Matthew Bartrum

CS260

Binary Trees has such a nice setup in programming. During the past weeks I enjoyed playing with different structures and seeing how they work with data. I enjoyed the less steps that are needed to remove and insert an item from the tree. This is because all the work that is done is handled in the structure. Building the tree was not as hard as I thought it was going to be. Setting up the routes for the data was much easier to build compared to the last two structures. I like the search method in a node on how it can reduce the time by either going right or left until the item is found. I would like to see a program structure like this work in a video game. Especially comparing the top scores or how they can get you in a ranked match. I am going to try to create this structure again in python language and see what the difference would be.

Pseudo code

**Search Binary tree**

While not finding a nullptr

If (bid is 0)

Return the current bid

If (bid is greater than 0)

Bid is going to the left

If(bid is less than 0 )

Bid is going to the right

**Search add node**

If (bid is greater than 0){

If ( left node is null){

Node left is bid

Else (add node to the left)

else ( right node is null){

Node right is bid

Else (add node to the right)

**Remove Node**

If (bid is greater than 0){

Remove node on left}

Else If (bid is greater than 0){

Remove node on right

Else

If (left and right is nothing)

Delete node

Else if (one child is on the left)

Delete left

Else if (one child is on the right)

Delete right

Else

Delete both left and right node