Chapter 13: Red-Black Trees

Arne Andersson
A 2-4 tree,
Each node has degree between 2 and 4
The 2-4 tree, each node is now implemented as a small binary tree.
The 2-4 tree, each node is now implemented as a small binary tree

Sometimes called red-black tree
General idea:
Maintain a binary search tree
high-degree pseudo-nodes, each pseudo-node
is implemented as a binary search tree

All leafs have the same depth

Each pseudo-node is restricted in size within a factor of 2,
and must be well balanced
Insertion, 2 cases:

(a) A pseudo-node gets unbalanced, we balance it, DONE

(b) A pseudo-node gets too large, we split it, continue up
Deletion, four cases:

A pseudo-node gets unbalanced, we balance it, DONE

A pseudo-node gets too small, make a rotation to find a sibling, continue to (c)

A pseudo-node gets too small and has a sibling, co-balance the two, DONE

A pseudo-node gets too small and the sibling is of minimal size, join the two, continue up
Red-Black trees (or symmetric binary B-trees), main properties

- Complicated to implement
- Only $O(1)$ rotations (worst-case) per update, important in some applications
Binary Search Trees of Almost Optimal Height

Arne Andersson*
Department of Computer Science
Lund University
Box 118
S-221 00 Lund
Sweden

Christian Icking  Rolf Klein
Universität- Gesamthochschule Essen
FB 6 / Praktische softwareorientierte Informatik
Schützenbahn 70
4300 Essen 1
Federal Republic of Germany

Thomas Ottmann†
Institut für Informatik
Universität Freiburg
Rheinst. 10-12
D-7800 Freiburg
Federal Republic of Germany