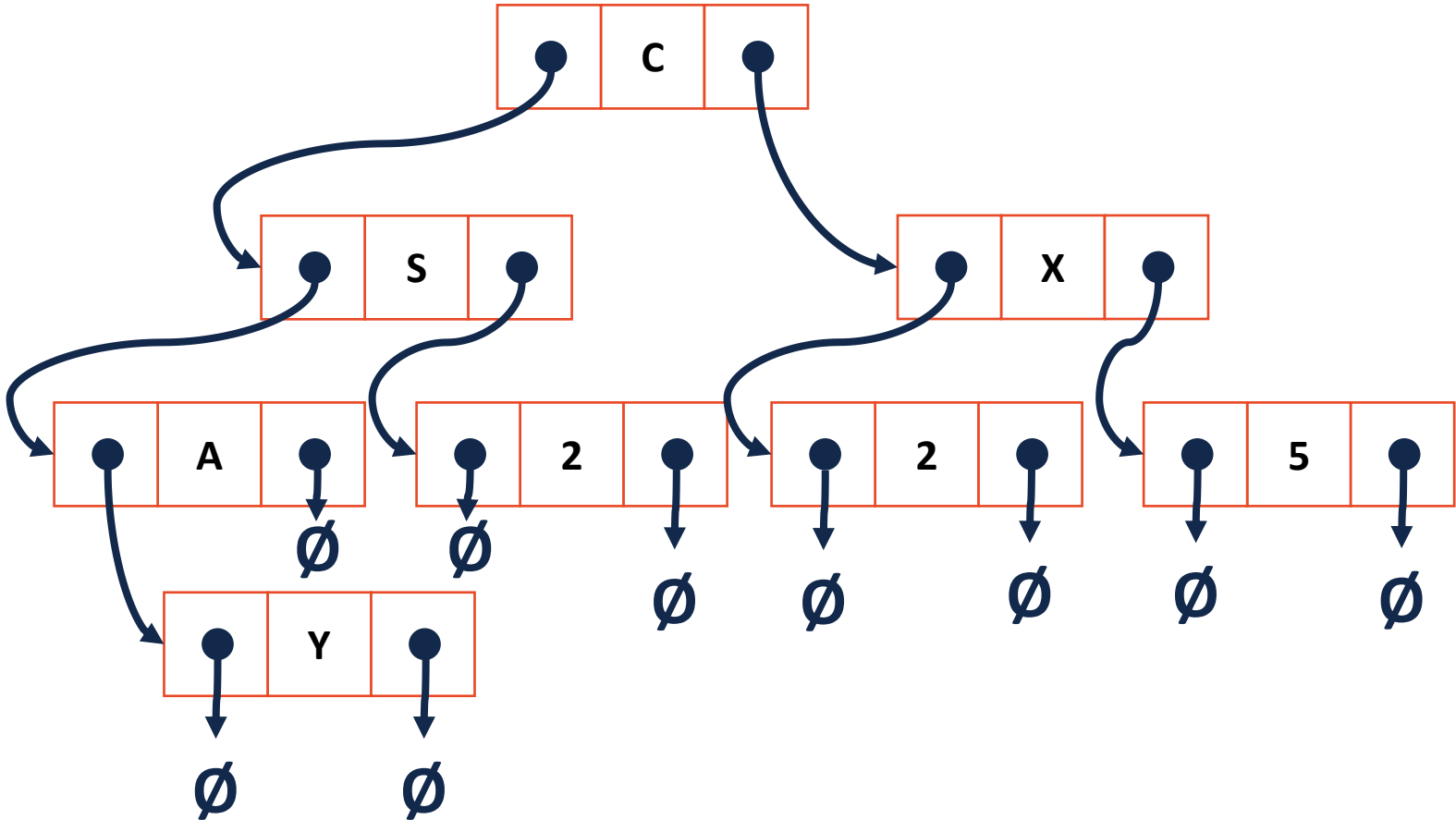
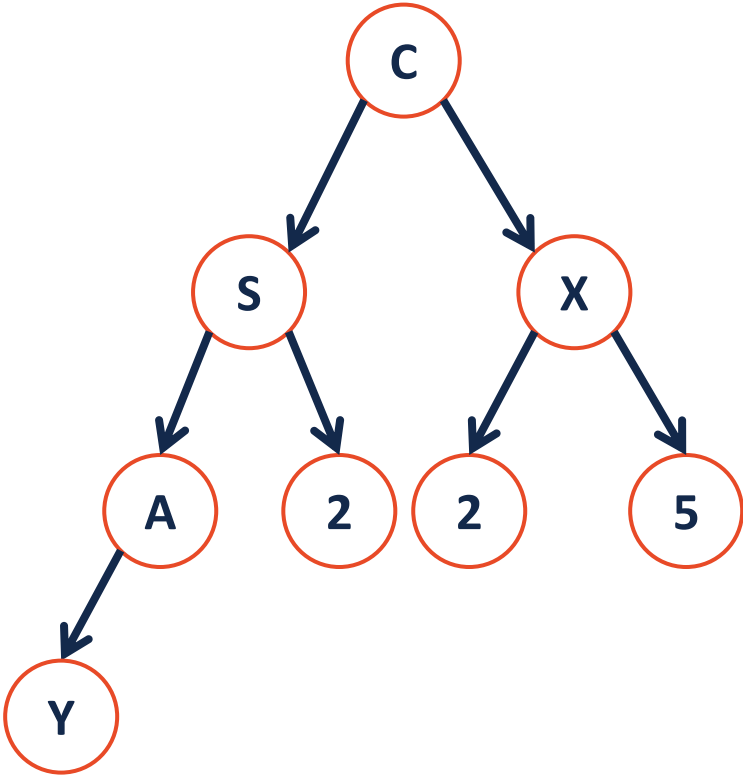
BinaryTree.h

```
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:



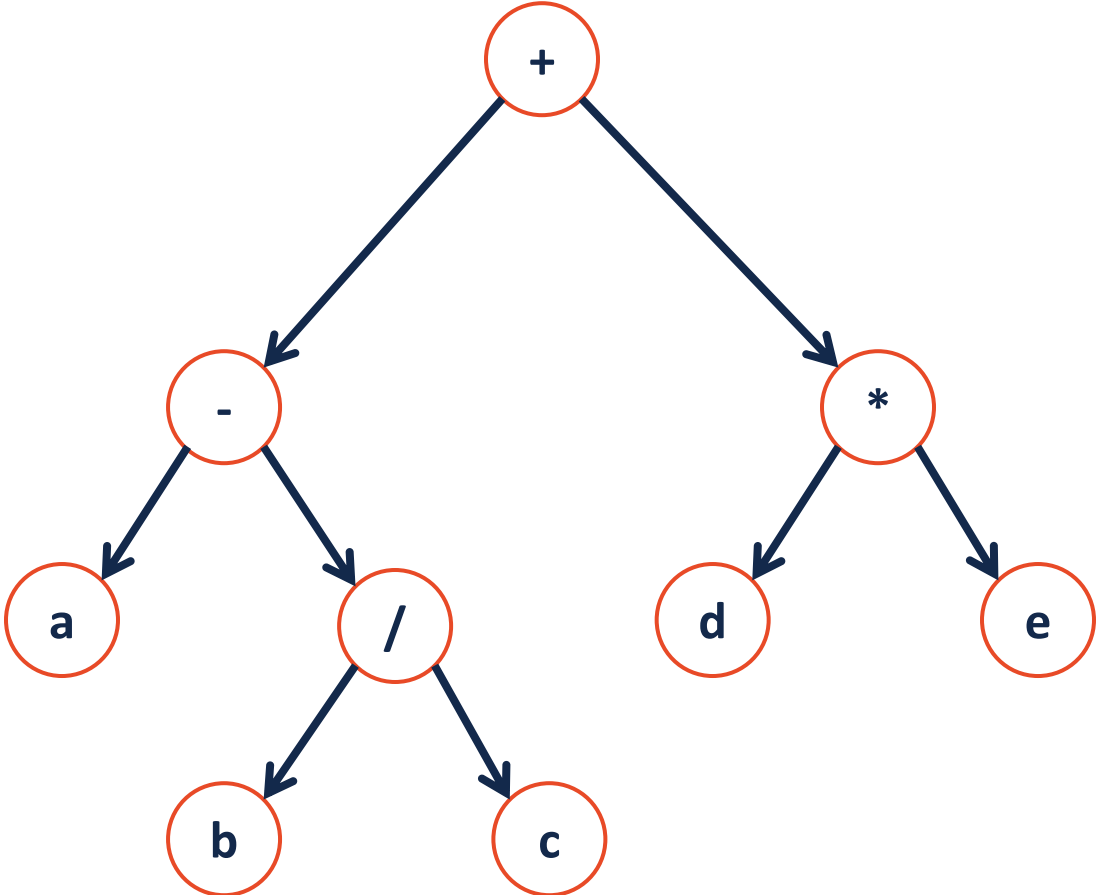
How many NULLs?

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

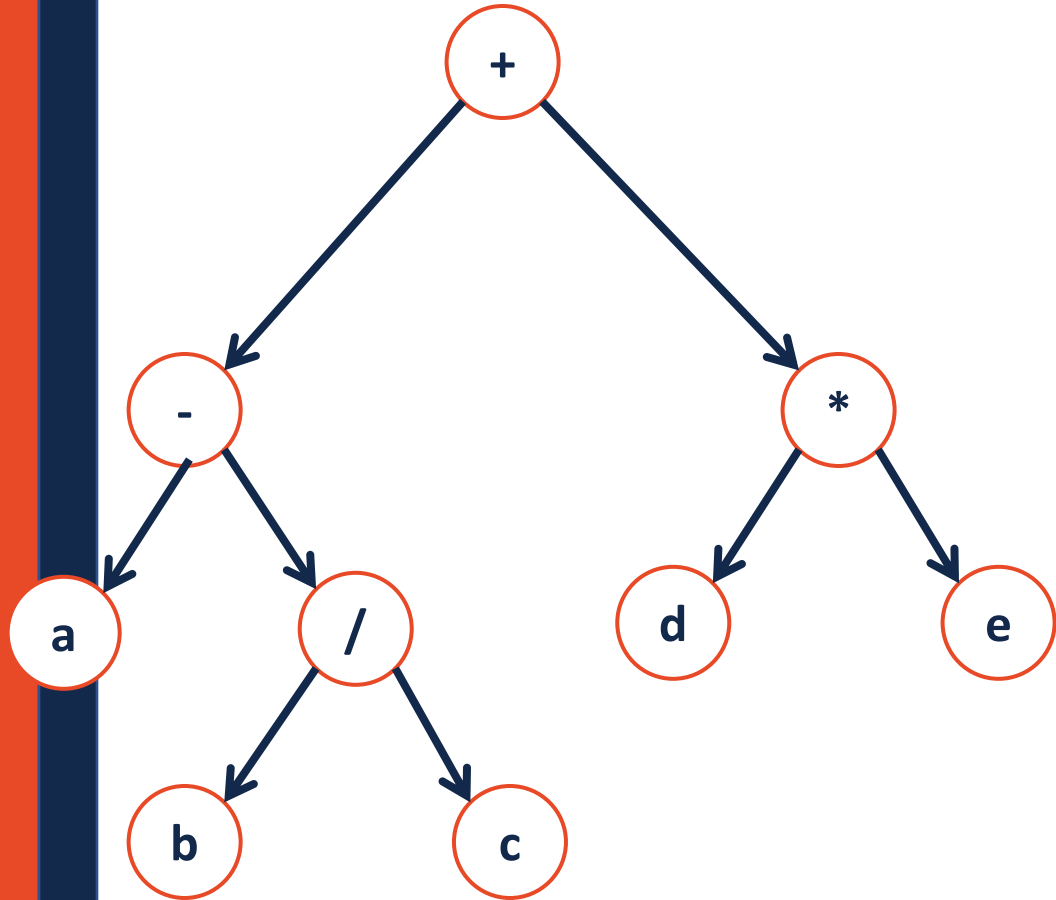
How many NULLs?

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

Traversals

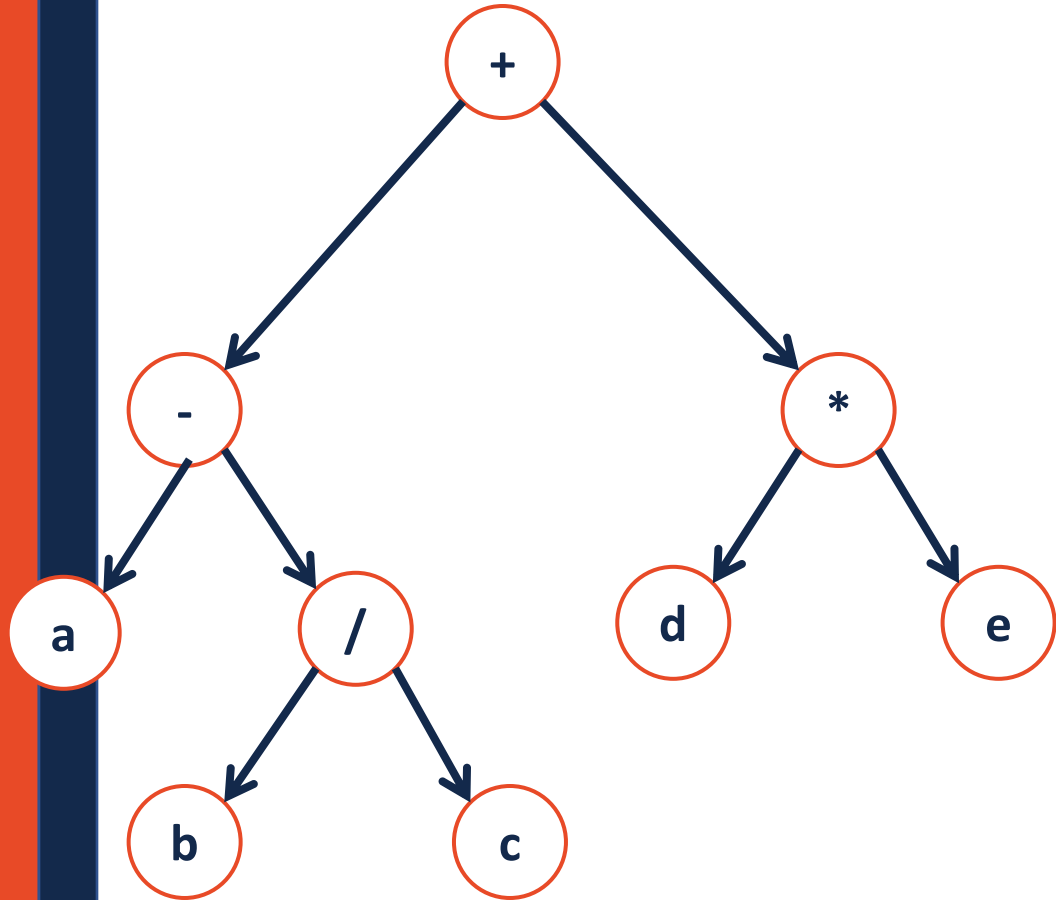


Traversals



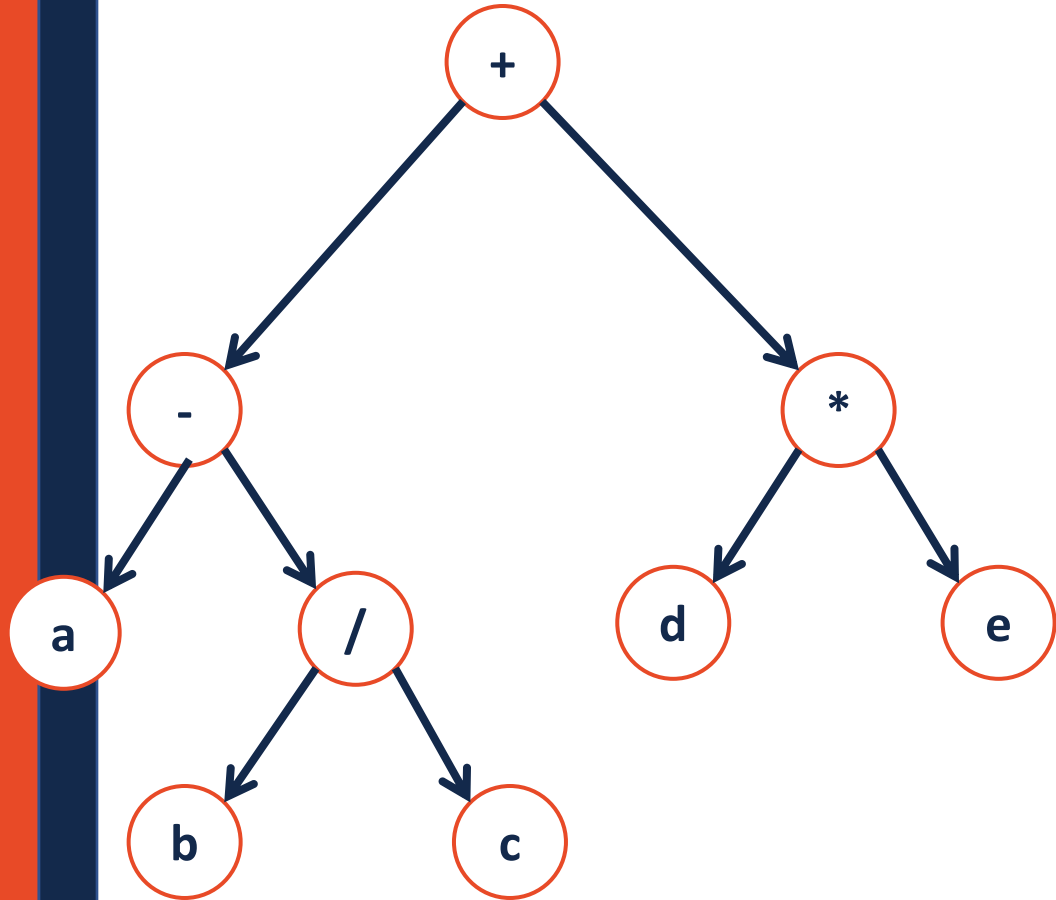
```
1  template<class T>
2  void BinaryTree<T>::__Order(TreeNode * root)
3  {
4      if (root != NULL) {
5          _____;
6          _____;
7          _____;
8          __Order(root->left);
9          _____;
10         _____;
11         _____;
12         __Order(root->right);
13         _____;
14  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


CS 225 – Things To Be Doing

Exam 4 (Programming/MP2) currently ongoing!

More Info: <https://courses.engr.illinois.edu/cs225/fa2017/exams/>

MP3: Available now!

Due: Monday, Oct. 9 at 11:59pm

Lab: lab_tree

Implement your first binary tree functions!

POTD

Every Monday-Friday – *Worth +1 Extra Credit /problem (up to +40 total)*