

Digital Search Tree

- A digital search tree is a binary search tree with one element in each node.
- The element to node assignment is determined by the binary representation of the element keys.
- All keys in the left subtree of a node at level i have bit $i = 0$ whereas those in the right subtree of a node at level i have bit $i = 1$.

Insertion

- Let k be the key to be inserted.
- If k is different from the root and bit 1 of k is 0, then follow its left child link.
 - If it is empty, place it over there.
 - Otherwise check for bit 2 and so on.
- Difference between insertion in DST and BST is that the search tree to move to is determined by a bit in the search key rather than by the result of the comparison of the search key and the key in the current node.
- Maximum depth of the tree is the number of bits in the numbers.

Deletion

- If the node is a leaf node, simply remove the leaf node.
- Else, replace the node by any of the leaf node in its subtree and then the leaf is removed.

Binary Trie

- A binary trie is a binary tree that has two kinds of nodes.
- A branch node and an element node.
- A branch node has two members – *left child* and *right child*. It does not contain any data.
- The element node has a single member – *data*.

Compressed Binary Trie

- Binary trie may contain branch nodes with one data member.

- If we add a data member – `bit_number` to each of the branch node, we can eliminate all degree 1 branch nodes of a binary trie and the modified trie is called a compressed binary trie.