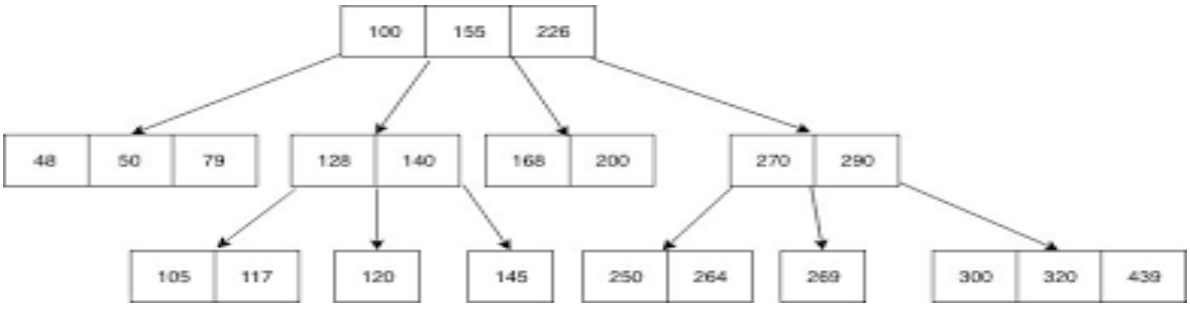
B-Tree

B Tree: B Tree is a specialized m-way tree that can be widely used for disk access. A B-Tree of order m can have at most m-1 keys and m children. One of the main reason of using B tree is its capability to store large number of keys in a single node and large key values by keeping the height of the tree relatively small.

Properties of B Tree

1. Every node in a B-Tree contains at most m children.
2. Every node in a B-Tree except the root node and the leaf node contain at least m/2 children.
3. The root nodes must have at least 2 nodes.
4. All leaf nodes must be at the same level.
5. A B-Tree is defined by the term minimum degree 't'. The value of t depends upon disk block size.
6. Every node except root must contain at least t-1 keys. The root may contain minimum 1 key.
7. All nodes (including root) may contain at most 2t – 1 keys.
8. Number of children of a node is equal to the number of keys in it plus 1.
9. All keys of a node are sorted in increasing order. The child between two keys k1 and k2 contains all keys in the range from k1 and k2.



Note:

When order(m) is given,

Maximum number of keys= m-1

Minimum number of keys = [m/2]-1

Applications of B Tree

1. B tree is used to index the data and provides fast access to the actual data stored on the disks.
2. Searching an un-indexed and unsorted database containing n key values needs O(n) running time in worst case. However, if we use B Tree to index this database, it will be searched in O(log n) time in worst case.

Why Uses B Tree

- Reduces the number of reads made on the disk
- B Trees can be easily optimized to adjust its size (that is the number of child nodes) according to the disk size
- It is a specially designed technique for handling a bulky amount of data.
- It is a useful algorithm for databases and file systems.
- A good choice to opt when it comes to reading and writing large blocks of data