

Problem Set 3: Balanced Trees

This problem set explores balanced trees, augmented search trees, data structure isometries, and how those techniques can be used to find clever solutions to complex problems. By the time you've finished this problem set, you'll have a much deeper understanding for how these concepts relate to one another. Plus, you'll have designed and implemented several truly beautiful data structures!

Due Thursday, May 6th at 3:00PM Pacific time.

Problem One: Order Statistics Trees

In this problem, you'll take an implementation of a red/black tree that only supports insertions and lookups, then convert into an order statistics tree by adding support for the `rankOf` and `select` operations. The `select` operation is the one we talked about in lecture: it takes in a number k , then returns the k th order statistic. The `rankOf` operation is a sort of inverse of `select`: it takes in a key, then returns the number of elements in the red/black tree smaller than the key.

Download the starter files for PS3 from myth at

```
/usr/class/cs10/assignments/a3
```

and edit the `RedBlackTree.h` and `.cpp` files with your solution.

Some notes on this problem:

- You are free to edit whatever parts of the provided starter code that you see fit to edit, provided that (1) you still back the data structure with a red/black tree and (2) all operations run in time $O(\log n)$, except for the destructor (time $O(n)$) and `printDebugInfo` (can be whatever you'd like). We don't think you will need to do much surgery on the provided `RedBlackTree` type, so if you find yourself fundamentally rewriting large parts of the code, chances are you're missing an easier solution.
- We've provided two different test harnesses. One of them (`./run-tests`) will run random inputs into the tree and see how the operations work. The other (`./explore`) can be used to interactively test your tree or to run small scripts on the tree. We recommend using both in the course of coding this on up.

To receive full credit, your code should compile with no warnings and should not have any memory errors (use `valgrind` to check this). We'll test your code on the myth cluster. There's information about how to run the test drivers in the `README` file.