Group Leader: Elizabeth McClellan

Group Tester: Drew Aaron

Group Requirement Leader: Andrew Hamilton

Group Documenter: Michael Beaver

Course: CS 355

Semester: Fall 2012

Date: September 24, 2012

**Group Meeting Minutes**

Members in attendance: Drew Aaron, Michael Beaver, Andrew Hamilton, and Elizabeth McClellan

Time met: 4:00pm to 5:15pm

Agenda: Determine how to print a Binary Search Tree; develop Big-O Presentation

Group members discussed different methods for printing a Binary Search Tree. Initially, the thought was that the BST needed to be printed to visually resemble a BST graph. This notion led to the idea of printing the BST line-by-line, due to the limitations of outputting data to the console.

Michael suggested pushing BNode data from each level of the BST into a dynamic buffer. When all the siblings are the buffer, it would be output to the console. The process would repeat until the entire tree is printed. Of course, this solution could prove cumbersome with large BSTs. Furthermore, implementing the logistics of when to print connections (i.e., slashes “\” and “/”) would prove to be rather frustrating. Therefore, this approach was abandoned.

It was realized that the BST did not have to be visualized graphically but as a linear list. This led to a need to solve the Print In-Order problem. The group determined the logic necessary to solve the problem: Recursively traverse left down the sub-tree, process the root BNode, and then recursively traverse right down the sub-tree until all BNodes are processed. The logic was simple enough, but the attempted implementations started growing unwieldy and progressively worse. Then the group discovered a simple solution by Dr. David Eck that perfectly matched the described logic. Hence, the group adapted and adjusted Dr. Eck’s implementation. Once the Print In-Order problem was solved, the Print Pre-Order and Print Post-Order problems were easy to solve because all the printing problems are similar.

The meeting concluded with the group discussing and developing its presentation on Big-O. An online graphic program was used to generate example graphs. These graphs were used to demonstrate that lower order terms and constant values may be disregarded when using Big-O notation.