Description

Checkout the assignment using the command `handin --checkout a7`. In this assignment we will be using current COVID-19 data as an example of how red-black trees can be used. When we studied hash tables, we saw how usefule they were for key-value pairs due to their amortized search time of $O(1)$. Hash tables are not as great for key-value pairs when a specific order is meant to be maintained. Red-black trees offer this capability with $O(\log n)$ search time, and is used in many map implementations in many different programming languages.

Instructions

The goal of this assignment will be to store key-value pairs in a red-black tree, where the value will be a struct containing the name of a country/region, and both total confirmed cases of COVID-19 in that country and total recovered from COVID-19 in that country. The tree will be sorted based on the recovery ratio $\frac{\text{\# recovered}}{\text{\# confirmed}}$, and as such this ratio will be the key. To complete this assignment, you must do the following:

- First you must complete the red-black tree insertion algorithm in our btree library we've been working on in class. You must complete the following functions: `left_rotate`, `right_rotate`, and cases 2b-1 and 2b-2 of `fix_violations`. Remember to test your code! I would suggest that you make another file called `rbtree_test.c` that performs a straightforward test similar to the one I do in the lectures to ensure that your code is working.
- Now you will be making all of your changes in the file `main.c`. You should read the file `/u1/class/cs202/covid-19/a7.csv` into a red-black tree. The comparison function passed to the red-black tree when it is created should compare two countries based on their respective recovery ratios. They should be in decreasing order.
- Finally your program should do the following with the red-black tree:
  - If no command line arguments are passed it should simply print all of the data using an in-order traversal of the tree. The country with the highest recovery ratio should be printed first.
  - If the `-l <low-ratio>` command line option is passed, it should print only those whose ratios are higher than `low-ratio`. Dido for the command line argument `-h <high-ratio>`.
  - If both arguments are passed, it will print all country's data whose ratio falls in the specified range. Your traversal should be smart enough to only visit the nodes in tree that it needs to.