Overview

- Red-Black Trees
- B-Trees

Closer Related Alternatives

- An Example of Multiple Insertions
- An Implementation of Insert

2-3-4 Trees
between 15 and 30 for a billion nodes
between 10 and 20 for a million nodes

- Tree Height:
- All paths from the top to the bottom are of equal length (tree height)
- Trees grow up from the bottom.

- Key Points:

2-3-4 Trees
All non-leaf nodes (except the root) have between $W/2$ and $W$ subtrees.

Insert: Split full nodes on the way down.

Extend (trivially) to $W < 4$.

The top-down find and insert algorithms that we used for $2 - 3 - 4$ trees:

- Allow $2$ to $W$ subtrees per node.
- Allow $1$ to $W - 1$ keys per node.

Generalize $2 - 3 - 4$ trees:

B-Trees
Red-Black Trees

* The "black" links bound the nodes of the original 2 - 3 - 4 tree.
* The "red" links bound together nodes as small binary trees.

The idea: represent 2 - 3 - 4 trees as standard binary trees (2-nodes only) by using one extra bit per node.

Red-Black Trees

April 12, 2000
that

— Thus, the tree is approximately balanced. The above properties ensure

twice as long as any other.

an empty node, these properties ensure that no such path is more than

By restricting the way nodes are colored on any path from the root to

same number of black nodes.

3. If a node is red, both its children are black.

4. Every path from a node to a descendant empty node contains the

2. Every empty node/tree is black.

1. Every node is either red or black, according to the color of the link

• A red-black tree satisfies the following properties:

Red-Black Trees (cont’d)
Red-Black Trees: An Example
Red-Black Trees: Find

- Use the (ordinary) BST find()!
A single or double rotation can eliminate the consecutive red nodes. However, if X's parent is red, this introduces two consecutive red nodes. Thus, the number of black nodes on the paths below X remain unchanged.

- On the way down, when we see a node X that has two red children, we make X red and its children black.
Red-Black Trees: Insert (cont'd)

- Goal: Eliminate consecutive red nodes without changing the number of black nodes along any path.

  - Single rotation + Coloring changes
or block nodes along any path.

- Goal: eliminate consecutive red nodes without changing the number

- Double rotation + Coloring changes

Red-Black Trees: Insert (cont'd)