

Retail Stock-keeping System

Due: 23:59 Thursday, June 16, 2016

List of non-trivial updates (2016-06-07):

- (2016-06-07) Error corrected in `rbtreepartial.cpp`, `PreOrder` function
- (2016-06-06) Initial specification released

In this assignment you will implement a `StockSystem` class modeling a simple retail storefront inventory. The data storage of the `StockSystem` class will be supported by a `RBTree` class which implements the red-black tree data structure as described in lectures.

You are encouraged to work in groups of **at most** two members. You must include in your `cs221/a3` directory a `titlepage-a3.txt` file listing the names, student numbers, ugrad IDs, and lab sections of all contributing members. After using `make clean` to remove any compiled binaries and temporary files, use `handin cs221 a3` to submit your assignment. After you have submitted, you can verify that the `handin` was successful with `handin -c cs221 a3`. If you wish to overwrite your submission with a newer submission, use `handin -o cs221 a3`.

Red-Black Tree Description

A documented declaration of a `Node` template class and `RBTree` class is provided in `rbtree.h`. Please refer to the comments in the header file for the definition and functional requirements. A subset of the `RBTree` functions has already been implemented for you in `rbtreepartial.cpp`. It is highly recommended that you study the completed functions as well as the other implemented classes to become familiar with their operation. You are required to complete the implementation of the remaining functions and any private functions you have declared in `rbtree.cpp`:

- `CopyTree`
- `RemoveAll` (private helper function)
- `RBRemoveFixUp`
- `ComputeHeight`
- constructors, destructor, `operator=`
- `Insert`
- `Remove`
- `RemoveAll`
- `Size`
- `Height`

Please note that this implementation of red-black tree is not to store duplicate values.

StockSystem class

The `StockSystem` class supports some simple functions for managing a retail store. A `RBTree` object is used to store records of the stock catalogue. The `StockSystem` class is to achieve its operations using calls to `RBTree` methods only. Please refer to `stocksystem.h` for the class definition and functional requirements.

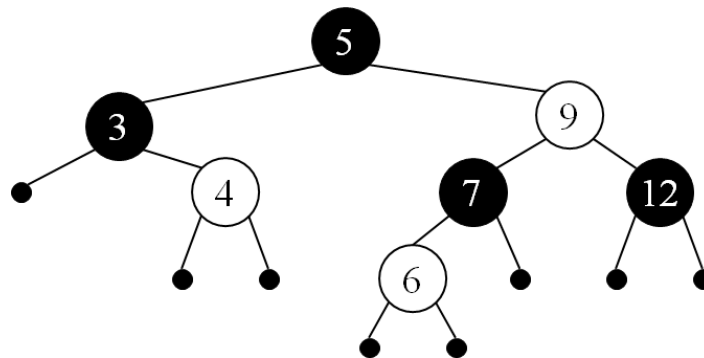
Testing Tips

As with assignment 2, the recommended approach is to begin your project by first completing all functions in `rbtree.cpp` and `stocksystem.cpp` as stubs which contain no logic and simply return a value of the appropriate type. Ensure that your code compiles with stubs, before completing the implementations one-by-one, compiling after completing each function.

The `StockSystem` uses only a subset of the available `RBTree` functions. Both classes will be thoroughly tested during grading – it is highly recommended to write a custom test driver or add your own code to the provided test driver to invoke all class functions and all their special and general cases. The grading driver for assignment 2 has been made available on the course website, you may download it to gain some understanding of the types of tests which may be run for your submitted program.

The instructors will be testing your tree using the `GetRoot()` function, but if you wish to test the structure of your red-black tree, you may make use of the `DumpPreOrder()` function. Performing a pre-order traversal starting from the root node and displaying both the node contents and node colour will allow you to analyse the structure of your tree.

e.g. For the following tree,



A pre-order traversal which outputs the node contents and colour as a string may produce:

5* 3* 4 9 7* 6 12*

Performing insertion of the above sequence into an ordinary BST will replicate the structure of the original red-black tree. You may find it helpful to use a visualization such as <https://www.cs.usfca.edu/~galles/visualization/RedBlack.html> to determine the expected structure of your tree after performing some sequence of insertions and removals.

Deliverables

Please ensure that your cs221/a3 directory contains the following files and submit using `handin -c cs221 a3`:

- `titlepage-a3.txt`
- `rbtreeprivate.h`
- `rbtree.cpp`
- `stocksystem.cpp`