Wyatt Emery

Talon Marquard

3/2/16

COSC 3020

Project 2 Task 4

Our function has successfully found the longest path in the tree. The tree was only difficult when it came to thinking about the recursion and how it would operate. Recursion is difficult to think about and so we had to break it down into cases that it would face at each part. Once that was thought through it was fairly straight forward. The function to find this was less of an issue that actually building the tree itself. We had to be careful that it satisfied all the rules and did not break any of them. We made sure every property of a tree was applied to our build. Our test cases were successful and accurately found longest path, and were legal trees. The random tree was hard to build since we had to program it to follow the rules of a tree, rather than apply the rules to the trees we created. In the end we figured it out, and got our rand tree function to stop building forests. To solve these issues we just had to sit down and describe the aspects we needed to apply and program it a piece at a time.

Our tree has to find the longest path in linear time, which it does as can be seen by the data below. The ratio when the size of the tree is doubled is approximately two, which means that it is growing linearly. As the size of the tree doubles, so does the time. The code also shows this because it visits each vertex only once.

|  |  |  |
| --- | --- | --- |
| size | time | ratio |
| 1000 | 0.002201 |  |
| 2000 | 0.004448 | 2.021375 |
| 4000 | 0.010791 | 2.425693 |
| 8000 | 0.023782 | 2.203983 |
| 16000 | 0.049909 | 2.098556 |
| 32000 | 0.074767 | 1.498085 |
|  |  |  |
| mean ratio: | | 2.049539 |

Together we discussed the problem and what approach we needed to take to complete this project. From there Wyatt built the outline, structure, and most of the functions for the code, with input and comments from Talon. Wyatt got the code building trees from input from a file or from a random generator as well as put together all classes and private and public functions and variables created and working. The FindLongestPath function was coded by Talon with feedback from Wyatt. The problems faced by the random build were overcome together. Talon also wrote up the reports and calculated the time complexity of the FindLongestPath function.