

CIS 2168: Assignment #8

Due on Wednesday, November 5, 2014

11:59pm

Anwar Mamat

Problem 1

BST reconstruction (50%)

Write a method “public boolean construct(String preorder)” to the Binary Search Tree class BST. Given the postOrder traversal of a BST (not including null nodes), this method reconstructs the tree. It returns true if the BST can be successfully constructed, and returns false if the input is not valid. The input string postOrder traversal is given as a comma separated string. For example: for the tree shown in Figure 1, the input is 1,4,7,6,3,13,14,10,8. If the BST is correctly constructed, the tree will be displayed. The default value

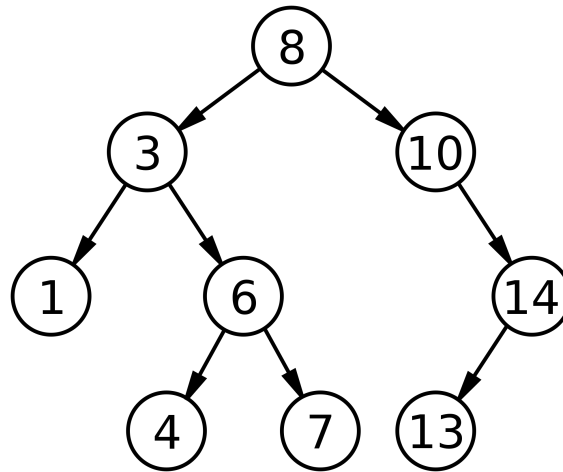


Figure 1: Binary Search Tree

for each key is “A”. For example, above tree will be displayed as Figure 3.

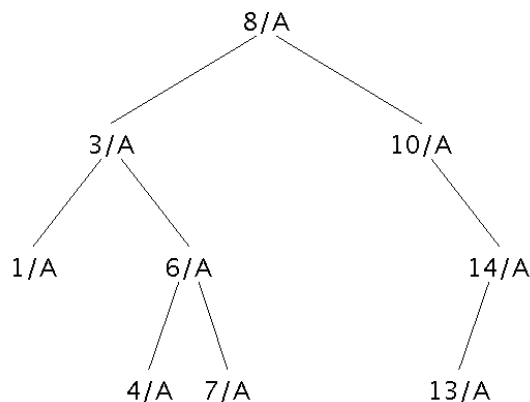


Figure 2: Binary Search Tree

Download the attached “assignment8.zip” file, extract it and open the “assignment8” project using NetBeans. Implement the “public boolean construct(String preorder)” method in the BST class. The method receives a string as an input and constructs the BST. You can add another recursive method if it is necessary. But you are NOT allowed to call any other BST methods such as put, get, delete, min, max etc. **You are not allowed to construct the tree by calling “put” method for each key.** You have to construct the tree using your “construct” method.

Problem 2

Binary Tree Iterator (50%)

Implement a preOrder iterator for the given BST so that we can preOrder traverse the binary tree using the iterator. For example: the given BST can be traversed using for for each loop:

```
Iterator<Integer> iter = bst.iterator();
while (iter.hasNext()) {
    System.out.print(iter.next()+" ");
}
```

or

```
for (Integer i: bst){
    System.out.print(i+" ");
}
```

For the tree shown in Figure 1, the output is

```
Enter postOrder traversal (comma separated, no space):1,4,7,6,3,13,14,10,8
constructed.
8,3,1,6,4,7,10,14,13,
5 8,3,1,6,4,7,10,14,13,
```

Another example:

```
Enter postOrder traversal (comma separated, no space):15,10,23,25,20,35,42,39,30
constructed.
30,20,10,15,25,23,39,35,42,
5 30,20,10,15,25,23,39,35,42,
```

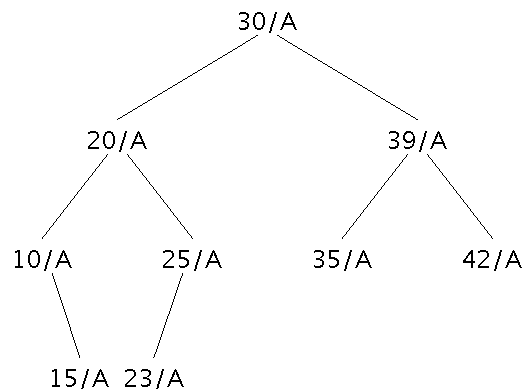


Figure 3: Binary Search Tree

What will you learn from this assignment

Stack, Recursion, binary tree, binary search tree

Grading

Homework is 100 points. 80 will reflect functionality and correctness. 20 points on your program will reflect your programming style, documentation. **If you code does not compile, you will not receive any credit.**

Commenting and Documenting Code

Code must be properly commented. The main idea is that the grader should be able to understand your code easily, not have to tear his or her hair out wondering what some statement is doing. The first time you have to deal with poorly commented code (if you haven't already), you will understand how annoying it is. In particular, the top of each code file should contain your name, the course and assignment numbers, and a brief summary of what's in the file. Line-by-line comments should be included as necessary to make the code easy to read. A clear coding style, together with informative variable and function names, will reduce the number of comments required. Obscure code or cryptic function names will cause loss of points (for bad style) and also require more extensive comments.

What to submit

A single zip file called Assignment8_firstname_lastname.zip, where firstname is your first name, and lastname is your last name. In this zip file, put:

1. Java source
2. A README file with:
 - Instructions to compile and run of your code (include a description of command line options).
 - If your solution is not perfect, explain what parts you did and what part you did not do.
 - List of files submitted
 - All your data and results (in plain text files).
 - Anything else you want TA know
3. Submit this zip file to Blackboard