Lab 06 Specification – A practical exercise to practice more on Binary Trees, 2-3 Trees, and Red Black Trees.
50 points

Lab Goals

- Do multiple exercises on Tree-Algorithms.

Suggestions for Success

- Take a look at the suggestions for successfully completing the lab assignment, which is available at: `https://www.cs.allegheny.edu/sites/amohan/resources/suggestions.pdf`

Learning Assignment

If you have not done so already, please read all of the relevant "GitHub Guides", available at the following website:
`https://guides.github.com/`
that explains how to use many of the features that GitHub provides. This reading will help you to understand how to use both GitHub and GitHub Classroom. To do well on this assignment, you should also read

- **Sedgewick ch3: 3.2 & 3.3**

Assignment Details

Now that we have completed discussing important concepts related to tree-algorithms and on balanced trees in the last few lectures, it is your turn. In this lab, you will practice a variety of problems to retain the knowledge of Tree-based algorithms. This includes solving one or more problems and analyzing algorithms.

This lab is primarily intended to be done remotely. All students are expected to work on this lab asynchronously for the most part. It will be highly recommended that every student remain online during the entire lab session in order to maximize learning. Students who are unable to join the lab due to a different time zone should promptly reach out to the Professor and an appropriate arrangement will be done individually.

Please make sure to push your code file(s) and commit by the deadline, April 3rd, 2020, morning 8:00 AM.

It is required for all students to follow the honor code. Some important points from the class honor code are outlined below for your reference:

1. Students are not allowed to share any solution files and/or other implementation details. It is acceptable to have a healthy discussion with your peers. However, this discussion should be limited to sharing ideas only.
2. Submitting a copy of the other's solution is strictly not allowed. Please note that all work done during lab sessions will be an opportunity for students to learn, practice, and master the materials taught in this course. By doing the work individually, students maximize the learning and increase the chances to do well in other assessments such as skill test, exams, etc $\cdots$

At any duration during and/or after the remote lab session, students are recommended to team up with the Professor and the TL's to clarify if there is any confusion related to the lab and/or class materials.

Section 1 - 2-3 Trees (20 points)

- **ADD** the following statement to all your solution files including the file named `BTree-solution`.

  This work is mine unless otherwise cited - Student Name

- Let us suppose that we are given an input string, the word, **dermatoglyphics**. Draw the 2-3 Tree by inserting the word (one character at a time from left to right). English alphabetical order should be used for positioning. For example A has a value of 1, E has a value of 5, and Z has a value of 26.
- In your final solution, you should simply list the **level-ordered traversal** sequencing of the individual 2-3 trees, after inserting all the characters (step by step) in the word into the tree and making sure the tree is fully balanced. That is, I expect to see 15 **level-ordered traversal** sequencing.
- If there is more than one key in a node then the node structure should be listed from left to right using a parenthesis. For example, if a node has two keys, namely E and S, then it should be listed as (E, S). During the delete process, if there are four vowels in the input word, then there should 4 **level-ordered traversal** sequencing provided.
- The **level-ordered traversal** sequencing for both insert and delete should be provided in the `BTree-solution` file. The file may be in a markdown or PDF format.

Section 2 - An Analysis on Red Black Tree-Algorithms (15 points)

- **ADD** the following statement to all your solution files including the file named `RB-analysis`.

  This work is mine unless otherwise cited - Student Name

- **Analyze** the algorithms provided in this section. We had recently completed discussing Red-Black Tree Rotation and Insertion procedure. I would like you to analyze both the algorithms, Insert and Fixup, and provide a detailed summary of your reflection on what is going on in those Algorithms (line by line). Note: The lecture slides include the algorithm in a simplified and informal format. The algorithms provided in this section is doing similar things but in a more formal format.

RB-Insert(T, z)

```
1.   y ← nil[T]
2.   x ← root[T]
3.   while x ≠ nil[T]
4.       do y ← x
5.          if key[z] < key[x]
6.             then x ← left[x]
7.             else x ← right[x]
8.   p[z] ← y
9.   if y = nil[T]
10.      then root[T] ← z
11.      else if key[z] < key[y]
12.           then  left[y] ← z
13.           else right[y] ← z
```

RB-Insert(T, z) Contd.

```
14.  left[z] ← nil[T]
15.  right[z] ← nil[T]
16.  color[z] ← RED
17.  RB-Insert-Fixup (T, z)
```

RB-Insert-Fixup (*T*, *z*)

```
1.   while color[p[z]] = RED
2.       do if p[z] = left[p[p[z]]]
3.             then y ← right[p[p[z]]]
4.                  if color[y] = RED
5.                     then color[p[z]] ← BLACK  // Case 1
6.                          color[y] ← BLACK    // Case 1
7.                          color[p[p[z]]] ← RED // Case 1
8.                          z ← p[p[z]]          // Case 1
```

RB-Insert-Fixup(*T*, *z*) (Contd.)

```
9.                 else if z = right[p[z]]  // color[y] ≠ RED
10.                     then z ← p[z]                    // Case 2
11.                          LEFT-ROTATE(T, z)           // Case 2
12.                     color[p[z]] ← BLACK              // Case 3
13.                     color[p[p[z]]] ← RED             // Case 3
14.                     RIGHT-ROTATE(T, p[p[z]])         // Case 3
15.       else (if p[z] = right[p[p[z]]])(same as 10-14
16.              with "right" and "left" exchanged)
17. color[root[T ]] ← BLACK
```

- Provide the answers to the following questions related to the Insert procedure:
  1. **Q1:** How does the Insert procedure outlined above differ from the BST Insert procedure discussed in earlier lessons?
  2. **Q2:** Which of the Red Black Tree properties might be violated during the execution of the insertion procedure outlined above? That is, we discussed the 5 red black properties in a recent lesson. Which of those properties might be violated during the insert process?
- Provide the answers to the following questions related to the Fixup procedure:
  1. **Q3:** Explain what is Case 1 in the algorithm? Explain the fundamental logic behind this case in a few words.
  2. **Q4:** Explain what is Case 2 in the algorithm? Explain the fundamental logic behind this case in a few words.
  3. **Q5:** Explain what is Case 3 in the algorithm? Explain the fundamental logic behind this case in a few words.
- The detailed summary of (line by line) description should be included in the **RB-analysis** file along with the answers to the questions **Q1 to Q5** presented above. The file may be in a markdown or PDF format.

Section 3 - An exercise on RB Insert Algorithm (15 points)

- **ADD** the following statement to all your solution files including the file named `RB-solution`.

  This work is mine unless otherwise cited - Student Name
- Let us suppose that we are given the first 15 numbers in the ascending order, [1- 15]. Draw the Red-Black Tree by inserting the numbers in the order [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15] step by step.
- In your final solution, you should simply list the **level-ordered traversal** sequencing of the individual Red-Black trees, after inserting all the values (step by step) and making sure the tree is fully balanced. That is, I expect to see 15 **level-ordered traversal** sequencing in your solution.
- Clearly write down the color of each node in the sequencing. If there is a red node with value 5, then simply include the color using the parenthesis, 5(R), in your solution. If there is a black node with value 10, then simply include the value, 10, in your solution. It will be understood that the node values without parenthesis is a black node.
- The detailed solution should be included in the **RB-solution** file. The file may be in a markdown or PDF format.

Submission Details

For this assignment, please submit the following to your GitHub repository before the **deadline** by using the link shared to you by the Professor:

1. Submission Files, namely, the “BTree-solution”, “RB-analysis”, and “RB-solution”.
2. It is highly important, for you to meet the honor code standards provided by the college. The honor code policy can be accessed through the course syllabus. Make sure to add the statement ”This work is mine, unless otherwise cited.” in all your deliverables such as markdown and PDF files.

Grading Rubric

1. There will be full points awarded for the lab if all the requirements in the lab specification are correctly implemented. Partial credits will be awarded if deemed appropriate.
2. Failure to upload the lab assignment code to your git repo will lead you to receive no points given for the lab submission. In this case, there is no solid base to grade the work.
3. There will be no partial credit awarded if your code doesn’t compile correctly. It is highly recommended to validate if the correct version of the code is being submitted before the due date and make sure to follow the honor code policy described in the syllabus. If it is a late submission, then it is the student’s responsibility to let the professor know about it after the final submission in GitHub. In this way, an updated version of the student’s submission will be used for grading. If the student did not communicate about the late submission, then automatically, the most updated version before the submission deadline will be used for grading purposes. If the student had not submitted any code, then, in this case, there are no points awarded to the student.
4. If you need any clarification on your lab grade, talk to the Professor. The lab grade may be changed if deemed appropriate.