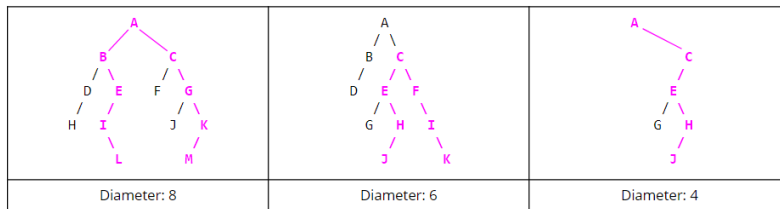


For this lab, please implement your solution to this problem in the `diameter.cpp` starter file and upload to Gradescope. This is to prepare you for the structure of the final exam, which will involve online code submissions. You only need to turn your code in, and it will still be graded on effort.

Name: _____ Uniqname: _____ ID #: _____

Lab 8 Handwritten Problem: Tree Diameter [5 Points]

The diameter of a tree is the maximum number of edges on any path connecting two nodes of the tree. For example, here are three sample trees and their diameters. In each case, the longest path is bolded and shown in purple. Note that there may be more than one longest path.



```
struct BinaryTreeNode {
    BinaryTreeNode* left;
    BinaryTreeNode* right;
    int value;
    BinaryTreeNode(int n)
        : value{n}, left{nullptr},
          right{nullptr} { }
};
```

Implement the `diameter()` function, which computes the diameter of a binary tree represented by a pointer to an object of the `BinaryTreeNode` class. Assume that `nullptr` represents an empty tree or a missing child. You may write helper functions.

Implementation: Your solution should run in $O(n^2)$ time, where n is the number of nodes in the binary tree. However, there is a way to solve this in $O(n)$ time ... try to see if you can figure it out!

```
int diameter(const BinaryTreeNode* tree) {
```