Allison Kane

Dr. Catherine Putonti

5 May 2016

COMP 271

Final Project

My final project for this semester was number 7, Contagion continued. This project required me to take two files that were filled with letters that represented a DNA sequence. My job was to take the letters from each file and turn them into words that get added to a tree. This project required a Quaternary tree because there are 4 separate letters instead of 2. After all of the words from the file are read into the tree, the code will run through the tree and check to see if a word is in fact a genome.

The data structures present in this code are necessary for a number of reasons. I decided to use a vector for the words because vectors do not need a preexisting size for functionality in this project. I also decided to use the class myTreeNode so each letter can be appended uniquely and I used a myTreeNode de-constructor in order to see if the child node already exists, and if it does, it will be deleted. I also used a class for myQuartTree to create a Quatenary tree that nodes will be appended to and deleted from. I also used arrays because they are simple for storing a known size.

To go through this code in main(), the computer will start by creating an file named “my\_results.txt” in the debug folder. The text file will include the final results of the program. The the code initializes myQuartTree to create a quaternary tree for testing. Then it will make a vector for words\_in\_genome and char genome\_file\_name [256]. Then, an if loop will use the function get\_words\_from\_the\_file. The user will be prompted for a file name and should enter “6.genome.txt”. This bool function takes in two parameters and take in the sequence of letters from the file and push them into a vector. The main() function then proceeds to add these letters into a tree starting from the root. Using an iterator, each node is created and appended by calling the CreateNode and AppendNode functions. The CreatNode and AppendNode function take a value and in order for a character to be created or placed upon a parent node. Then main will proceed to do a similar process to the get\_the\_reads\_from\_file () function. Again, the user is prompted for a file name and should input “6.reads.2.txt”. This function will read in strings of 30 characters long and it will take each letter and put it into another vector. After those functions have been called, the main() will then call the function that will test each case using the traverseTree function. The traverseTree function will use the tree from the class myQuarttree to check if the tree has any matches to a genome. The vectors will be written out to the file “my\_results.txt” by calling the write\_out\_the\_file\_to\_vector function, which will simply take the final tested tree and will print out the results to a text file in the debug folder.

For the IDE I used on this project, I used the XCode for Mac with OS El Capitan. I did not work with any partners for this project.