CENG 242

Homework #1

(Due: March 19th 2010, Friday 23:55)

In this homework, you will be given a binary tree and asked to find whether this tree is a binary search tree or not. Besides if it isn't a binary search tree, you should form a binary search tree from the given binary tree.

Assume that the tree is defined as:

```
data Tree = Empty | Branch Integer Tree Tree
```

You will write a function **getBST** in the form:

```
getBST :: Tree -> (Bool,Tree)
```

The function should return a pair of a Bool as the first element and a Tree as the second element.

Constraints:

- The nodes of the given tree will just include integers.
- A binary tree which cannot be formed into a binary search tree isn't going to be given to the function. (For ex: the same integer cannot appear more than once in the given tree)
- If the given tree is already a binary search tree, you should return the same tree as the second element of the resulting pair.
- If the given tree isn't a binary search tree, you can form any binary search tree that contains the same nodes of the given tree .

Examples:

- getBST (Branch 5 (Branch 3 (Branch 2 Empty Empty) (Branch 4 Empty Empty)) (Branch 7 (Branch 6 Empty Empty) (Branch 8 Empty Empty)))

  (True, (Branch 5 (Branch 3 (Branch 2 Empty Empty) (Branch 4 Empty Empty)) (Branch 7 (Branch 6 Empty Empty) (Branch 8 Empty Empty))))

- getBST (Branch 9 (Branch 7 Empty (Branch 8 Empty Empty)) (Branch 13 (Branch 6 Empty Empty) (Branch 3 Empty Empty)))

(False, (Branch 9 (Branch 7 (Branch 6 (Branch 3 Empty Empty) Empty) (Branch 8 Empty Empty)) (Branch 13 Empty Empty)))

or;

(False, (Branch 9 (Branch 7 (Branch 3 Empty (Branch 6 Empty Empty)) (Branch 8 Empty Empty)) (Branch 13 Empty Empty)))

or;

(False, (Branch 7 (Branch 6 (Branch 3 Empty Empty) Empty) (Branch 9 (Branch 8 Empty Empty) (Branch 13 Empty Empty)))

or;

Another binary search tree result with and the same content.

Specifications:

- All the work should be done individually.
- Your codes should be written in Haskell and have the name “hw1.hs”
- In evaluation, black box method will be used. So be careful about the name of functions, data structures etc.
- You will submit your code through Cow system.
- You should test your codes in inek machines with hugs before submitting.

Bonus:

In this homework, there is a 20 point bonus part for the non-binary search tree inputs. If you build all the trees as balanced binary search trees, you can get the bonus.

Note:

A show function must be implemented to view a Tree in the interpreter. So use this function in order to see your trees:

```
showTree :: Tree -> String

showTree Empty = "Empty"

showTree (Branch x left right) = "(Branch " ++ show x ++ " " ++
showTree left ++ " " ++ showTree right ++ ")"
```