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
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,**
#ifndef BINARYTREE_H
#define BINARYTREE_H

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

};

#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

```
template<class T>
void BinaryTree<T>::__Order(TreeNode * root)
{
    if (root != NULL) {
        ______________________;
        ___Order(root->left);
        ______________________;
        ___Order(root->right);
        ______________________;
    }
}
```
Traversals

```cpp
template<class T>
void BinaryTree<T>::__Order(TreeNode * root) {
    if (root != NULL) {
        ____________________;
        __Order(root->left);
        ____________________;
        __Order(root->right);
        ____________________;
    }
}
```
Traversals

```cpp
template<class T>
void BinaryTree<T>::__Order(TreeNode * root) {
    if (root != NULL) {
        ____________________;
        __Order(root->left);
        ____________________;
        __Order(root->right);
        ____________________;
    }
}
```
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)