**Notes:**

* You are required to upload your in–class implementations of problems 1, 2(i), 3 to canvas. This is due by 9:50 AM today.
* You are required to turn in a written report (Word or PDF file) for the homework part (problems 2(ii) and 4) of the lab and upload implementations to canvas. These are due by 8:00 AM, October 25, 2017).

**Objectives:**

* Insert elements in a Red-Black Tree
* Use the Red-Black Tree in a real application

**Problems**

From the last assignment, extend the BSTNode class to form a new class RBNode to add a color field and a value field to each node (let the color field be of type *boolean* and the value field be a generic type). Also extend the BinarySearchTree class to create the RBTree class and include the following methods:

1. Implement Left Rotation and Right Rotation methods in the RBNode class. The Left-Rotate algorithm is given in the next page. Following that algorithm, devise the Right-Rotate algorithm and implement the methods.
2. Implement a method RBInsert(K key, V value) in RBTree.java to insert a node to a Red Black Tree (this method uses the insert method of BinarySearchTree, look at the algorithm next page).

In-class: Implement the case for which parent of x is the left child of grandparent of x.

Homework: Implement the case for which parent of x is the right child of grandparent of x.

1. Write a driver program to test the RBTree class. Make sure you use all the methods implemented above.
2. Use the RBTree class to build a search tree using the given input file that consists of two fields: a UPC key and the corresponding description. Use the search tree created to find the description associated with a given set of UPC keys. The input file *UPC-random.csv* provides the key and corresponding descriptions in a comma separated file and the various search keys are provided in the file *input.dat*. First test the program by entering couple of keys manually and print the description. Once you are convinced the program is working correctly, test the program for the given search keys and determine the total time taken to complete the search.

|  |  |
| --- | --- |
|  |  |

