Justin-Anthony Joco 0028483667
ECE 368

Programming Assignment 3 Report

The assignment was to load an input file into a binary tree, pack the rectangles stored in the tree, and write the packed information into an output file.

My program builds the tree as it reads the file. While reading the file, if the line gives leaf node information, it stores that information into a leaf node, which is then stored into a stack. Once the program reaches a character node, it pops the top two nodes in the stack, and those nodes become children of the character node. It continues this until it reaches the end of file. Consider that we have an input file with n nodes. Since we are constantly pushing and popping the stack (O(1) for each) and building a tree with n nodes, the time complexity is O(n).

As for calculating the rectangular coordinates, first, the program calculates the widths and heights of the smallest rectangles, then computes the x and y coordinates. To compute the smallest rectangles, the following algorithm was used:

- If non-leaf node is ‘H’, add the children’s heights and store the higher width
- Else if the node is ‘W’, add the children’s widths and store the higher height. \

Thus each non-leaf node should store the width and height of the smallest rectangle that contains its children.

To calculate the coordinates, I used the following algorithm:

- If non-leaf node is ‘H’, the left y coordinate is the right’s y coordinate + its height. Set left x-coordinate and right x-coordinate as equal.
- If non-leaf node is ‘V’, the right’s x coordinate is the left’s x coordinate + its width. Set left y-coordinate and right y-coordinate as equal.
- Update children of non-leaf node

As a result of both parts, the program visits each node at least twice. Thus, the time complexity of computing the rectangle coordinates is also O(n).