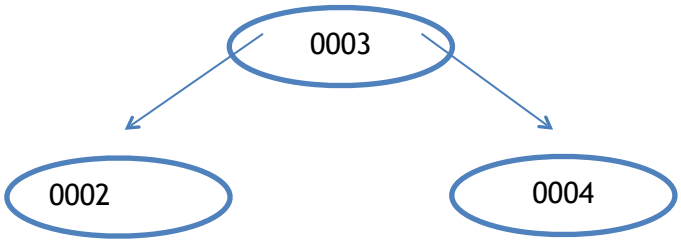
Project Documentation - Rough Draft

Leopold, Ruth
Orsini, Seraphina
Picard, Anthony
Porter, Ethan

For our Project we decided to build an educational tool that can help users understand inserting and deleting items into a particular data structure. These data structures include AVL Trees, Red Black Trees, and Binary Heaps. This project will consist of a Graphical Interface that allows the user to select which tree they would like to modify. They will then be brought to a window that has a control box and a canvas on which the visualization of the tree will be drawn.

In the control box there will be:

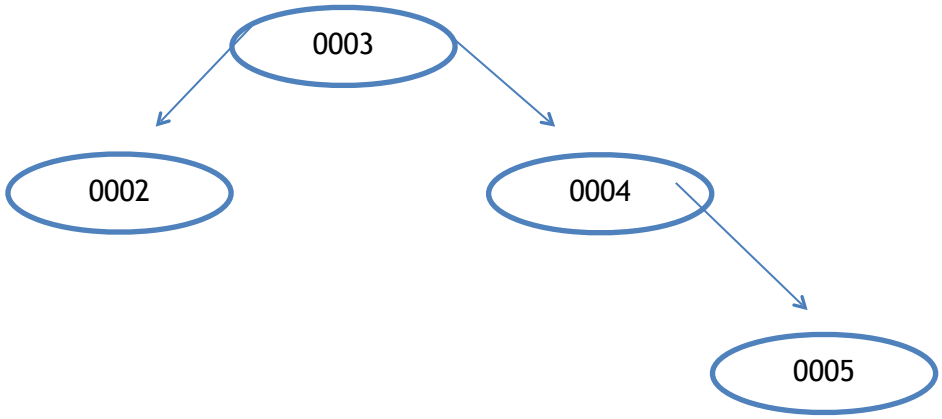
- A drop down menu that allows for the user to select either Delete or Insert.
- A text field that will allow the user to enter an integer that will be inserted into or deleted from the data structure this action depends on what is selected in the drop down menu above.
- A "GO" button this button will take the value of the text field and do the action selected in the drop down menu
- A text Area that displays a list of actions that are performed to either insert or delete an item. For Example if you are going to insert 0005 into a AVL tree that looks like this:



The list of actions would read:

1. Compare 0005 to the root 0003.
2. 0003< 0005 check right child
3. 0004<0005 check right child
4. There is no right child insert 0005 as the right child of 0004
5. Check for rotations
6. No rotations needed insertion complete

This is the resulting tree that is displayed in the canvas see another sample in mockup of the system.



Description of each data structure in or program:

- AVL Tree will do the same as a Binary search tree only it will use balancing conditions For ever node in the tree, the heights of its left sub-tree and right sub-tree differ by no more than 1. If this balancing condition is violated the tree is corrected using a number of single or double rotations until the tree is returned to a balanced state.
- Red Black trees are based on a Binary Search tree and have the following properties
    - Every node is either red or black
    - All null leaves are colored black
    - A red node cannot have red children
    - Every path from node to null nodes must contain the same number of black nodes

  This means the system will correctly insert a node and change it color according to what its parent node is the system will also perform rotations to ensure that all the properties are satisfied.
- For a Binary heap with the min-heap property the value of each nod is greater than or equal to the value of its parent, this means the min value is in the root.

Conclusion

After completing the project the group decided that two text areas would be better than one. We decided this based on how a user would prefer to search through all their previous steps. Instead of each rotation being listed with their descriptions we are going to only list the rotation in the first text area and move the description of the rotations into another text area. This text area will list all the properties of the current data structure along with the rotations. This will all the user to only reference the material they want to look at and not have to scroll through pile of information that doesn't pertain to what they are looking for.