Programming Assignment 6

Due 21 OCT @ 11:59pm

Write a program that implements a subset of the OrderedSymbolTable API using a binary tree. You will provide the implementation details in the BinarySearchTreeST. The provided main method IN BinarySearchTreeST.java will read in and execute a sequence of operations.

The keys() method should return an Iterable<Key> that iterates through the keys in ***sorted order***. You can use the provided List implementation (DynamicArray), e.g.

```
public Iterable<Key> keys()
{
    DynamicArray<Key> list = new DynamicArray<Key>();
    keys( this.root, list );
    return list;
}

private void keys( Node root, DynamicArray<Key> list )
{
    ...
}
```

Grading Notes

You must:

- Use the template provided for you
- Have a style (indentation, good variable names, etc.)
- Comment your code well (no need to over do it, just do it well)

You may not:

- Make your program part of a package.
- Use *code* from anywhere except your own brain.

Submission Instructions:

- Name a folder with your gmu username
- Put your java files in the folder (but not your .class)
- Zip the folder (not just the files) and name the zip “username-pa6.zip”
- Submit to blackboard

Grading Rubric

No Credit:

- Non-submitted assignments
- Late assignments
- Non-compiling assignments
- Non-independent work

1pt	Submission Format
1pt	Style and Comments
1pt	min and max
2pts	put
1pt	get
2pts	deleteMin, deleteMax
2pt	keys iterator

Example Run

```
> java BinarySearchTreeST operations.txt
isEmpty?=true
insert=[8->student8]
insert=[4->student4, 8->student8]
insert=[4->student4, 8->student8, 12->studentX]
insert=[4->student4, 6->student6, 8->student8, 12->studentX]
insert=[4->student4, 6->student6, 8->student8, 10->student10, 12->studentX]
min=4
insert=[4->student4, 6->student6, 8->student8, 10->student10, 12->student12]
insert=[2->student2, 4->student4, 6->student6, 8->student8, 10->student10, 12->student12]
max=12
insert=[2->student2, 4->student4, 6->student6, 8->student8, 10->student10, 12->student12, 16->student16]
insert=[1->student1, 2->student2, 4->student4, 6->student6, 8->student8, 10->student10, 12->student12, 16->student16]
deleteMin
deleteMin
deleteMax
isEmpty?=false
size=5
Final symbol table=[4->student4, 6->student6, 8->student8, 10->student10, 12->student12]
```