Data Structures
Homework #3: Search trees, Red-Black tress, and ordered maps

This homework aims to help students understand, use and modify the implementation of binary search and red-black trees provided in the Goodrich's book.

Part 1:
Insert the binary search tree implementation (see section 10.1.3 of the textbook) in your own programming environment, compile and test this implementation. Provide a displayTree() function to visualize the content of the tree. Use the examples of figures (10.31, 10.32), and (10.37,10.38) of the textbook to test the insertion and removal functions.

Part 2:
Extend the class BinarySearchTree (from part 1) to support all the functions of the ordered map ADT (see section 9.3).

Test your implementation using the Flight Database application discussed in section 9.3.2 of the textbook.

Part 3:
Insert the red-black tree implementation (see section 10.5.2 of the textbook) in your own programming environment, compile and test this implementation. Provide a displayTree function to visualize the content of the tree. Use the examples of figures (10.31, 10.32), and (10.37,10.38) of the textbook to test the insertion and removal functions.

Part 4:
Write a new C++ class that implements all the functions of the ordered map ADT (see section 9.3) using red-black trees (from part 3).

Test your implementation using the Maxima Sets application discussed in section 9.3.2 of the textbook.

How to submit: Comment and organize your program as a project and upload it on the submission system. Your submission should be composed of four Parts. Each part includes .cpp, .h, and main.cpp to allow testing your implementation. Email attached submissions will not be considered.

Deadline: Check the deadline date on the NYU new classes system.