2-3 tree

Background

A 2-3 tree is a multi-way search tree in which each node has two children (referred to as a two node) or three children (referred to as a three node). A two node contains one element. The left subtree contains elements that are less than the node element. The right subtree contains elements that are greater than the node element. However unlike a binary search tree a two node can have either no children or two children it cannot have just one child. A three node contains two elements, one designated as the smaller element and one designated as the larger element. A three node has either no children or three children. If a three node has children, the left subtree contains elements that are less than the smaller node element. The right subtree contains elements greater than the larger node element. The middle subtree contains elements that are greater than the smaller node element and less than the larger node element. Due to the self balancing effect of a 2-3 tree all the leaves are on the same level. A 2-3 tree of size N has a search time complexity of O(log N).

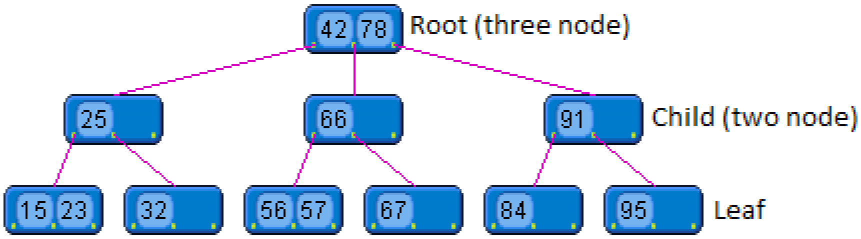

Figure 1: Parent/child nodes

Inserting elements into a 2-3 tree

All insertions in a 2-3 tree occur at the leaves of the tree. The tree is searched to determine where the new element will go, then it is inserted. The process of inserting an element into a 2-3 tree can have a ripple effect on the structure of the rest of the tree. Inserting an element into a 2-3 tree can be divided into three cases.

1. The simplest case is that the tree is empty. In this case a new node is created containing the new element. The node is then designated as the root of the tree.
2. The second case occurs when we want to insert a new element at a leaf that is a 2-node. In this case the new element is added to the 2-node making it a 3-node.
3. The third insertion situation occurs when we want to insert a new element at a leaf that is a 3-node. In this cases the 3-node is split and the middle element is moved up a level in the tree and the insertion process is repeated. When the root of the tree is split, the height of the tree increases by one.

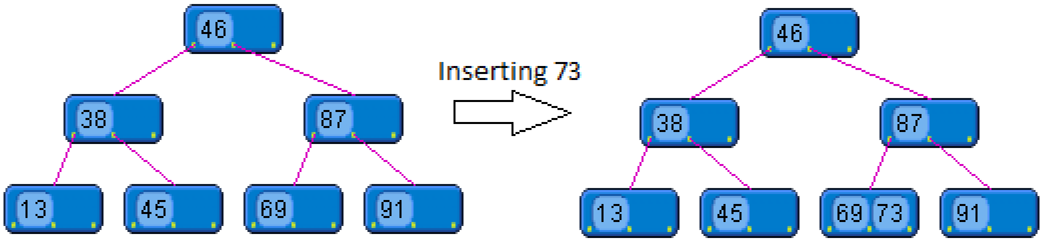

Figure 2: Inserting a new element at a leaf that is a 2-node.

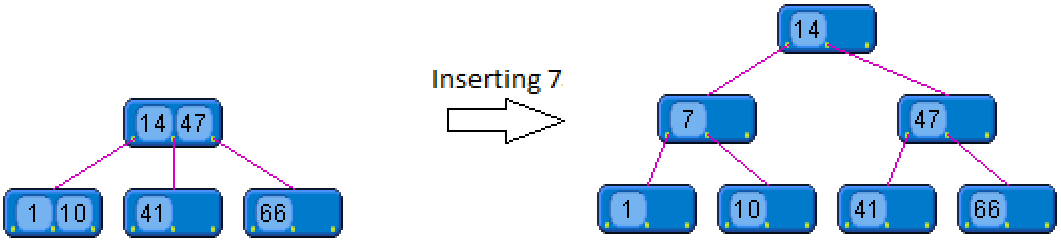

Figure 3: Inserting a new element at a leaf that is a 3-node.

Deleting elements from a 2-3 tree

Deleting elements from a 2-3 tree is also made up of three cases.

1. The simplest case is that the element to be removed is in a leaf that is a 3-node. In this case, removal is simply a matter of removing the element from the node.
2. The second case is that the element to be removed is in a leaf that is a 2-node. This condition is called underflow and creates a situation in which we must rotate or merge nodes in order to maintain the properties of the 2-3 tree.
3. The third case is that the element to be removed is in an internal node. In this case, we can simply replace the element to be removed with its inorder successor. The inorder successor of an internal element will always be a leaf element. After replacement we can simply remove the leaf element using the first or second case.

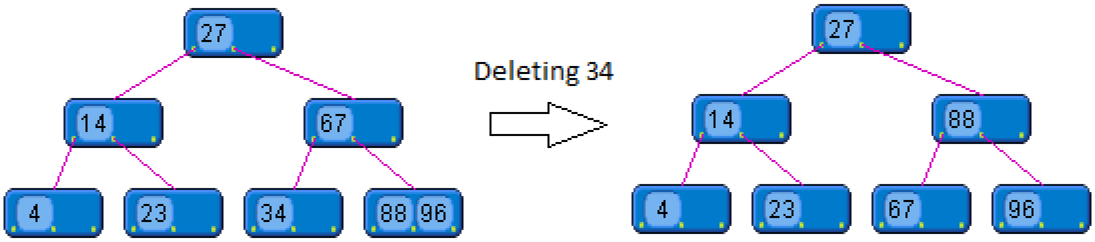

Figure 4: Deleting element from a 2-node, local rotation performed to fix the tree.

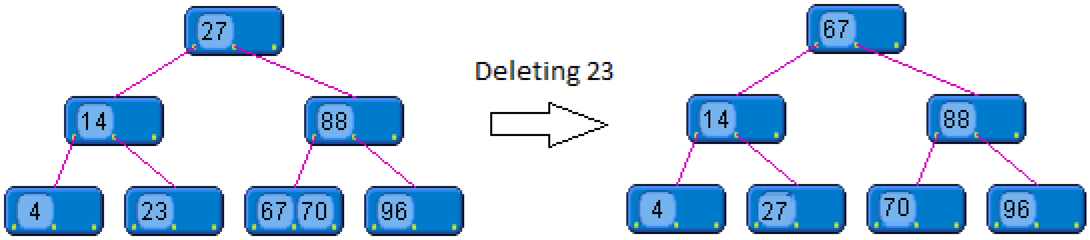

Figure 5: Deleting element from a 2-node, global rotation performed to fix the tree.