Computer Science 2210

**Project 5 – B-Trees**

# Purpose

The purpose of this assignment is give you experience with a **B-Tree**. In this assignment, you will build **B-Trees**, populate them, display them, and collect some statistics about retrieval characteristics.

# Specifications

Create a base class for **Nodes** with derived classes for **Index** nodes and **Leaf** nodes. The proper use of **inheritance** and **subtype** **polymorphism** makes some of the code in the **BTree** class simpler. Other data structures such as **Lists** may be useful in the **Node** class and its descendants.

Create a **BTree** class that permits adding values, displaying the tree, gathering statistics, and so forth. As part of the effort to add values, it will be necessary to provide for splitting **Leaf** and **Index** nodes properly. It is not necessary to provide for the deletion of single values – thus this is not a complete implementation of a fully functional **BTree**. This class should use recursive methods where that simplifies the code. When displaying the tree use an analog to the **Pre-Order** traversal – display the parent before the child. Other data structures such as **stacks** may be useful in this implementation.

Write a driver program that builds a **BTree** populated initially with 500 unique random integers (from a uniform distribution) between 0 and 9999. At each level, the “**Nodes**” should have room for **N** values. When a **Node** is full and an additional item needs to be added to that **Node**, the split should divide the **Node** into two separate **Nodes** each having half {**(N+1) / 2**} of the slots filled.

The **BTree**’s requirement for node value uniqueness means that you may have to try to insert more than 500 values before you have 500 successes. Test the program with at least (the number of values per **node**) **N** = 3, **N** = 7, **N** = 11, and **N** = 25.

Provide a menu in the console driver or appropriate controls in a **GUI** application that will allow the user to do the following:

* Set the “arity” of a new tree and create the tree
* Populate the tree with 500 unique random integers from a uniform distribution
* Display the tree in some reasonable manner – being sure to label the output appropriately (it is probably not possible to display the entire tree with 500 or more values on the screen at one time – in a console application). Include statistics about the tree (its depth, percent filled, number of index nodes, number of leaf nodes, and so forth) in your display
* Add a new user-specified integer value to the **BTree**
* Find a value and display its entire node if it is present; display an appropriate message it is not present. Display the nodes visited in the search for the leaf containing the value being sought.

# Deliverables

The same types of deliverables as for previous assignments should be submitted for this assignment. You may work in teams of two or three if you wish. Submit the final zipped folder named according to the **Course** **Facts** document, including each team member’s name in the submitted zipped file’s name. Make sure to include the TA on the submission.

# Class Diagram of One Possible Solution

The following class diagram shows one possible design for a console application solution. Your solution need not match this one exactly. Indeed, you may use a Windows Forms application instead.

