

BLG 335E, Analysis of Algorithms I, Fall 2016, Project 5

Handed Out: *December 14, 2016*

Due: *December 28, 2016*

Problem: In this project, you are expected to implement certain basic Red–Black Tree operations and then augment your data structure with extra operations for order statistics.

Part A. Implementation (70 points)

1. (40 points)

Implement a basic Red–Black Tree that supports insertion and printing. You do NOT have to implement updating and deletion.

The key for each of the nodes should be the corresponding person's name. Age and gender values should be kept as extra attributes within your nodes.

Your insertion operation should insert a new node into the relevant position in the Red–Black Tree and then recolor and rotate existing nodes in order to meet the constraints and rebalance the tree.

2. (30 points)

Augment your Red–Black Tree implementation with two new methods, n^{th} adult and n^{th} non-adult, that return the name of the n^{th} adult and non-adult respectively. Include the **num_adult[x]** and **num_non-adult[x]** fields in your tree nodes, respectively holding the number of adult and non-adult in the sub-tree rooted at **x**, including **x** itself. Your implementation should make use of these fields in parallel with the **size[x]** field in the **OS_SELECT** example¹. Make sure you update these fields as necessary whenever you modify the tree by another operation. (A person is an adult if his/her age is greater or equal to 18)

Execution:

You are given an input file (**input.txt**) containing rows of data, each denoting a person. A name, an age value and a gender value is encoded for each person. Read the contents of the file and use your insertion operation to insert each person into your Red–Black Tree.

Your program should run from the command line with the following format:

```
$ ./<student_ID> <input_file>
```

input_file: The relative path to the input file, e.g. 'input.txt'.

An example execution command could be as below:

```
./040080154 input.txt
```

¹ Refer to the lecture slides if you would like to review the example.

For your output, use your printing operation to display the final state of your Red–Black Tree after all the elements from the input file have been added. Afterwards, use the **num_adult** and **num_non-adult** methods you implemented to find and print the 5th adult and the 3rd non-adult. You may display your output on the console rather than using output files.

Your printing operation should use the pre-order traversal to recursively print the nodes in the red-black tree. The output should properly represent the depth of nodes by indentation as shown below(use "-" instead of spaces). You must state if a node is black(B) or red(R).

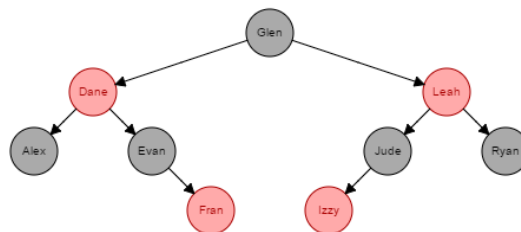
Sample input:

Glen	F	29
Ryan	F	17
Alex	M	13
Dane	F	14
Leah	F	23
Jude	F	26
Evan	M	18
Izzy	M	27
Fran	M	30

Sample output with indentation(the image displays the final state of the tree after insertions):

```
(B) Glen F 29
-(R) Dane F 14
--(B) Alex M 13
--(B) Evan M 18
---(R) Fran M 30
-(R) Leah F 23
--(B) Jude F 26
---(R) Izzy M 27
--(B) Ryan F 17

5th adult: Jude
3rd non-adult: Ryan
```



Part B. Report (30 points)

In your report, you will be expected to address the following questions. You do NOT have to provide details about your code. However, we ask you to **put a screen-shot** of your program's output.

1. (15 points)

Briefly explain what you would do to correctly update the name of a person as a node in the Red–Black Tree.

2. (15 points)

Briefly explain what you would do to correctly increment(by 1) the ages of all people in the Red–Black Tree.

Additional Notes:

1. Submissions will be done through the Ninova server. You must submit all your source code in a single cpp file and a softcopy report (PDF).
2. Make sure that GNU C++ compiler (g++ 4.8.5) compiles your project, and the application runs in Linux smoothly. You can use [ITU ssh server](#) to compile and test your application. This is important because we will evaluate your homework in Unix using g++.
3. Use comments wherever necessary in your code to explain what you did.
4. You are not allowed to use the standard template library (STL).

Note: If you have any questions, please feel free to contact Res. Asst. Çağatay Koç via e-mail (kocca@itu.edu.tr).

Academic dishonesty including but not limited to cheating, plagiarism, collaboration is unacceptable and subject to disciplinary actions.