ECE 368

PA 03

Shulin Wang

Complexity of construction

The binary tree was constructed through iterations and the use of a stack with the array based implementation. The tree itself was constructed in the load file function, along with initialization of an empty stack, which was used later in order to help to build the tree. During the iteration, the information was gathered from the input file, then it was constantly been pushed into the stack and pop from the stack, and the tree was re-constructed with the help of the stack. The stack operation would perform at least once, as well as the tree operation, during each iteration. The number of iterations was depended on the size of the input file, hence, the time complexity of constructing a binary tree would be O (n).

Complexity of computation

The coordinates of the rectangles of the binary tree was calculated through packing function. In the packing function, there were three individual functions that were assigned to do different tasks, and the coordinates were found with the help of those helper functions. Each helper function was written with the use of recursion, which meant that the tree would traverse one time for each of the functions, and since the tree was log (n) deep, the complete time complexity of the packing function was log (n) + log (n) + log (n), and we could express it in term of Big O Notation as O (log (n)), and this was the time complexity of computing the coordinates of the rectangles for the given binary tree.