TreeGenerator

Anusha Fatima and Syed Shariq Ali

May 2016

1 Instructions

Please load the Main file in GHCI and call the main function. The user will be prompted to choose a type of tree. These are the following options:

1. Binary Tree
2. BST
3. Red Black Tree

The user is then prompted to input the numbers separated by space. If the user is interested in a Binary Tree, the numbers must be given in a heap order (with the root node followed by its left child and right child , followed by the left child of left child and the right child of left child and so on). If the user wants an empty node for the binary tree, he should denote the empty node in the tree using 00 or -100 to maintain the heap order property. For the other two cases, the program creates the correct tree for the numbers and generate it's latex file.

The file generated is a tex file with the name of the type the user chose. The file can be found in treeGenerator folder. There is one restriction in the visualization of Binary Tree, the user can give a list of at most 15 numbers which means the final tree can have at most 4 levels. The program will malfunction otherwise.

2 Further Development

Here are a list of possible improvements to this project:

1. Allow for other data structures like 2-4 trees, hash tables etc.
2. Remove the restriction on Binary tree.
3. Allow for performing of simple operations on a given tree such as rotation , insertion , removal etc.
4. Give user the ability to construct incorrect BST and Red Black Trees.

3 Division of Work

Anusha :

1. Created the modules and added relevant functions to convert an array of integers into the relevant trees.
2. Documentation and Debugging
3. Added code for user input

Shariq :

1. Wrote code to output latex for the different types of Trees
2. Documentation and Debugging
3. Integrated all the modules together