Chapter 6. Transform and Conquer

2-3-4 Trees, Red-Black Trees

1. Introduction

Binary Search trees are a convenient search structure. One great advantage is that the inorder traversal produces a sorted list. Efficiency depends of the tree’s height: \( \lfloor \log_2 n \rfloor \leq h \leq n-1. \)

A disadvantage of BSTs is their worst case efficiency - \( \Theta(n) \). This happens when the tree is very unbalanced.

Several representations have been developed to overcome the disadvantages of BSTs – AVL trees, multi-way search trees - 2-3 trees, 2-3-4 trees, red-black trees.

A multiway search tree is a search tree that allows more than one key in the same node of the tree.

Below we discuss one particular type of multi-way search trees – the 2-3-4 trees

2. 2-3-4 Trees

2.1. Definition

1. Three types of nodes:
   2-node: contains one key, has two links
   3-node: contains 2 ordered keys, has 3 links
   4-node: contains 3 ordered keys, has 4 links
2. All leaves must be on the same level, i.e. the tree is perfectly height-balanced. This is achieved by allowing more than one key in a node
Example

2.2. Operations

Search – straightforward: start comparing with the root and branch accordingly

Insert:

The new key is inserted at the lowest internal level

cases:

a. insert in a 2-node. The 2-node becomes a 3-node
b. insert in a 3-node. The 3-node becomes a 4-node

\[ R \rightarrow MP \quad \rightarrow \quad MP R \]

\[ \begin{array}{c}
\text{\ small square boxes indicate keys} \\
\end{array} \]

c. insert in a 4-node.

1. **Bottom-up insertion**: The 4-node is split, and the middle element is moved up – inserted in the parent node. The process is called **promotion** and may continue up the top of the tree.
   If the 4-node is a root (no parent), then a new root is created.
   After the split the insertion proceeds as in the previous cases.

\[ \begin{array}{c}
C \rightarrow N \\
\text{\ small square boxes indicate keys} \\
\end{array} \]

\[ \begin{array}{c}
\text{\ small square boxes indicate keys} \\
\end{array} \]

2. **Top-down insertion**

In our way down the tree, whenever we reach a **4-node**, we break it up into two **2-nodes**, and move the middle element up into the parent node. In this way we make sure there will be place for the new key.

**Complexity of search and insert operations**: $O(\log N)$
3. Red-Black Trees

Advantages of 2-3-4 Trees
Balanced
O(log N) search time

Disadvantages:
Different node structures

A red-black tree is a binary search tree with the following properties:

• edges are colored red or black
• no two consecutive red edges on any root-leaf path
• same number of black edges on any root-leaf path (=black height of the tree)
• edges connecting leaves are black

A red-black tree corresponds to a 2-3-4 tree in the following way:

<table>
<thead>
<tr>
<th>2-3-4 tree</th>
<th>red-black tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-node</td>
<td>2-node</td>
</tr>
<tr>
<td>3-node</td>
<td>two nodes connected with a red link (left or right)</td>
</tr>
</tbody>
</table>
Summary:

- 2-3-4 tree
- Red-black tree

Complexity of search and insert: \(O(\log N)\)