Data Structures and Algorithms, CS146, Spring, 2018
Programming Assignment III

Due at 11:59pm, on Monday, May 14, 2018

Note:

This programming assignment is **for individual work only. 0 points will be given if there are similar report contents or codes.** You can discuss approaches with other persons, but it is not allowed to use someone's codes or copy codes from Internet. **Code plagiarism checker tool will be used to check the similarity of codes.** Please check on the University Policies in the syllabus. You may be asked to explain your codes by instructor or to present your codes in the class.

Problem Statements

Based on the simulation of CPU execution queue from the Programming Assignment I, a CPU process might have waiting time in the queue based on the priority code.
For this programming assignment, you are a software engineer and are asked to develop a simulation software application for CPU to **dynamically** keep track of the process scheduling according to the sorted priority codes in the waiting queue. A process with a larger priority code will pre-empt others even if they have been waiting longer in the process queue.

Programming Requirements

1. The CPU process scheduling simulation software application MUST be written in Java and is required to implement Red-Black (RB) Tree, which pseudo codes are specified in textbook. (You MUST use the pseudo codes provided in textbook to write your Java codes. Any other codes will be treated as failing requirements and **will receive 0 points**.)
2. The CPU process waiting list always contains 20 processes with name of a process and a priority code. Each process will be randomly assigned a distinct priority code.
3. Your software application **MUST** at least contains the following functions:

a) Build up a Process RB Tree. The color property of each node must be presented. (**MUST follow RB Tree properties** specified in textbook and ppt slides. Your own tree structure will not be accepted.) (A better way to verify if your Process RB tree is correct or not, just **draw the RB tree**.)

b) Allow users to insert any process to the RB Tree based on its priority code. (**Must draw the RB tree** to show the result of the color property changes of process nodes)

c) Allow users to search a process's name by entering a priority code.

d) Allow users to delete any process from the RB Tree. (**Must draw the RB tree** to show the result of the color property changes of process nodes)

e) Allow users to make a sorted list of process based on the priority codes (**MUST** including names, priority code and color of each process node.)

4. Each java file/class/subroutine/function call **must** contain a header comment and explanation in the beginning of it. (Points will be taking off if no comments.)

Submission Requirements

1. The deadline to submit/upload your report and source codes to Canvas is 11:59pm on Monday, 5/14/2018. (**No Late submission**, **Please do not wait until the last minute to upload and submit.**)
2. Zip all your files into one zipped file with file name: **Your_Last_Name-PA3.zip**, Any file with incorrect file name will not be accepted.
3. You **MUST** write a report (**doc or pdf**) as much detail as possible with the following items included in the report:
   a) The key concepts of **your** RB Tree requirement specifications design.
   b) Explanations of Classes/subroutines/function calls and of each purpose.

c) **A lot of screen shots** of each simulation process/procedure including inputs and outputs. This is to verify your codes and to check to see if your codes match all the functional requirements. **(0 points will be given if self testing screen shots are not included in the report.)**

d) Each test of your program outputs MUST include process's names, priority codes, and **color codes**. (If testing outputs only show priority codes without process's names and color will receive 0 pints)

e) The procedure (step by step) of how to unzip your files, install your application and run/test your codes.

f) Problems encountered during the implementation.

g) Lesson Learned ("I learned RB Tress" is not acceptable.)

4. Grading is based on the full functioning, completeness of requirements and clarity of your codes and report.