Applications of Algorithms

Assignment 3

Randomly Generated Binary Search Trees and Red-Black Trees

(A) In the first part of this assignment you will investigate the claim in Theorem 12.4 that a randomly built binary search tree on n distinct keys has expected height $\mathcal{O}(\log n)$. In addition, you will investigate the time taken to build and destroy binary search trees.

You must do the following:

(i) Code up the TREE-INSERT and TREE-DELETE algorithms from Chapter 12 of the textbook.

(ii) Run experiments to build binary search trees from randomly shuffled lists of keys by repeatedly calling TREE-INSERT. Record the height of each tree built. Run experiments for different values of n (the number of keys) to illustrate the asymptotic growth of the height as n increases. To get the 'expected height' of a randomly built binary search tree, repeat the experiment a number of times for each value of n and take the average height.

(iii) For each binary search tree constructed in part (ii), record also the time taken to build the tree and plot your results on a graph.

(iv) For each binary search tree constructed in part (ii), destroy the tree by repeatedly calling TREE-DELETE to remove all nodes. Record the time taken to destroy the binary search trees and plot your results on a graph.

(B) In the second part of this assignment, you will investigate the claim in Lemma 13.1 that a red-black tree with n internal nodes has height at most $2\log(n+1)$.

(i) Code up the RB-INSERT algorithm from Chapter 13 of the textbook. This will require other procedures from the textbook such as RB-INSERT-FIXUP, LEFT-ROTATE and RIGHT-ROTATE which must also be coded up.

(ii) Run experiments to build red-black trees from randomly shuffled lists of keys by repeatedly calling RB-INSERT. Record the height of each red-black tree built. Run your experiments for different values of n (the number of keys) to illustrate the asymptotic growth of the height as n increases. Repeat the experiment a number of times for each value of n and take the average height. Plot your results on a graph.

(iii) For each red-black tree constructed in part (ii), record also the time taken to build the tree and plot your results on a graph.

You must submit the following:

(1) Your source code for all the algorithms (**please note**: I would prefer C++ or Java).
(2) A document with the all the graphs required and a description of how the graphs were obtained (range of dimensions, key values, number of trees of each size, etc.).

You must submit your files to the AA Moodle page by **Monday 18 October** at **23h00**.