**Algorithms and Collections**

**Coursework Part 2**

Comparisons Required to Search for an Item – BST (Successful)

Comparisons Required to Search for an Item – BST (Unsuccessful)

Comparisons Required to Search for an Item – TreeSet (Successful)

Comparisons Required to Search for an Item – TreeSet (Unsuccessful)

Comparisons Required to Search for an Item – HashSet (Successful)

Comparisons Required to Search for an Item – HashSet (Unsuccessful)

Comparisons Required to Search for an Item – BST added In Order

Adding/Removing Items Before Search

Serialization

Changing the BinarySearchTree class to make it serializable and to then serialize and deserialize it required multiple changes.

The first changes were made to the BinarySearchTree class itself. A *serialVersionUID* was added, a **static final long** used during deserialization to confirm that the receiver is getting an object compatible with serialization from the same class as the object the sender serialized. Then, a private writeObject method and readObject method were defined so that the program knows how to serialize and deserialize the BinarySearchTree.

Afterwards, the Item class was made Serializable also, as well as having its own *serialVersionUID* added. Lastly the code to test the serialization and deserialization were added to the main function in the BSTAndSetsTest class, contained within a try-catch to account for exceptions.

Critical Appraisal

Acknowledgments

Slides accompanying Collin’s book covering TreeSets

Watt and Brown’s lecture on Hashing