Datastructuren

October 21, 2009
Due Date: Monday, November 9

Programming Assignment No 4

This programming assignment is a continuation of the work you did in the last assignment. We will look in this assignment a) at the dynamics of BSTs randomly generated by pairs of insertions and deletions and b) the behavior of splay trees.

Again we assume the keys of the BSTs to be integers. The purpose of this experiment is to compute the Internal Path Length: IPL = $\sum(i-1)l_i$ where i ranges over all the levels of the tree and l_i is equal to the number of nodes in level i and the average path length: APL = IPL/n where n is the number of nodes in the generated tree and compare this to the APLs of perfectly balanced BSTs. (See also page 224 in Drozdek for the definitions of IPL and APL.)

In the following you can use your implementation in Assignment 3 of an algorithm for the random generation of permutations.

1. Generate 100 binary search trees on 500 keys (the first 500 natural numbers). For each of the binary search trees do 500^2 random insertion-deletion pairs. You can guide the insertion by randomly generating the permutation on 500 keys. The key to be deleted is randomly chosen out of the 500 available keys; if the key is found in the BST, then the tree and permutation are updated otherwise do nothing. Compute for each of the trees the APL – this means that you also have to keep track of the number of nodes in the generated trees. What relationship do you see between the number of keys and the APL? How does this compare to the APL of perfectly balanced BSTs? Each generated tree will give rise to a number of nodes and the APL for the tree. Plot the collection of all pairs (number of nodes in the tree, APL of th tree) in one color and also the pairs (number of nodes of the tree, APL for the perfectly balanced tree on the number of nodes) in another color by using, for instance, GnuPlot.

2. Argue whether searching is efficient in a BST without balancing or not based on your findings in Part 1. Discuss also other, sensible ways of analyzing the data obtained in Part 1 – for instance, for each of the 100 trees you could compute the ratio of the APL of the generated tree and the APL of a perfectly balanced tree on the same number of nodes as the generated tree, and further analyze these 100 ratios. Of course: carry out your proposed way of analyzing the data.

3. Generate a BST on 500 keys (again the first 500 natural numbers) where the permutation is the identity permutation. Carry out, say 2000, random searches in the tree. Assume that each of the searches/accesses is done in a splayed way. It is instructive to plot the BST after each of the accesses – do this at least after each of the 100 accesses. (If need be reduce the number of keys to the first 100 natural numbers in order to get feasible plots. Otherwise stick to 500 keys.) Compute the amortized cost of the 2000 searches. What strikes you about this number?

Turn your C++ code and the paragraphs of text with the interpretation of your results as a zipped file to Simon Zaaijer (email: szaaijer@liacs.nl) on or before Monday, November 9. Make sure both the C++ files and the interpretation contain the names of your team. Furthermore turn in a hardcopy of your files (C++ and text) in the Lecture of Monday, November 9 . You can work on this assignment in pairs. Refer questions to szaaijer@liacs.nl or deutz@liacs.nl.