

Data Structures and Programming

Spring 2017, Programming # 2

Due: May 25, 2017

A binary tree can be easily represented in the following way. Start with the root. If there are any children, list them in parentheses. Use - (a dash) for a child that is missing; however do not display a dash or parentheses for a leaf. For example, the tree in Fig. 1 is represented as $9(5(3(2\ 4)\ 7(6\ -))\ 13)$.

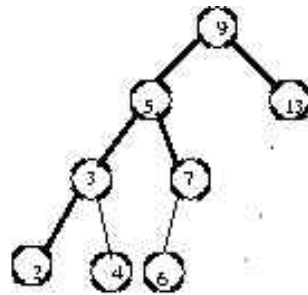


Figure 1: Binary tree represented by $9(5(3(2\ 4)\ 7(6\ -))\ 13)$

Suppose we want to display a tree on the screen in the following way. The method (called `printTree`) should display the tree with its root at the left of the screen. In other words, the tree should be "on its side." For example, the tree in Fig. 2 is printed with the root in the left margin as shown in Fig. 3. To space out the tree properly, let each column start 5 space to the right of the previous one. The

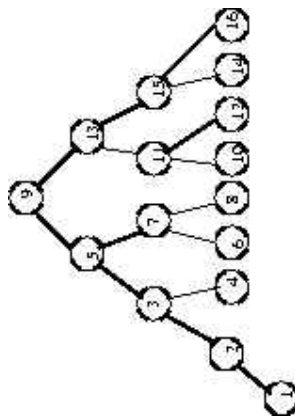


Figure 2: A binary search tree T

method should be recursive and use a "backward" inorder traversal (right-root-left) to output the tree.

In this assignment, you are asked to carry out the following:

- Implement `printTree(T)`
- Implement `Search(T, x)` on ordinary binary search trees and `Search_s(T, x)` on splay trees.

Then do the following:



Figure 3: Printing tree T

- (1) Read a tree from the input file "tree.txt" in which the tree is represented by its parenthesis form. Store the tree in a standard binary tree structure in which each of T 's nodes contains a key, a left-child pointer and a right-child pointer. You may assume that the tree in "tree.txt" is a binary search tree.
- (2) printTree(T)
- (3) Read from the file "op.txt" which contains the sequence of search operations to be performed on a tree T' explained in the following. For instance, 3 1 0 4 2 3 5 2 stands for searching keys 3 1 0 4 ... in the given order.
- (4) Like in Prog. # 1, we assume that in "op.txt" keys are of values 0 - 9. Find the frequency of occurrences for each key in the input search sequence. Suppose $n_0 > n_1 > n_2 \dots > n_9$ is the ordering where $n_i \in \{0, \dots, 9\}$. Construct a binary search tree T' recursively in the following way:
 - (a) Assign n_0 to the root of T'
 - (b) The root of the left (right, resp.) subtree is the one which has the max number of occurrences among those whose key values are smaller (greater, resp.) than n_0 .
- (5) printTree(T')
- (6) By treating T' constructed in Step (4) as an ordinary binary search tree, perform the operations in "op.txt" to T' . Like in Prog. # 1, output $(50, n_1)(100, n_2)(150, n_3) \dots (1000, n_{20})$, where n_i is the cumulated number of key comparisons after the $50 \times i$ search operations is completed.
- (7) Repeat Step (6) by treating T' constructed in Step (4) as a splay tree. In addition, perform printTree after the sequence of search operations is completed.

NOTE:

You need to document each of the subroutines in detail. In addition, your report must contain the following:

- (1) program files (source, object ...). Please name your executable file as "hw2.out".
- (2) output: in the format explained above. Your program should read from file "tree.txt" and "op.txt". We will execute your program in the following way: ./hw2.out tree.txt op.txt
- (3) a "readme" file describing the language (C, C++, JAVA, ...), the environment (Windows, Linux, FreeBSD, Mac, ...), as well as the type of compiler needed to execute your program. (The TA will execute your codes; so be sure that your program is well-documented and executable.) Be sure to include your name, student ID No, and phone number (home and/or cell phone) in the "readme" file in case we need to reach you.

- Place all of the above in a zip file with your student ID and the version number as the file name. (For instance, if your ID is B89900000, and the file to be uploaded is version 1, your file name should be "B89900000_1.zip".)
- Upload the zip file using "sftp" to 140.112.42.22:3922 with
 - username: ds17
 - password: datastructure

Find the /hw2/ directory and place your file in the directory. Note that those files already uploaded cannot be removed or altered. If you plan to upload a new version again, please increase the version number in your file name.

The "sftp" is readily available using "Filezilla" (supported by MS windows, linux, and osx), or simply use the "scp" command in unix.

NOTE: Should you have questions about the above, please send your inquiries to your TA at r05921040@ntu.edu.tw

The work must be the result of your individual effort. **No collaboration!!!**