**2-3 Trees**

AVL and Red-Black trees are specialized bst.

2-3 trees are not bst, just search tree.

2-3 tree is made up of nodes designated as either 2-nodes or 3-nodes

2-node is the same as a bst node:

* it contains a data field and references to 2 child nodes
* one child node contains data less than the node’s data value
* the other child contains data greater than the node’s data value

3-node:

* contains 2 data fields, ordered so that 1st is less than the 2nd, and references to 3 children:
  + One child contains data values less than the 1st data field
  + One child contains data values between the 2 data fields
  + One child contains data values greater than the 2nd data field

All the leaves of a 2-3 tree are at the lowest level

Diagram, shape

Description automatically generated

If there is no node 5, it is not a 2-3 tree.

x should be less than y.

Table

Description automatically generated with medium confidence

T(n) = O(h) 🡺 <= h <= ---> we can say h = (logn)

Inserting an Item into a 2-3 Tree

2-3 tree maintains balance by being built from the bottom up, not the top down

Instead of hanging a new node onto a leaf, we insert the new node into a leaf

Running time is logarithmic (theta).

Diagram

Description automatically generated with low confidence

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

A picture containing text, clock

Description automatically generated

A picture containing text, clock

Description automatically generated

A picture containing text, clock

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Since a node with 3 values is a virtual node, move the middle value up and split the remaining values into 2 nodes.

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Text

Description automatically generated

“if the root is null” condition happens only if tree is really empty. At before, we have considered we have reached a leaf node that is empty but here “if the root is null” condition only happens when the overall tree is empty.

EXAMPLE: Create a 2-3 tree using the words : “The quick brown fox jumps over the lazy dog”







Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

A picture containing text, clock

Description automatically generated

Diagram

Description automatically generatedDiagram

Description automatically generated

Analysis of 2-3 Trees and Comparison with Balanced Binary Trees

2-3 trees don’t require the rotations needed for AVL and Red-Black trees

The number of items that a 2-3 tree of height h can hold is between -1 (all 2 nodes) and -1 (all 3 nodes)

Therefore, the height of a 2-3 tree is between and

The search time is O(logn)

Removal from a 2-3 Tree

Removing an item from a 2-3 tree is generally the reverse of the insertion process

If the item to be removed is in a leaf, simply delete it

If it’s not in a leaf, remove it by swapping it with its inorder predecessor (or inorder successor) in a leaf node and deleting it from the leaf node

If removing a node from a leaf causes the leaf to become empty,

* items from the sibling and parent can be redistributed into that leaf
* or the leaf can be merged with its parent and sibling nodes

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated with medium confidence

Diagram

Description automatically generated

inorder predecessor of 11 is 9, inorder successor of 11 is 15.

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated with medium confidence

Diagram

Description automatically generated

Diagram

Description automatically generated

At most we follow a path from root to a leaf so remove operation will be (logn) as insertion.

We always delete from a leaf. So to be able to find that leaf, we have to have logarithmic time.