between 15 and 30 for a billion nodes
between 10 and 20 for a million nodes
best case: \( \log N \) (\( \approx \) 20-nodes)
worst case: \( \log N \) (\( \approx \) 22-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
subtrees.

\( W / 2 \) and \( W \) are non-leaf nodes (except the root) have between

- Insert: Split full nodes on the way down

\( W < 4 \). Extend (trivially) to \( 4 \).

The top-down find and insert algorithms that we used for \( 2 - 3 \) trees

- Allow 2 to \( W \) subtreess per node.
- Allow 1 to \( W \) - 1 keys per node.

Generalize \( 2 - 3 \) - 4 trees.

B-Trees
Red-Black Trees

Together.

The "black" links bound the nodes of the original 2 – 3 – 4 tree by "red" links.

Idea: represent 3- and 4-nodes as small binary trees bound together one extra bit per node.

Represent 2 – 3 – 4 trees as standard binary trees (2-nodes only) by using
Thus, the tree is approximately balanced.

- Twice as long as any other.
- An empty node, these properties ensure that no such path is more than twice as long as any other.

By restricting the way nodes are colored on any path from the root to the same number of black nodes.

4. Every path from a node to a descendant empty node contains the same number of black nodes.

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

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)
In the following diagram, black nodes are drawn with a thick circle, and red nodes are drawn with a thin circle. I omit the empty (black) nodes.

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.

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
of black nodes along any path.

- Goal: eliminate consecutive red nodes without changing the number

  • Single rotation + Coloring changes

Red-Black Trees: Insert (cont'd)
of black nodes along any path.

- Goal: Eliminate consecutive red nodes without changing the number

  Double rotation + Coloring changes

Red-Black Trees: Insert (cont’d)