B Trees

Eric McCreath

B-Trees

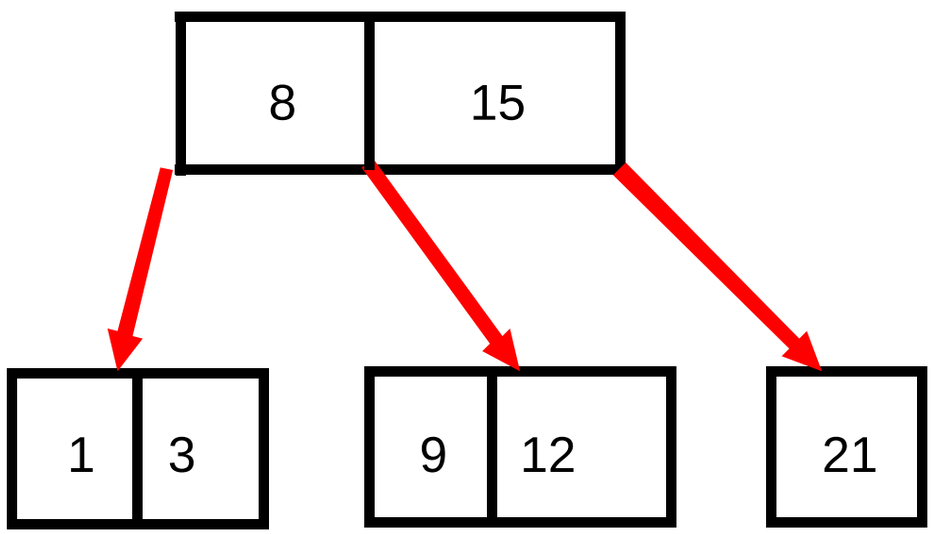
- B-Trees are a generalization of binary search trees. However, they are not binary trees. So rather than having a single key in each node more than one key can be placed in each node. These keys are ordered within the nodes and partition the key values of the subtrees. So if there are 4 keys in a node then there would be 5 subtrees.
- The number of keys within an inner node will be maintained in the range n to $2n$, so the number of subtrees a node will have will range between $n+1$ and $2n+1$. All the leaf node are kept at the same depth. This makes for good balance.

Why B-Trees?

- B-trees (and their variants) are used extensively for storing tables on hard disks as, in comparison to a binary search tree, they reduce the depth to lookup an element. This in tern reduces the number of disk blocks needed to be read.

B-Trees

The below tree is a b-tree whcih stores the set {1,3,8,9,12,15,21}.



Looking up elements within a b-tree can be implemented using using a recursive approach that is similar to that of a binary search tree. However, insertion and deletion are a little more tricky as one needs to maintain the constrains to help keep the tree balanced.