Date: Aug 28, 2025

M-Way Search Tree

- M-way search tree has $m - 1$ values per node and m sub-tree structures.
- There are m pointers P_1, P_2, $\ldots$, P_m and the key values are k_1, k_2, $\ldots$, k_{m-1}.

Properties

- The key values in the sub-tree pointed to by P_i are less than k_i such that $0 \leq i \leq m-1$.
- The key values of the sub-tree pointed to by P_{i+1} are greater than k_i.

Properties of B-Tree

- Every node in a B-tree has at most m children.
- Every node in a B-tree except the root and the leaf nodes has at least $\frac{m}{2}$ children (this property reduces the height of B-tree).
- The root node has at least two children if it is not a terminal node.
- All leaf nodes are at the same level.

Searching

- Let x be the element to be searched.
- If x is less than the first element of the root, follow the left most pointer.
- Else search within the node using linear search.

Insertion

1. In a B-tree, all insertions are done at the leaf node.
2. Search the position at which the new key value should be inserted.
3. If the leaf is not full, i.e. it contains less than $m - 1$ key values, then insert the new element in the node, keeping the elements in the node ordered.
4. Otherwise
    a. Insert the new value in order into the existing set of keys.
    b. Split the node at its median into two nodes.

   c. Push the median element up to its parent node.
   d. If the parent node is already full, then repeat step 3b and 3c.

Deletion

1. Locate the leaf node which is to be deleted.
2. If it contains greater than $\frac{m}{2}$ keys (i.e. minimum number of key values), then just delete the value.
3. Else
   a. If left sibling has more than minimum number of key values, push the largest key into its parent node and pull down the intervening element from the node to the leaf node when the key is deleted.
   b. If right sibling has more than minimum number of key values, push the smallest key into its parent node and pull down the intervening element from the node to the leaf node when the key is deleted.
   c. If both the left and right siblings has minimum number of nodes, then combine the two leaf nodes to create a new leaf node and delete the key. If pulling the intervening element from the parent node leaves it with less than minimum number of key values in the node, then propagate the process upwards.

Delete an internal node

1. Promote the successor or predecessor of the key to be deleted to occupy the position of the deleted node.
2. Thus, predecessor or successor will always be in the leaf node so the processing will be as if a value from the leaf node is deleted.