Eric Gustin

CPSC223-01 HW09

11/19/19

***Discussion and Analysis***

*Testing Strategy & Possible Improvements*

My testing strategy consisted of first writing down on paper every single case that I could think of. This included removing a leaf node, removing the root node, removing a node that has a single left child, removing a node that has a single right child, and removing a node with two children. Removing a node with two nodes required an interesting testing strategy because I had to account for the case were the right child of the “node to-be removed” only has a right child and where it has a left child. Interestingly, this affects how the in-order successor is chosen. If the right child of the “node to-be removed” doesn’t have a left child, then it will be the in-order successor, otherwise, you need to iterate through the BST continuing to go left until a leaf is reached, and that will be the in-order successor.

*Implementation Issues/Challenges and How I Addressed Them*

I initially had difficulty understanding how the recursion was working in the remove function which resulted in a tcache error where I was deleting the same node more than once. Eventually I was able to understand how the recursion worked during the backstepping and was able to implement the remove function correctly.