University of Waterloo
CS240 - Fall 2015
Assignment 3

Due Date: Wednesday October 28 at 5pm

Please read `http://www.student.cs.uwaterloo.ca/~cs240/f15/guidelines.pdf` for guidelines on submission. All problems are written problems; submit your solutions electronically as a PDF with file name `a03wp.pdf` using MarkUs. We will also accept individual question files named `a03q1w.pdf`, `a03q2w.pdf`, `a03q3.pdf`, `a03q4w.pdf`, `a03q5w.pdf`. if you wish to submit questions as you complete them.

There are 62 possible marks available. The assignment will be marked out of 60.

Problem 1 AVL Trees [6+6+6+8=26 marks]

a) Consider the AVL tree shown in Figure 1. Draw the tree again by adding the balance factors to all nodes (similar to slide 17 of Module 4).

Perform the operation Insert(5) on the tree. Draw the tree before each single or double rotation, and draw the final tree.

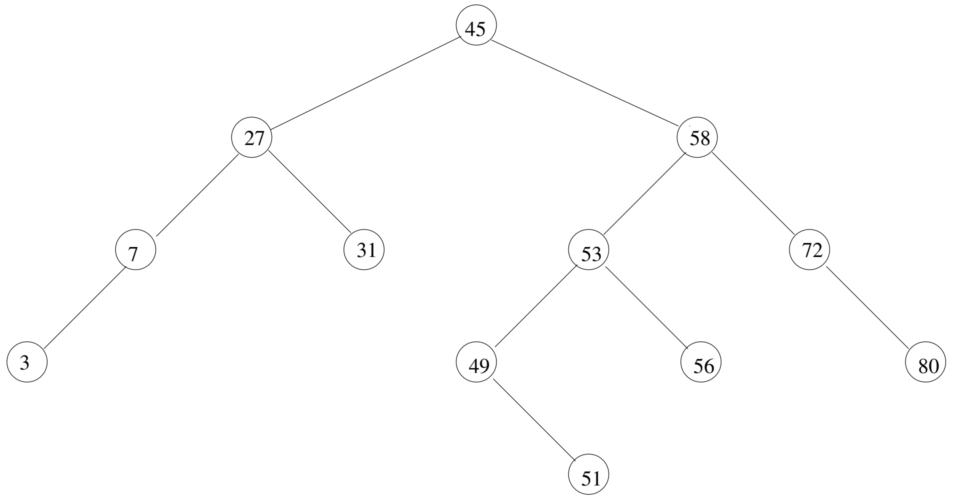

Figure 1: AVL tree of problem 1.

b) Consider the tree in Figure 1. Perform the operation delete(27) on the tree, swapping with the inorder successor. Draw the tree before any single or double rotation is performed, with the balance factors updated up until any call to fix is required. Draw the final tree with balance factors.

c) We define *foo trees* as follows. A foo tree is a binary search tree where for every node, the heights of the left and right subtree differ by at most 2. Prove that a foo tree with n nodes has height $O(\log n)$.

d) Describe an efficient algorithm for computing the height of a given AVL tree. Your algorithm should run in time $O(\log n)$ on an AVL tree of size n. In the pseudocode, use the following terminology: T.left, T.right, and T.parent indicate the left child, right child, and parent of a node T and T.balance indicates its *balance factor* (-1, 0, or 1). For example if T is the root we have T.parent=nil and if T is a leaf we have T.left and T.right equal to nil. The input is the root of the AVL tree. Justify correctness of the algorithm and provide a brief justification of the runtime.

Problem 2 AVL Trees [4+4=8]

Consider the binary tree T in Figure 2.

a) Draw the tree that results by calling $fix(T)$. Note that the balance factors are shown in the lower part of the nodes.

b) Draw the balance factors in the new tree.

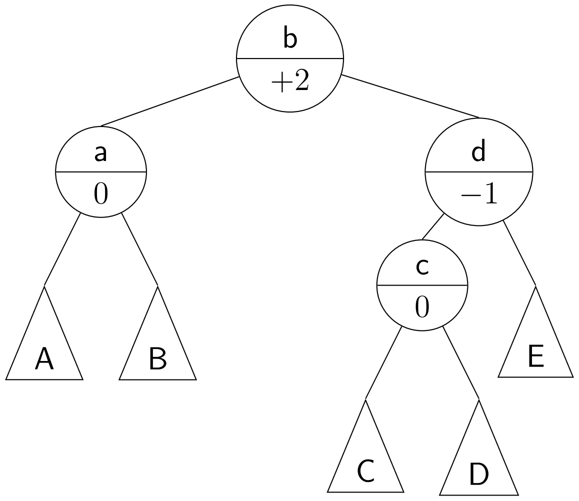

Figure 2: Binary tree T of problem 2.

Problem 3 2-3 Trees [2+2+2+4=10 marks]

a) [2 marks] Show the result of inserting $10, 6, 11$ into an initially empty 2-3 tree.

b) [2 marks] Show the result of inserting $10, 6, 11, 8, 5, 3$ into an initially empty 2-3 tree.

c) [2 marks] Finally, insert $10, 6, 11, 8, 5, 3, 7, 9, 4$ into an initially empty 2-3 tree.

d) [4 marks] Suppose that the keys $1, 2, \ldots, n$ are inserted into a 2-3 tree. Give an exact closed form formula for the tree height h in terms of n.
Hint: Consider the structure of the tree directly after the root splits.

Problem 4 B-Tree [4+2+4=10 marks]

a) [4 marks] Using the recipe described in class, insert the following keys, in the order given, into an initially empty B-tree of minsize two: 34, 4, 8, 5, 40, 11, 6, 12, 16, 21, 7, 9. Show the B-tree after every three insertions: four pictures in total.

b) [2 marks] Draw a new B-tree of minsize 2 that has height 2 and contains the keys $\{1, 2, \ldots, n\}$, for minimum n.

c) [4 marks] Remove the keys $1, 2, \ldots, n$, in the order given, from the answer to part b) of this question. Draw the resulting B-tree after every four removals.

Problem 5 Range cardinality [8 marks]

You are given an array A of n integers such that $0 \leq A[i] < k$ for all $0 \leq i < n$. Describe a preprocessing algorithm that runs in time $O(n + k)$ and creates a data structure that allows queries of the following type to be answered in constant time: How many integers in A are in the range $[a, b]$, given integers a and b with $0 \leq a \leq b < k$?

Describe how to answer the query. Provide a brief justification for the running time of both your preprocessing algorithm and your algorithm to answer the query.