**What is a B-Tree?**

An organizational structure for information storage and retrieval in the form of a tree in which all terminal nodes are the same distance from the base, and all nonterminal nodes have between *n* and 2 *n* subtrees or pointers (where *n* is an integer).

* Normal tree with more than one key
* Balanced search trees
* B-tree nodes may have many children
* They were designed to work well on Direct Access secondary storage devices (magnetic disks)
* Similar to red-black trees, but show better performance on disk I/O operations
* B-trees (and variations like B+ and B\* tress) are widely used in File Systems and Databases
* B-trees try to read as much information as possible in every disk access operation

*Properties*

1. Every node has at most m children
2. Every node (except root and leaves) has at least m/2 children
3. The root has at least two children
4. All leaf nodes are in the same level
5. An internal node with k children contains k-1 keys

*\*\*terminal nodes = leaf nodes, non-terminal nodes = internal nodes*

*Rules:*

* Rule 1: The root can have as few as one element (or even no elements if it also has no children); every other node has at least MINIMUM elements.
* Rule 2: The maximum number of elements in a node is twice the value of MINIMUM.
* Rule 3: The elements of each B-tree node are stored in a partially filled array, sorted from the smallest element (at index 0) to the largest element (at the final used position of the array).
* Rule 4: The number of subtrees below a nonleaf node is always one more than the number of elements in the node.

Subtree 0, subtree 1, ...

* Rule 5: For any nonleaf node:

An element at index i is greater than all the elements in subtree number i of the node, and

An element at index i is less than all the elements in subtree number i + 1 of the node.

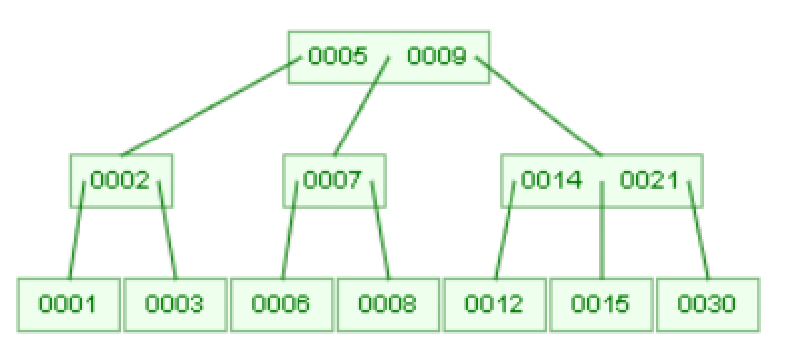
* Rule 6: Every leaf in a B-tree has the same depth. Thus it ensures that a B-tree avoids the problem of a unbalanced tree.

**Exercise 1:** Create a b-tree and insert the following:

30, 21, 5, 3, 6, 9, 12, 14, 7, 8, 2, 1, 15s

m = 2

Solution:



**Exercise 2:** Create a b-tree and insert the following:

C, D, F, R, T, W, Z, U, O, P, L, J, A, M, N, B

m = 2

Solution:

