

Sonya Carmona

- **What is the time complexity of each method (corresponding to a command) in your implementation? Reflect on the worst-case time complexity represented in Big O notation.**

My remove ID method time complexity would be $O(n)$ because it uses one while loop with only multiple if and else statements to delete and connect nodes to make the AVL tree. My remove Inorder n is the same as my remove ID because it realizes the logic and uses its implementation. My insert method time complexity would be O^2 this is because of having nested loops and calling other classes to perform the balance and rotation when adding a new node into the tree. My search ID method time complexity would be $\log(n)$ because the source code would be $O(n)$ but in the main, it uses a for loop for the vector that makes it $\log(n)$. The search name would be the same time complexity as my search Id because it depends on it and uses the same logic. My print order, preorder, and postorder would have a time complexity of $\log(n)$ it iterates through the tree and doesn't stop until it gathers all the nodes in the tree. My print level count has a time complexity of $\log(n)$ it iterates through the tree similar to my print in order to gather the height and return it.

- **What did you learn from this assignment and what would you do differently if you had to start over? [5 points]**

I learned that I need to implement more object-oriented programming. I need to make sure that my programming is readable and understandable. If I implement this I will be able to digest my work more and fix bugs easier. For this project I had trouble going back and understanding and knowing the area where my code went wrong making the debugging process longer than it needed to be.