BinarySearchTree.java BinarySearchTreeDriver.java BinaryNode.java BinarySearchTree.txt

Create a BinaryNode class. Then create a BinarySearchTree class which has the following functionality: add a BinaryNode (when adding, let a BinaryNode that is equal to the parent go to the right), an in order traversal, a pre order traversal, a post order traversal, a reverse order traversal, the number of leaves in the tree, the number of levels in the tree (that would include counting the root), the width of the tree (the width does not necessarily go through the root), the height of the tree (in this case height is defined as the number of nodes removed from the root, so don't count the root in the height), the total number of nodes in the tree, a boolean method that indicates if the tree is full (all nodes filled on all levels), a boolean method that returns whether a Comparable is contained in the tree, a getLargestMethod that returns the largest Comparable, a getSmallest method that returns the smallest Comparable, a method that returns the tree in levelOrder, a method that prints the tree as a tree (by which I mean level by level – BUT ONLY up to level 5), and a method that will remove a Comparable from the tree and return the Node that is removed (when removing a node with 2 children, let the left branch take precedent over the right branch).

Input File: The input file will start with a value indicating how many lines will need to be read to populate the tree. The next N lines will be composed of 1 or more words, each needing to be placed into the tree. The next value will be an int indicating how many words will be checked to see if they are contained in the tree followed by that number of words to be checked, finally you will read an int indicating how many words you will be removing and then a list of those words.

Sample File :

```
4
us and them
and after all we're only ordinary men
me
and you
2
them
every
4
us
them
yesterday
only
```

Sample Output :

```
Tree-->after all and and and me men only ordinary them us we're you
PRE ORDER
us and after all them and only men me and ordinary we're you
POST ORDER
all after and me men ordinary only and them and you we're us
```

```
IN ORDER
after all and and and me men only ordinary them us we're you
REVERSE ORDER
you we're us them ordinary only men me and and and all after
Number of leaves: 4
Number of levels: 8
The Tree width: 10
The Tree height: 7
Number of nodes: 13
Tree is not full.
Tree contains them
Tree does not contain every
Largest value: you
Smallest value: after
Level order: [us, and, we're, after, them, you, all, and, only, men,
ordinary, me, and]
Level Order Tree after removing: Value:us, Left:null, Right:null
Level order: [and, after, them, all, and, we're, only, you, men, ordinary,
me, and]
Level Order Tree after removing: Value:them, Left:null, Right:null
Level order: [and, after, and, all, only, men, ordinary, me, we're, and, you]
Cannot remove yesterday from tree
Level Order Tree after removing: Value:only, Left:null, Right:null
Level order: [and, after, and, all, men, me, ordinary, and, we're, you]
Proper Tree Display
and|
after|and|
null|all|null|men|
null|null|null|null|null|null|me|ordinary|
null|null|null|null|null|null|null|null|null|null|null|null|and|null|null|we'
re| //this is one line
null|null|null|null|null|null|null|null|null|null|null|null|null|null|null|nu
ll|null|null|null|null|null|null|null|null|null|null|null|null|null|null|null
|you| //this is one line
```