

CSI 33 Section E01
Fall 2017
Due: Wednesday, November 29, 2017

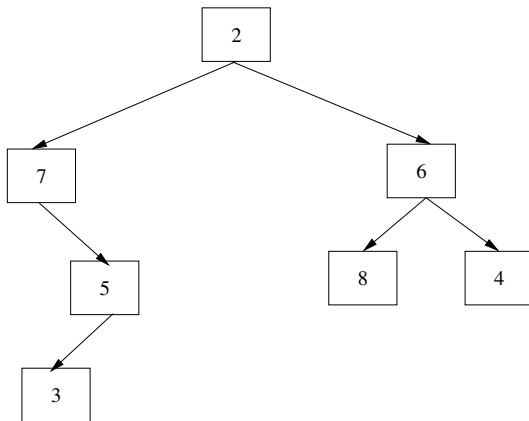
Project 5
November 22, 2017

Programming Project Number 5: Modifying the TestTreeNode Program

This is a clarification of exercise 2 included in the C++ Supplement on the `TreeNode` class, handed out for November 13, 2017. The program builds a binary tree, as a set of `TreeNode` objects having one names as the root of the tree. It then traverses the tree using the three main traversal algorithms for binary trees.

It then prints a version of the tree represented as an array, using `NULL` values to designate empty subtrees for the leaf nodes.

1. First, create a project including the files `TreeNode.h`, `TreeNode.cpp`, and `TestTreeNode.cpp`.
2. Next, modify the program to generate a different binary tree, the one which is illustrated in the first figure of the handout showing a tree of integers:



3. Make sure the traversals still show every node of the new tree.
4. Generate the array representation of the new binary tree. The problem here is to allocate an array large enough to hold all the empty subtrees of every leaf node, as `NULL` values. The size of the array must be twice as big as the size of the number of nodes that would be in a full binary tree with the same height as the tree you are representing. For example, if your tree has height 4, your array must have $2^{(4 + 2)}$, or 64 positions. This is one of the problems with array representation of binary trees—all the empty subtrees must be included.