P0.Rotation

CS400 Projects

For future projects in this course, you will be paired with class mates and together work on group programming projects. You can let us know through **this form (https://forms.gle/Naikzs18ABoj6m5E7)** what your preferences for these groups are by Wednesday, Feb 1. Group projects will span multiple weeks during which you will start by working with your group on a proposal for an application built around a key data structure. Then, each of you will individually implement the data structure, and finally you will build the proposed application together with your team mates.

Project 0 is most similar to the individual data structure implementation week of these future projects: through which you independently write Java classes or methods that implement data structures or algorithms discussed in class, test your implementation to identify bugs, and work through fixing them.

Work Individually

The algorithm that you will individually implement for project 0 is a rotation on a BST. You must develop the code for this assignment entirely on your own. If you encounter difficulties while working, you are encouraged to discuss general algorithms and debugging strategies with anyone you would like. But you are NOT ALLOWED to view any rotation implementations of others, and you are NOT ALLOWED to show or allow access to your implementation to anyone outside of the CS400 course staff.

Setup

1. Download the source files for the activity either through the link: **RedBlackTree.java (http://pages.cs.wisc.edu/~cs400/RedBlackTree.java)** and **SortedCollectionInterface.java (http://pages.cs.wisc.edu/~cs400/SortedCollectionInterface.java)**, or by using `wget` with

   `wget http://pages.cs.wisc.edu/~cs400/RedBlackTree.java`
   and
   `wget http://pages.cs.wisc.edu/~cs400/SortedCollectionInterface.java`

2. Create a new project folder in your preferred IDE or for your preferred editor and copy the downloaded files into it.

Implementation

1. Review the contents of the provided `RedBlackTree.java` and `SortedCollectionInterface.java` source files. The RedBlackTree class implements a regular (non self-balancing) binary search tree. This week, you will implement tree rotations for this binary search tree. In a future project, we will extend this class further to implement a self-balancing binary search tree called a Red-Black tree for whose insertion algorithm you will make use of your rotation implementation from this week. The Red-Black tree class also provides code for displaying the contents of the tree via in-order traversal (the sorted sequence of values in the tree).
2. To work on your tree rotation implementation, implement the method named `rotate` to follow the rotation operations described in lecture.
3. Implement the three test methods at the bottom of the `RedBlackTree` class file to convince yourself that your rotation implementation is correct. Use examples for rotations from lecture and make sure that you test for rotations within a tree, as well as at the root node of a tree. Note that there are two different traversals that the tree type implements, a level order traversal (`.toLevelOrderString()`) and an in-order traversal (the iterator and `.toInOrderString()` method of the `RedBlackTree` class). Both methods will be helpful for testing your implementations. More details can be found in the comments of `RedBlackTree.java`.

Assignment Submission

Please submit your progress in `RedBlackTree.java` early and often through **[gradescope (https://gradescope.com/)](https://gradescope.com/)** : as a form of backup, and to get feedback from the automated grading tests. The only file that you need to submit is `RedBlackTree.java`. If you define extra classes in other source files as part of your solution, then you should include those files with your submission as well. Your most recent submission in gradescope will be marked "active" by default, but you can change this to an earlier on-time or less buggy submission if you prefer. Ensure that your final "active" submission for this assignment is clearly organized, consistently styled and well documented with comments.

Grading

Formative (12pts - you'll get immediate feedback and as many submission attempts as you need to earn these points) and summative tests (8pts - you won't receive feedback or scores for these tests until after the late deadline for the assignment has passed) on gradescope.