[CS302-Data Structures] Homework 8: Red Black Trees

Instructor: Kostas Alexis
Teaching Assistants: Shehryar Khattak, Mustafa Solmaz
Fall 2018 Semester

Section 1. Left Leaning Red Black Trees

A left-leaning red–black (LLRB) tree is a type of self-balancing binary search tree. It is a variant of the red–black tree and guarantees the same asymptotic complexity for operations, but is designed to be easier to implement.

All of the red-black tree algorithms that have been proposed are characterized by a worst-case search time bounded by a small constant multiple of log N in a tree of N keys, and the behavior observed in practice is typically much multiple faster than the worst-case bound, close to the optimal log N nodes examined that would be observed in a perfectly balanced tree.

Specifically, in a left-leaning red-black ***2-3*** tree built from N random keys:

- A random successful search examines log2 N − 0.5 nodes.
- The average tree height is about 2 log2 N
- The average size of left subtree exhibits log-oscillating behavior.

Section 2. Exercise on using Left Leaning Red Black Trees (LLRBTs)

On the basis of the implementation example provided at http://www.teachsolaisgames.com/articles/balanced_left_leaning.html (uploaded also in the canvas announcement but mentioned here to acknowledge the author), you are requested to use LLRBTs with correspondence to 2-3 Trees (check relevant flag) and perform:

- Insertion of 10 random variables
- Removal of the fourth random variable you inserted

You are required to deliver:

- Code that implements the above and outputs to terminal the immediate parent (value and color) alongside the just inserted node (value and color) for each of the insertions you do (except from the first where there is no parent in any case),
- Terminal output presenting what is mentioned above and a document file (e..g, *.pdf) that presents your current tree.