CPSC 2430 Spring 2020 Programming Assignment #3

Due date: May 18, 2020 11:59pm

DO NOT USE any of the STL (including the stack and queue type) in your code.

Binary Search Tree (BST)

Implement a BST using a linked representation. The class should be named CityBST, and the class structure should match the tree class structure covered in lectures. Implement all basic class functions (*default constructor, copy constructor, destructor, copy assignment*) and basic tree functions:

void insert(string, string, int): adds a city with name, country, and population.
bool remove(string cityname): returns false if the city is not in the tree.
int getHeight(): returns the height of the tree.

TreeNode (City)

TreeNode is similar to the TreeNode you used in lab3, but it uses a string as a key and has additional members.

```
struct TreeNode{
  string key; //name of the city
  string country;
  int population;
  TreeNode* left;
  TreeNode* right;
};
```

Reading in Cities

The list of cities to add to the tree will be in **cities.csv** . Discard the first header line, then read in each city similarly to how you read in Items for PA 1. Pass all the information to the CityBST insert function to add each city to the tree.

The nodes in the tree should be ordered by the name of the city. Strings can be compared with < and > symbols, similar to integers, based on the alphabetical order. For example, *Ankara* will be on the left side of *Shanghai* since A is before S in the alphabet.

BST Functions

Implement the following functions:

1. print(): prints the names of all the cities in the tree using an inorder traversal.

2. printCity(string city): finds and prints all the information about the city.

Example

```
City: Sao Paulo
Country: Brazil
Population: 18,845,000
```

3. string biggestCity(): returns the name of the city with the largest population. Returns an empty string (i.e., “”) if the tree is empty.
Hint: use a recursive helper function that returns the largest city as a TreeNode*.

Testing and submission

Provide a driver/client program to demonstrate your functions. Test ALL the functions of your class thoroughly in the driver, including all the edge cases.

Submit your program using the following command:
/home/fac/hkong/submit/cpsc2430/submit_pa3

The CityBST class should be in two files named **citybst.h** and **citybst.cpp**. The driver should be in a file named **pa3.cpp**.

Execution example

```
[pa3]./run
Hello! Processing the cities.csv file.
Tree height is 4

Printing tree:
Delhi
Los Angeles
Mexico City
Mumbai
New York
Sao Paulo
Shanghai
Tokyo

Which city do you want to print? New York
City: New York
Country: United States
Population: 19,354,922

Which city do you want to remove? Shanghai
Shanghai has been removed. Printing updated tree:
Delhi
Los Angeles
Mexico City
Mumbai
New York
Sao Paulo
Tokyo

The biggest city is Tokyo.
```