1 B-Trees

Definition 1. A B-tree $T$ is a rooted tree (with root $\text{root}\ [T]$) having the following properties:

1. Every node $x$ has the following fields:
   (a) Each node will have format $O\ (B)$;
   (b) $n\ [x]$ is the number of keys currently stored in node $x$;
   (c) The $n\ [x]$ keys themselves stored in nondecreasing order:
      \[ \text{key}_1\ [x] \leq \text{key}_2\ [x] \leq \cdots \leq \text{key}_{n\ [x]}\ [x] \]
   (d) $\text{leaf}\ [x]$ is a boolean value which is TRUE if $x$ is a leaf and FALSE if $x$ is an internal node.
2. Every leaf has the same depth which is the tree’s height $h$.
3. There are lower and upper bounds on the number:
   (a) Except root, each node has $\geq \frac{B}{2}$ childern;
   (b) Root has $\geq 2$ childern, unless it is the only node in the tree.
4. If $x$ is an internal node, it also contains $n\ [x]+1$ pointers $c_1\ [x] , c_2\ [x] , \cdots , c_{n\ [x]+1}\ [x]$ to its childern. Leaf nodes have no children.
5. The keys $\text{key}_i\ [x]$ separate the ranges of keys stored in each subtree: if $k_i$ is any key stored in the subtree with root $c_i\ [x]$, then
   \[ k_1 \leq \text{key}_1\ [x] \leq k_2 \leq \text{key}_2\ [x] \leq \cdots \leq \text{key}_{n\ [x]}\ [x] \leq k_{n\ [x]+1} \]

Proposition 2. Given $N$ items in disk and block size $B$, the depth of B-tree is $O\ (\log_B N)$. 
2 Basic Operations on B-Tree

There are several related operations on B-tree: initialization, splitting, merging and inserting etc..

2.1 Initial Tree

First $B$ keys are all inserted in root node with non-decreasing order.

- Empty root node
- Insert $(k, v)$

\[
\begin{array}{cccc}
\emptyset & & k_1 & k_2 & k_3 & \cdots & k_B \\
\downarrow & \downarrow & \downarrow & \downarrow & \\
v_1 & v_2 & v_3 & v_B \\
\end{array}
\]

Figure 2.1: Initial Tree

2.2 Splitting B-Tree Node

**Definition 3.** Splitting of a full node $x$ (assume there are $2t - 1$ keys in the node) is dividing around the median key $key_{t} [x]$ into two nodes having $t - 1$ keys each. The median key moves up into $x$’s parent.

Figure 2.2 is an example about splitting of node.

2.3 Splitting the Root

If the node we are going to split is the root of a B-tree, then the tree will grows in height by one. And the procedure can be illustrated in Figure 2.3.
Based on the technique of splitting node and root, an insertion example can be implemented as follows:

![Figure 2.3: Splitting of Root](image)

2.4 Merging B-Tree Nodes

The strategy for merging B-tree nodes is:

- If target node has $\frac{B}{2}$ children, try to steal a child from sibling;
- If that fails, merge target and a sibling.
2.5 Merge Root

The merge of root is the inverse process of splitting the root. The process can be derived from Figure 2.3 and after the merging, the height is 1 less.

Figure 2.5: Deletion Example