

# ECAP770

ADVANCE DATA STRUCTURES

---

Ashwani Kumar  
Assistant Professor



# Learning Outcomes



After this lecture, you will be able to

- Understand B tree

# B Tree

- B Tree is a self-balancing data structure.
- For better memory efficiency user need to follow specific set of rules for searching, inserting, and deleting operations.
- B Tree is a m-way tree that can be used for disk access.

# B Tree

- A B Tree of order m can have at most  $m-1$  keys and m children.

or

- Each node can contain more than one key
- Each node can have more than two children

# B tree



# Why B Tree

- B-Trees is used to reduce the number of disk accesses.
- Most of the tree operations (search, insert, delete, max, min ) require  $O(h)$  disk accesses where h is the height of the tree. B-tree is a fat tree.
- The height of B-Trees is kept low by putting maximum possible keys in a B-Tree node.

# Why B Tree

- Data structures like binary search tree, avl tree, red-black tree, etc. can store only one key in one node. If you have to store a large number of keys, then the height of such trees becomes very large and the access time increases.
- The height of the B-tree is low so total disk accesses for most of the operations are reduced significantly compared to balanced Binary Search Trees like AVL Tree, Red-Black Tree, ..etc.

# B Tree properties

B-Tree of Order m has the following properties:

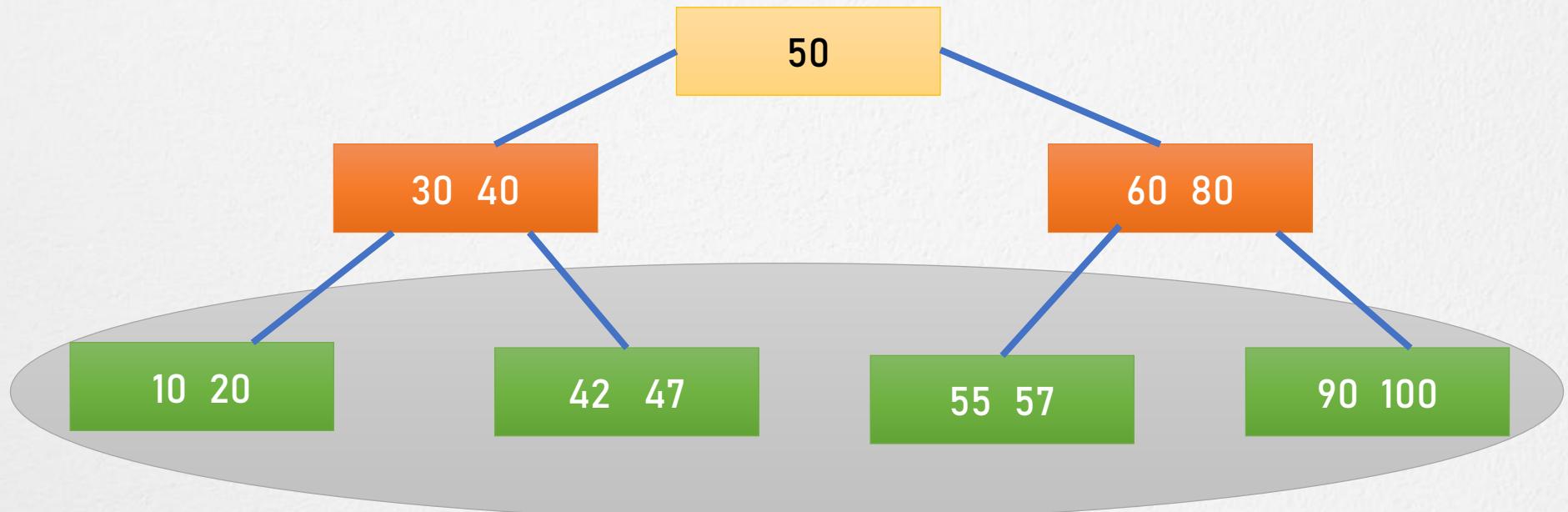
1. All leaf nodes must be at same level.
2. All nodes except root must have at least  $[m/2]-1$  keys and maximum of  $m-1$  keys.
3. All non leaf nodes except root (i.e. all internal nodes) must have at least  $m/2$  children.

# B Tree properties

4. If the root node is a non leaf node, then it must have at least 2 children.
5. A non leaf node with  $n-1$  keys must have  $n$  number of children.
6. All the key values in a node must be in Ascending Order.

# B Tree properties

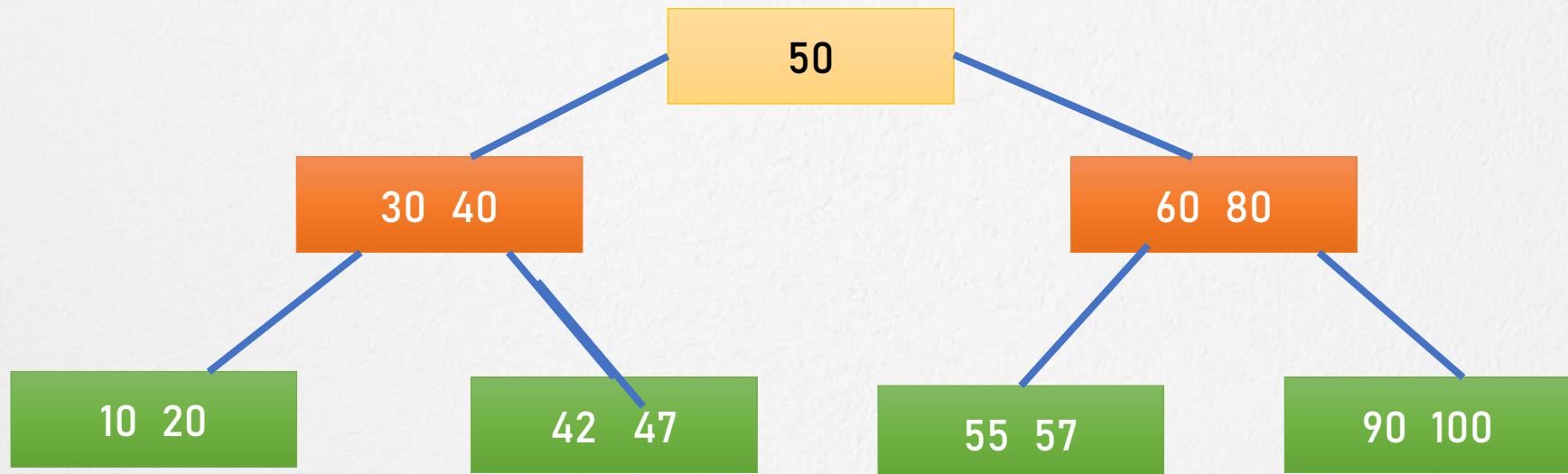
All leaf nodes must be at same level.



# B Tree properties

All nodes except root must have at least  $[m/2]-1$  keys and maximum of  $m-1$  keys.

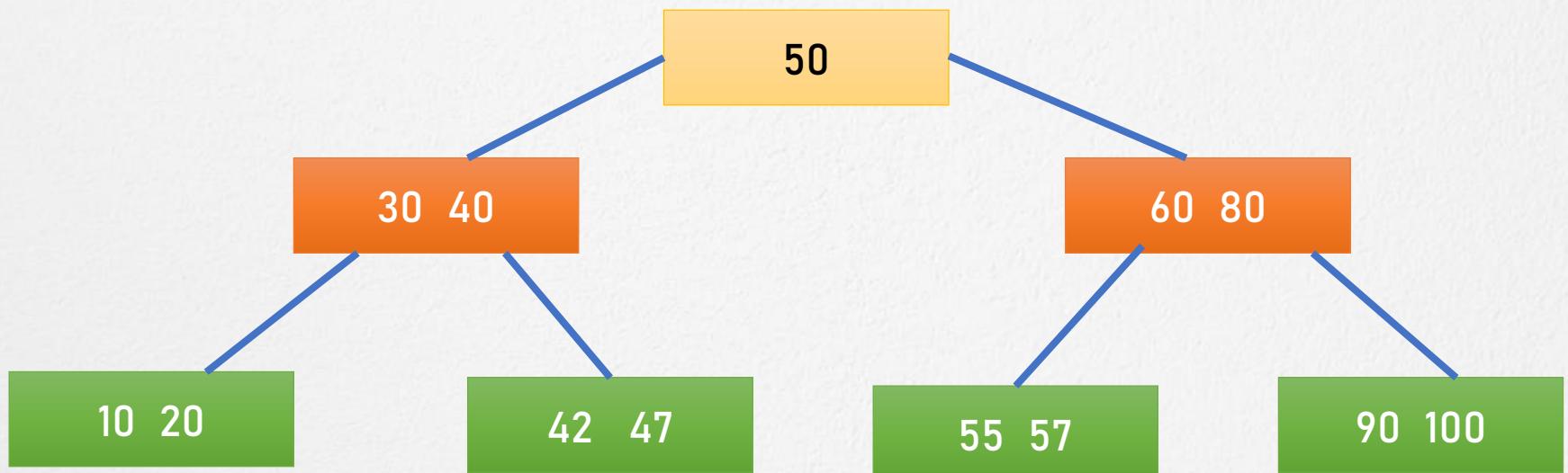
$m=4$



# B Tree properties

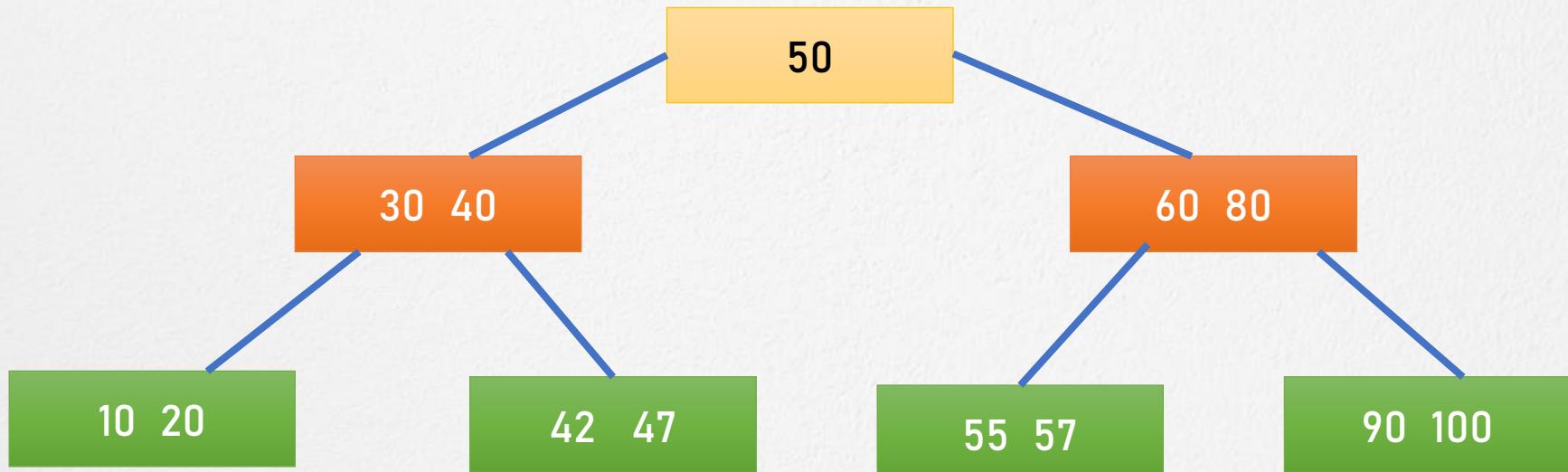
All non leaf nodes except root (i.e. all internal nodes) must have at least  $m/2$  children.

$m=4$ ,  $4/2=2$



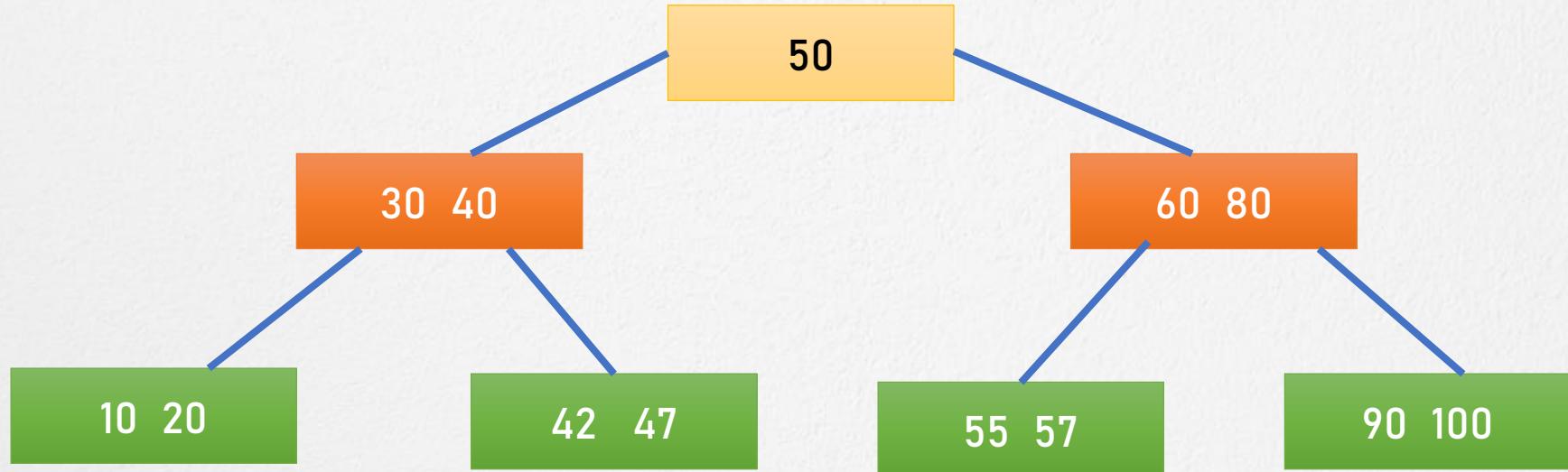
# B Tree properties

If the root node is a non leaf node, then it must have at least 2 children.



# B Tree properties

All the key values in a node must be in Ascending Order.



That's all for now...