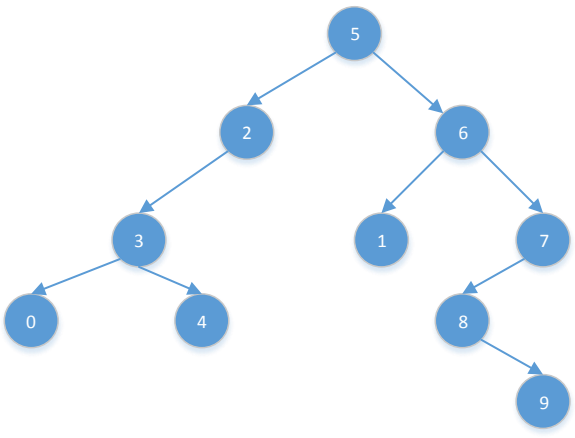
CS 2413 Data Structures – Spring 2016 – Programming Project 5
Due April 10, 2016 – 11:59 PM

Objectives

1. [50 Points] Create all the methods and fields to implement the parent array implementation of the binary tree data structure. **You are required to demonstrate the working of the following methods for each of the methods by invoking them from a main program.**
    a. empty constructor
    b. non-empty constructor (sets the initial size of the arrays)
    c. destructor
    d. copy constructor
    e. overloaded equal to operator
    f. ostream operator – pre order traversal
    g. size method
    h. height method
    i. getLeft method
    j. getRight method
    k. preorder traversal method
    l. inorder traversal method
    m. postorder traversal method
2. [20 Points] Read the redirected input and create all the data structures.
3. [20 Points] Demonstrate the working of all the methods.
4. [10 Points] Document your project thoroughly as the examples in the textbook. This includes but not limited to header comments for all classes/methods, explanatory comments for each section of code, meaningful variable and method names, and consistent indentation.

Project Description

Consider a binary tree with N nodes where each the nodes are given unique numbers in the range 0..N-1. Such a binary tree can be stored in in an array P of size N, wherein P[k] will indicate the parent of node numbered k in the binary tree. See the binary tree and the corresponding parent array below.



Parent Array P

0	1	2	3	4	5	6	7	8	9
3	6	5	2	3	-1	5	6	7	8

Since 5 does not have a parent with placed a -1.

To recognize if a node is left (0) or right child (1) we have,

0	1	2	3	4	5	6	7	8	9
0	0	0	0	1	-1	1	1	0	1

Examples: Node 0 is a left child of node 3 hence a 0.
Node 6 is a right child of node 5 hence a 1.

The input to your project will be as follows: The first line of input will be number of nodes in the binary tree. If a node does not have a left or right child we will place a -1 otherwise, we will put the number of the left or right child. For the above tree, the input is given below.

10
5 2 6
2 3 -1
3 0 4
6 1 7
1 -1 -1
0 -1 -1
8 -1 9
9 -1 -1
4 -1 -1
7 8 -1

Class Structures

You are required to implement the following class structure along with the implementation of the methods associated with each of them.

```
template <class DT>
class ParentBinaryTree {
        protected:
                DT* ParentArray;
                int   numNodes;
        public:
                //All the required methods
};
```

Constraints

1. In this project, the only header you will use is #include <iostream>.
2. None of the projects is a group project. Consulting with other members of this class on programming projects is strictly not allowed and plagiarism charges will be imposed on students who do not follow this.