B-Tree is a self-balancing search tree.

When the amount of data is very huge, the keys (data/elements) cannot be stored in the main memory. So the keys will be stored in disks.

It is well known that disk access time is very high compared to main memory access time. So we should optimise the tree so that the number of disk access required during insertion, deletion and search are reduced.

For most of the trees, each operation insert, delete or search require O(h) disk accesses where 'h' is the height of the tree.

The Idea of the B-tree is to keep the height as low as possible, which can be done by accommodating more number of keys in the Node.

Since the height is low for B-Tree, total disk accesses for most of the operations are reduced significantly compared to other balanced Binary Search Trees like AVL Tree, Red-Black Tree, ..etc.

Properties of B-Tree

1. All leaves are at same level.
2. A B-Tree is defined by the term minimum degree/order 'm' which determines maximum number of keys that can be stored.
3. All nodes except root must have at least [m/2]-1 keys and maximum of m-1 keys.
4. A non leaf node with n-1 keys must have n number of children.
5. If the root node is a non leaf node, then it must have at least 2 children.
6. All the key values within a node must be in Ascending Order

Note : A B-tree with order 3 is a 2-3 tree.
2-3 tree

A B-tree of order 3 is a 2-3 tree.

1. Every internal node i.e. non-leaf node will be a **2-node** or a **3-node**.
   - It the internal node is a 2-node if it has one data element and two children.
   - It the internal node is a 3-node if it has two data elements and three children.
2. A node with one value is either a leaf node or has exactly two children (non-null). Values in left subtree < value in node < values in right subtree
3.
4. A node with two values is either a leaf node or has exactly three children (non-null). Values in left subtree < first value in node < values in middle subtree < second value in node < value in right subtree.

5. All leaf nodes are at the same level of the tree