Intro to Algorithms

Homework

Binary Search Symbol Table Implementation

(NOTE: This is *NOT* the "Binary Search Tree" Implementation)

The Assignment

Your assignment is to implement the `delete()` and `floor()` methods for the "binary search symbol table". You *must* implement the Sedgewick APIs and you *should* copy the Sedgewick code (pg 379, 381, 382) and augment the book's *BinarySearchST* class with these (not-implemented-in-textbook) methods.

My tests will be exercising:

- the **no-argument** ctor
- get
- put
- keys() (the version which returns *all* keys, in-order, in the symbol table)

You **must** make **each of these methods** *public* so they can be tested!

Warning: Implementing these methods will tend to suck in other API methods as well. (Those methods need not be public, but I recommend that you declare them "public" as well.)

In addition, you **must code the following constructor**

```
  /** Initializes the ST with initial keys and corresponding values.  The keys
   * and values are inserted in array order: i.e., first (initialKeys[0],
   * initialValues[0], then (initialKeys[1], initialValues[1])
   * .... (initialKeys[n-1], initialValues[n-1])
   */
  public BinarySearchST(Key[] initialKeys, Value[] initialValues)
```

Packaging

Assume that you've installed your YU Git repo in a directory named `GIT`.

- Your homework assignments for this course must be rooted in `GIT/IntroToAlgorithms/homework`. I'll refer to this directory as `ROOT`.
- Your code will reside in a package named `edu.yc.oats.algs`.
- Your code for this assignment will be rooted in: `ROOT/BinarySearchST`. I'll refer to this directory as `DIR`.
- Your class must be named **BinarySearchST.java**, and it must be reside `DIR/src/main/java/edu/yc/oats/algs`.
- Your assignment may not use any external libraries: just the SDK and your code in this package.

Grading

- If your code cannot be compiled & run (either because it doesn't follow the packaging conventions above, or because of a compilation bug) -- **automatic 0** for the assignment
- If your code runs, but doesn't pass my tests, you'll get a *maximum* of 8. The actual grade will depend on how close your code was to passing the tests.
- If your code runs, passes the tests, but is "really ugly", you get a 9.
- Maximum grade is 10.