Cory Hershman

LookupTable Post-Mortem

This post-mortem will discuss the steps that were taken to complete the LookupTable project. First, the requirements were analyzed and listed. Next, classes and methods were proposed. The proposed classes were Node, RedBlackTree, and main. The node class would have 5 data variables: id, last name, first name, age, and color. Node class would have 4 pointer references: parent, left child, right child, and sibling. The sibling reference was later found to be unnecessary. The RedBlackTree class would have 9 methods: insert, delete, zigZigRes, zigZagRes, recolor, findSuccessor, promote, display, and find. After further analysis of the specification, it was realized that deletion was not necessary for the project, so the delete, findSuccessor, and promote methods were cut. Later on, more private utility methods were added to the RedBlackTree as well as 3 more search methods to carry out the input commands from the main class: findMatches, findMatchesLetter, and printComparisons. The initial design was simple to create and code. However, debugging the program took the longest amount of time. The main bug was parts of the list disappearing after certain records were entered. The biggest problem was missing pointer assignments in the zigZigRes and zigZagRes. All of the appropriate nodes were accounted for and considered when creating these methods, but some of the reverse assignments were missed. For example, when a node handed off one of it’s children to another node, the child’s parent assignment was not updated to the new parent, leading to some parts of the list being cut out. After these bugs were fixed, the documentation was completed. The most important experience for the future was the extensive use of the debugging tool to watch the data structure as it ran.