CS 400

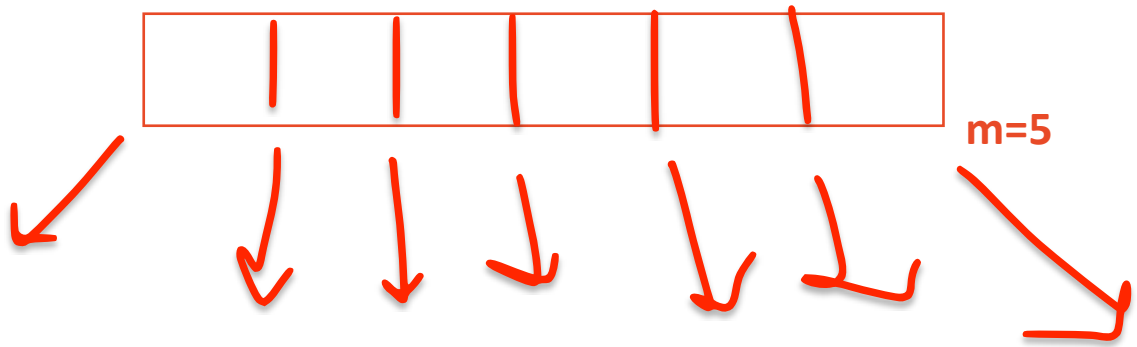
B-Tree Insert

ID: 08-02

B-Tree Insertion

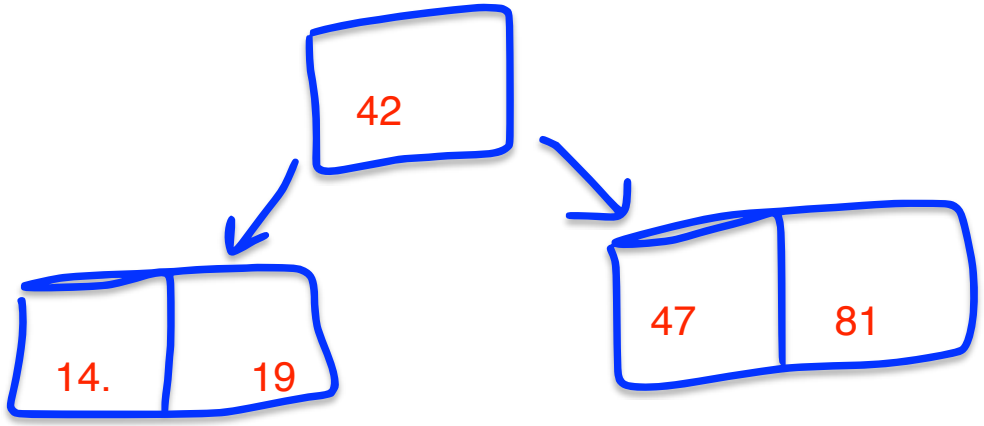
For a B-tree "of order **m**":

- All keys within a node are in sorted order.
  (Note: These are two different meanings for the word "order".)
- Each node contains no more than **m-1** keys.
- Each internal node can have at most **m** children,
  so a B-tree of order m is like an m-way tree.

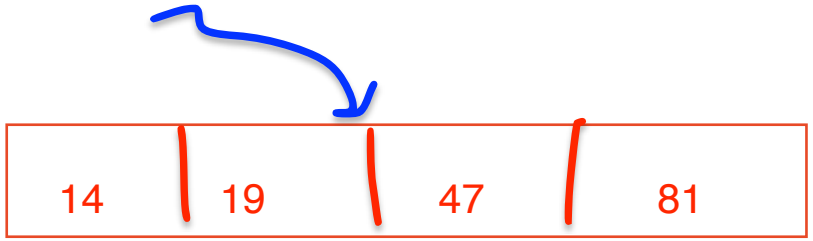


B-Tree Insertion

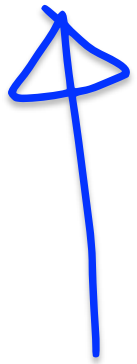
When a B-tree node reaches **m** keys:



14, 19, 42, 47, 81 을 노드에 넣는다고 하자.



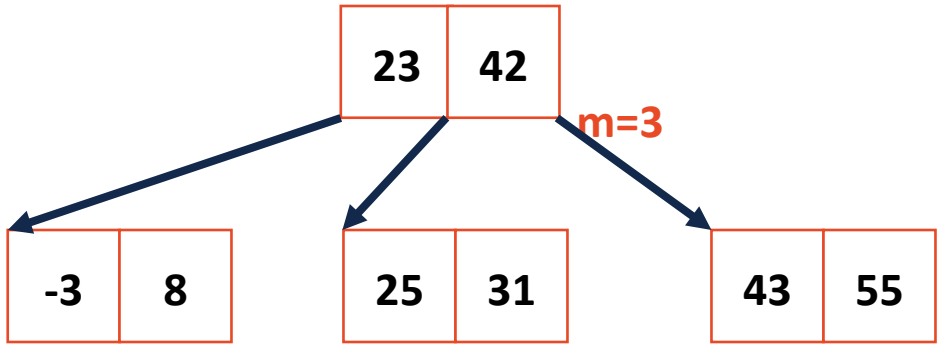
m=5



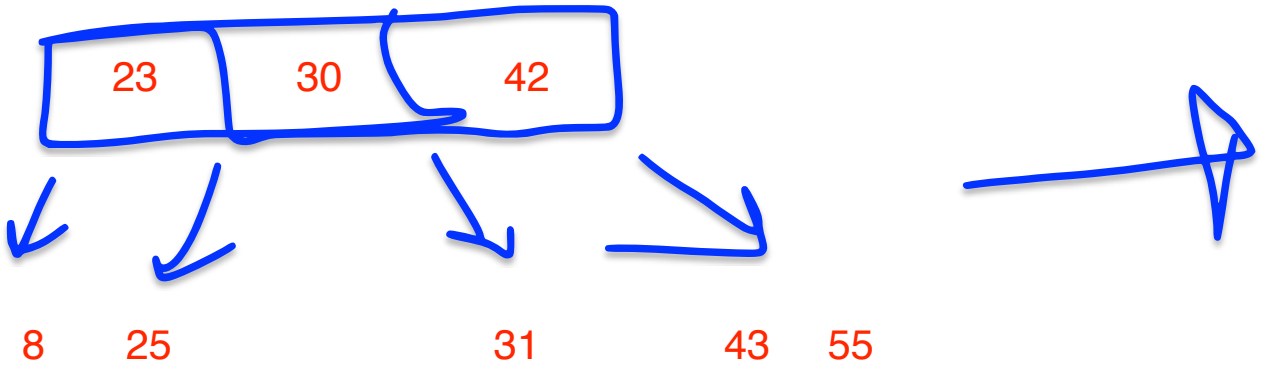
when we reach this point, we can't stck any more data in our root key node.
Instead, we need to go ahead and when we insert this value,
create a new node.

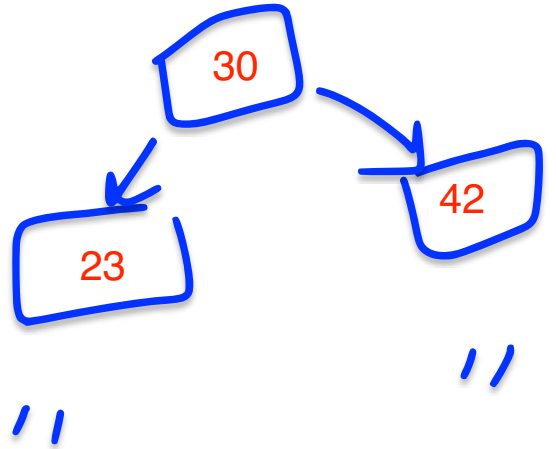
B-Tree Recursive Insert

Inserting 30 : it's between 23 and 42
and it's between 25, 31



node is full => throw it to the parent node





B-Tree Properties

For a B-tree "of order **m**":

1. All keys within a node are in sorted order.

2. Each node contains no more than **m-1** keys.

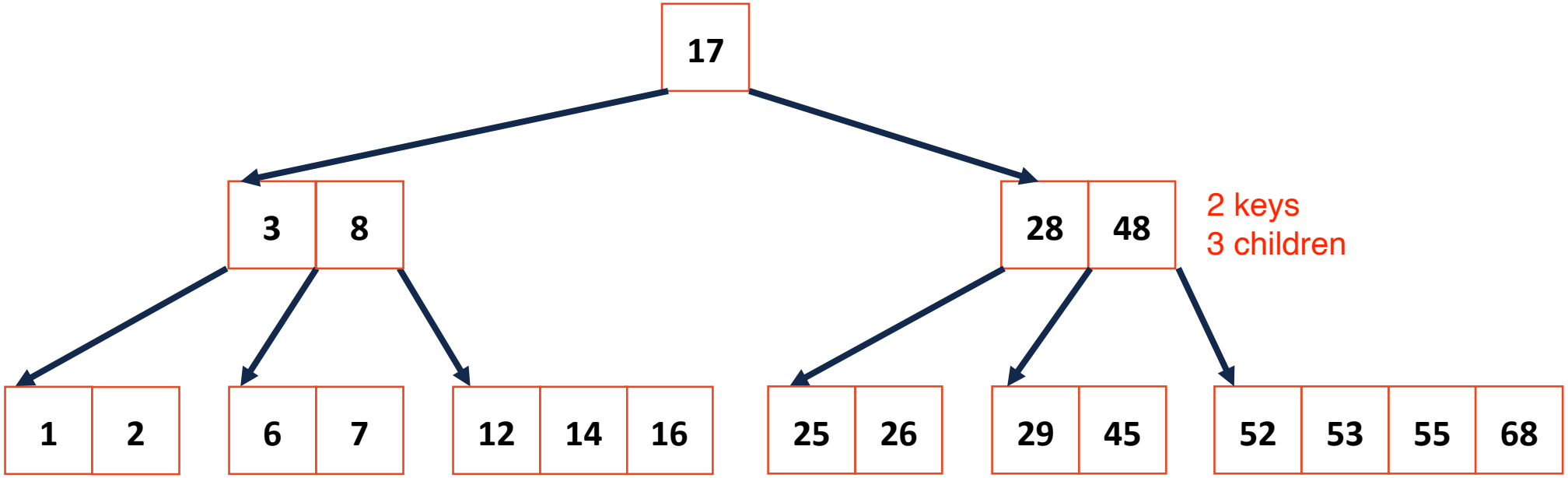
3. Each internal node has exactly **one more child than key**
   (at most **m** children, so a B-tree of order m is like an m-way tree).
   - A root node can be a leaf or have **[2, m]** children.
   - Each non-root, internal node has **[ceil(m/2), m]** children.

4. All leaves are on the same level.

BTree is always going to have the same height
no matter which path you take down the B-Tree

B-Tree

the order must be at least 5 &&
the order has to be bound between 3 and 6
=> 5 or 6



2 keys
3 children

because this leaf node has 4 keys, we
the order of the Btree must be at least