Binary Search Trees

Description In this lab your goal is to implement standard operations of **binary search trees**, including insert and delete. See section 12.3 in the textbook. *There are millions of ways of implementing the delete operation, but you must follow exactly the same algorithm in the textbook.* In this assignment the keys are integers. Your code will be tested for examples consisting of distinct keys.

In the input, each starts with 'e', 'i', 'd', 'oin','opre', or 'opost'. For each line, you will have to do the following.

- i(key): Insert (key) into the BST. For example, i2 implies "Insert key 2 into the BST."
- d(key): delete (key) from the BST. For example, d2 implies "delete key 2 from the BST." Do nothing if the BST doesn't have the key.
- opre: output all keys via preorder walk.
- opost: output all keys via postorder walk.
- oin: output all keys via inorder walk.
- e: finish your program.

Example input and output

The following example shows an execution of the program in interactive mode. See the input files and output files under the testfiles folder for examples where input and output are separated.

```
i3
i1
i5
i7
oin
1
3
5
7
d7
oin
1
3
5
opre
3
1
5
opost
1
```

5
3
e

See the lab guidelines for submission/grading, etc., which can be found in Files/Labs.