B+ Tree

- Data records are only stored in the leaves.
- Internal nodes store just keys.
- Keys are used for directing a search to the proper leaf.
- If a target key is less than a key in an internal node, then the pointer just to its left is followed.
- If a target key is greater or equal to the key in the internal node, then the pointer to its right is followed.
- B+ Tree combines features of ISAM (Indexed Sequential Access Method) and B Trees.

B+ Tree

- Implemented on disk, it is likely that the leaves contain key, pointer pairs where the pointer field points to the record of data associated with the key.
  - allows the data file to exist separately from the B+ tree, which functions as an "index" giving an ordering to the data in the data file.

B+ Tree

- Very Fast Searching
- Insertion and deletion are expensive.

$$\left\lceil \log_{\lceil \frac{p}{2} \rceil} N \right\rceil$$

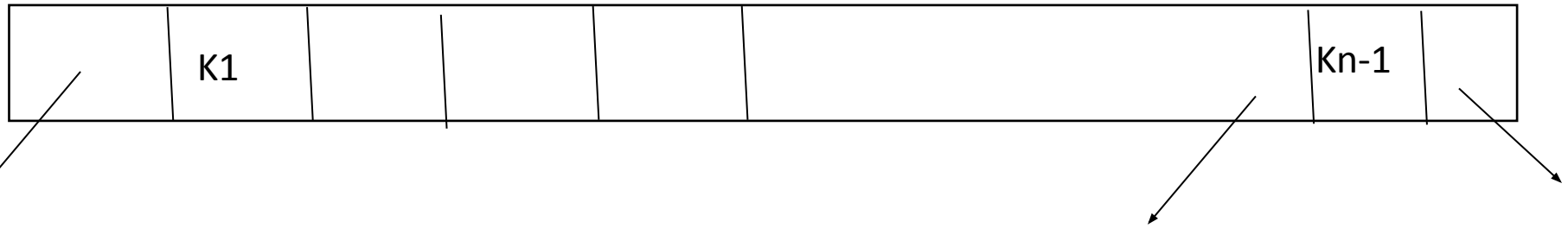
N number of search values
p order, number of block pointers per node

B+ Tree Structure

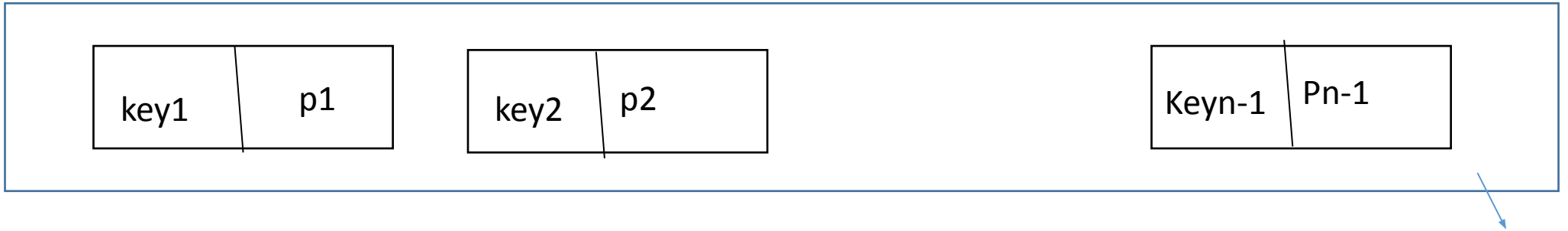
- Variation of B Tree.
- Structure of leaf nodes differs from internal nodes.
- Leaf nodes are linked together to provide ordered access on search field to records.
- Some search field values from leaf nodes are repeated in the internal nodes of B+ tree.

B+ Tree Structure

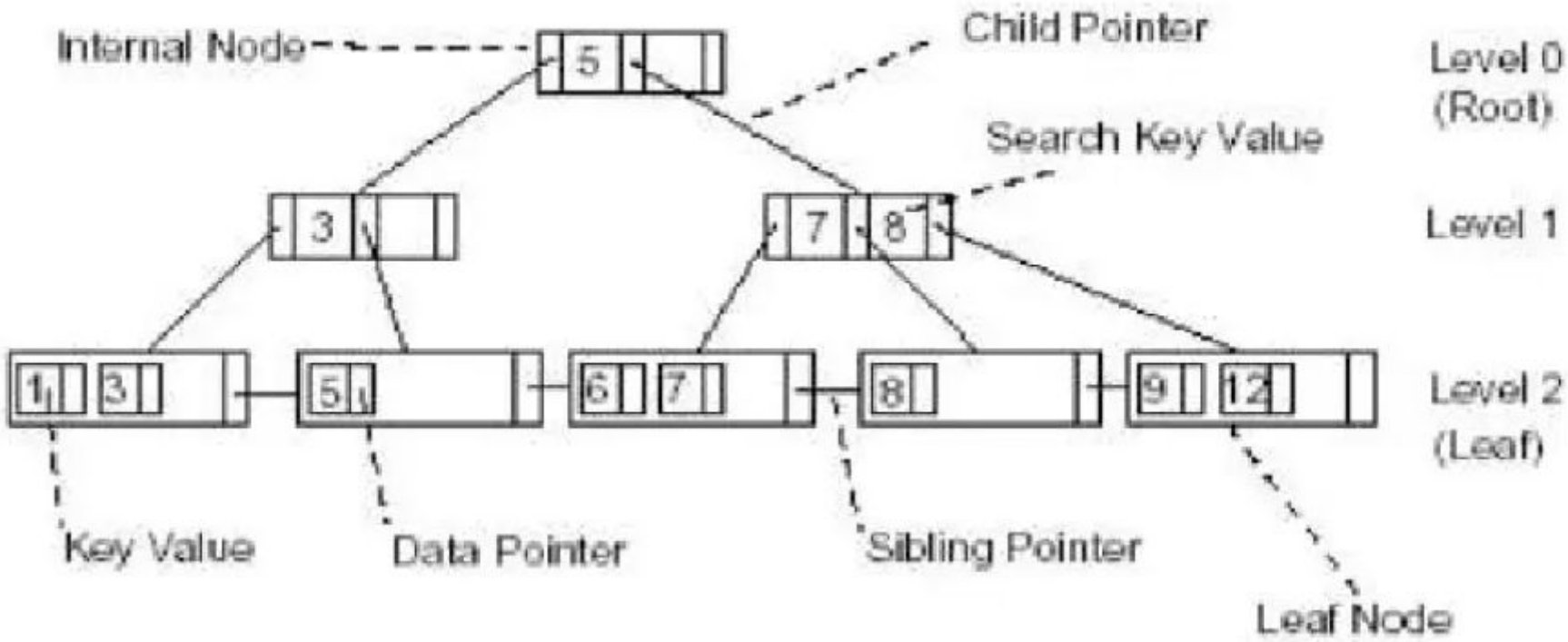
- Structure of internal Node



- p1
- Key≤k1 Key≤kn-1 Ke>kn-1
- Structure of Leaf Node

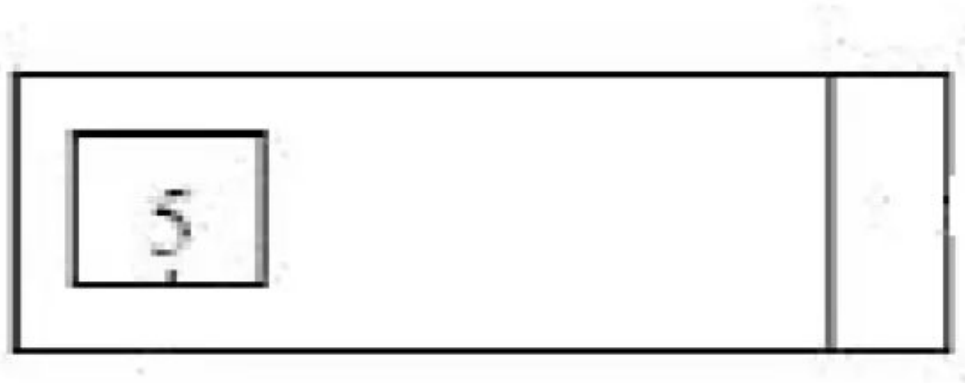


B+ Tree Structure



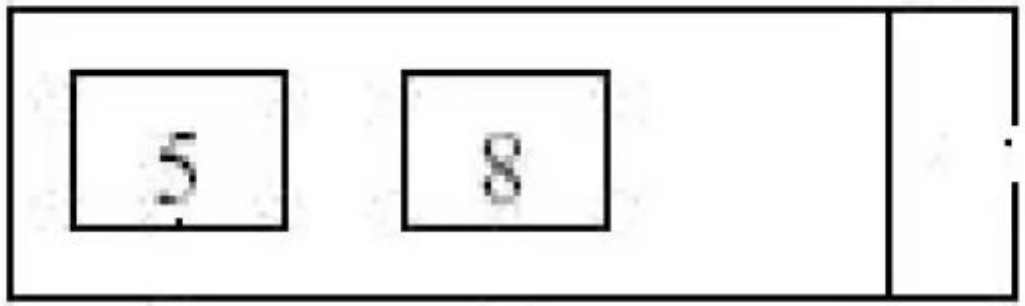
B+ Tree order 3 Insertion

- Insert value 5, 8, 1, 7
- Inserting value 5
- Since the node is empty, the value must be placed in the leaf node.



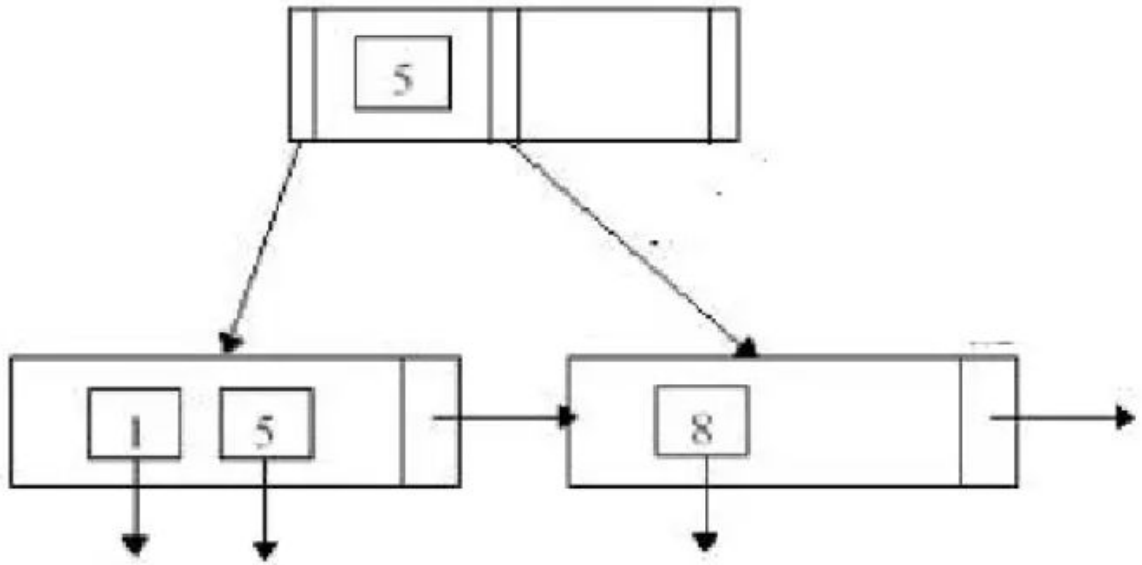
B+ Tree order 3 Insertion

- Inserting value 8
- Since the node has room, we insert the new value.



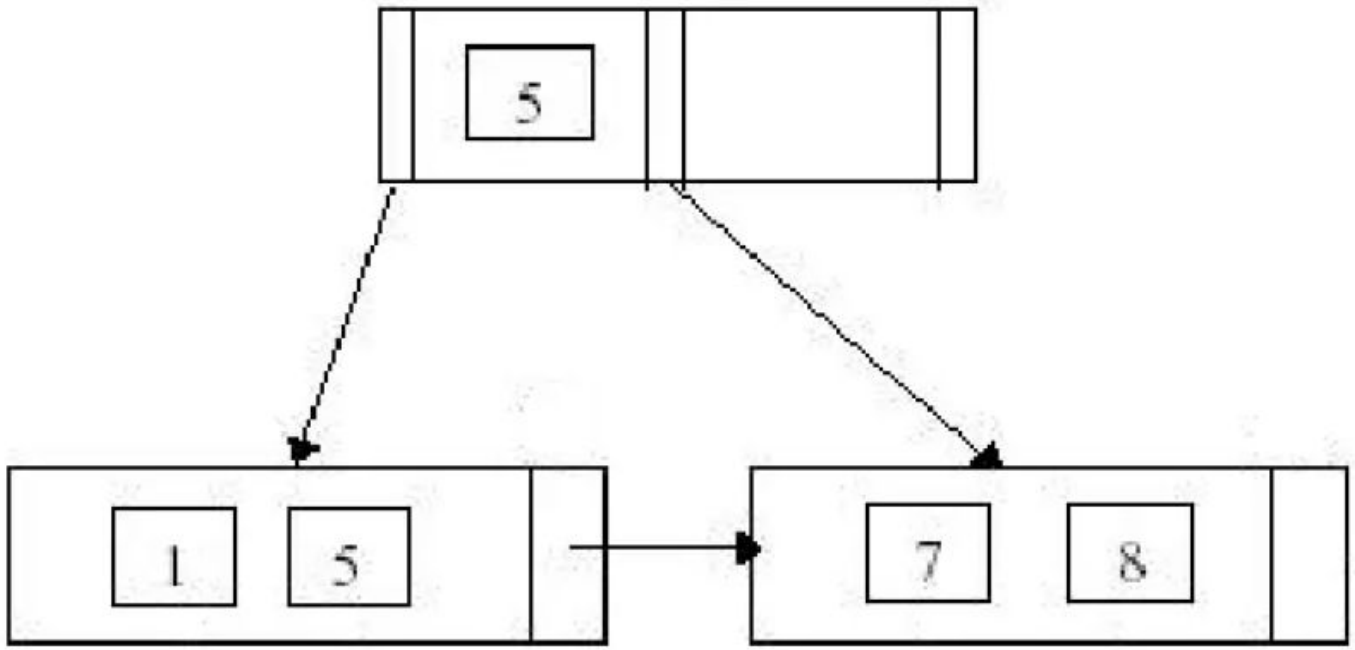
B+ Tree order 3 Insertion

- Insert value 1
- Since the node is full, it must be split into two nodes.
- Each node is half full.



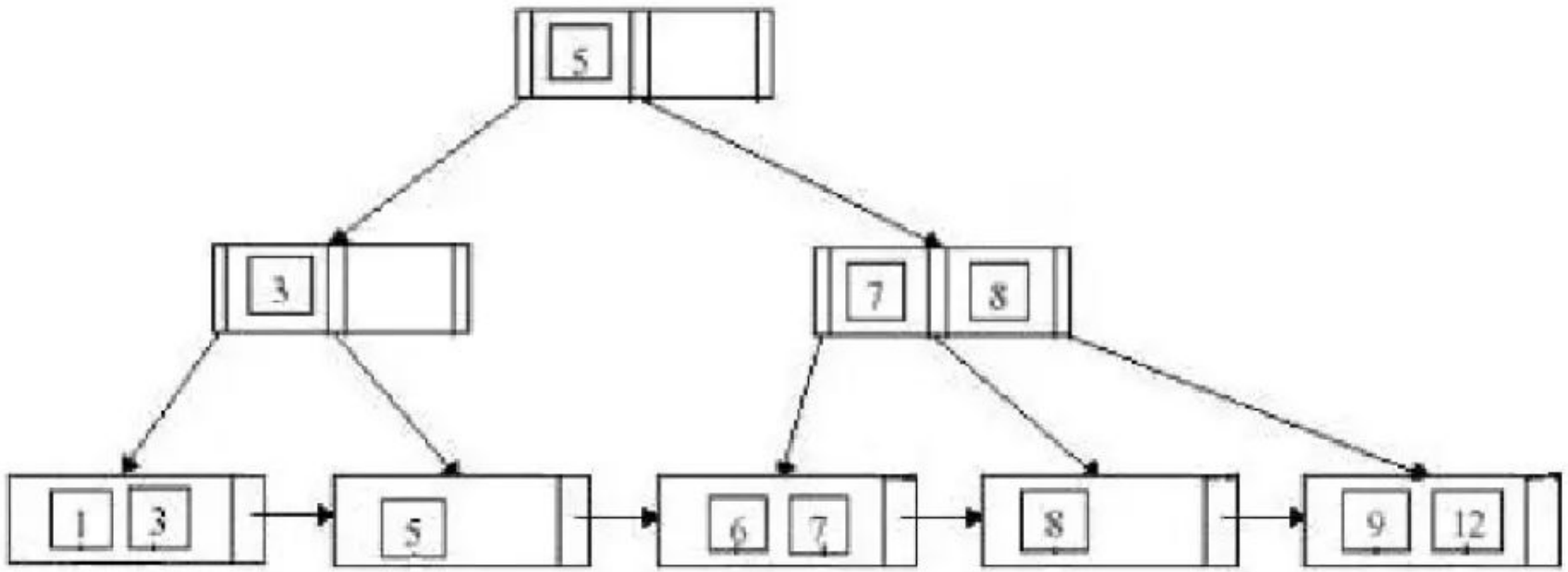
B+ Tree order 3 Insertion

- Inserting value 7.



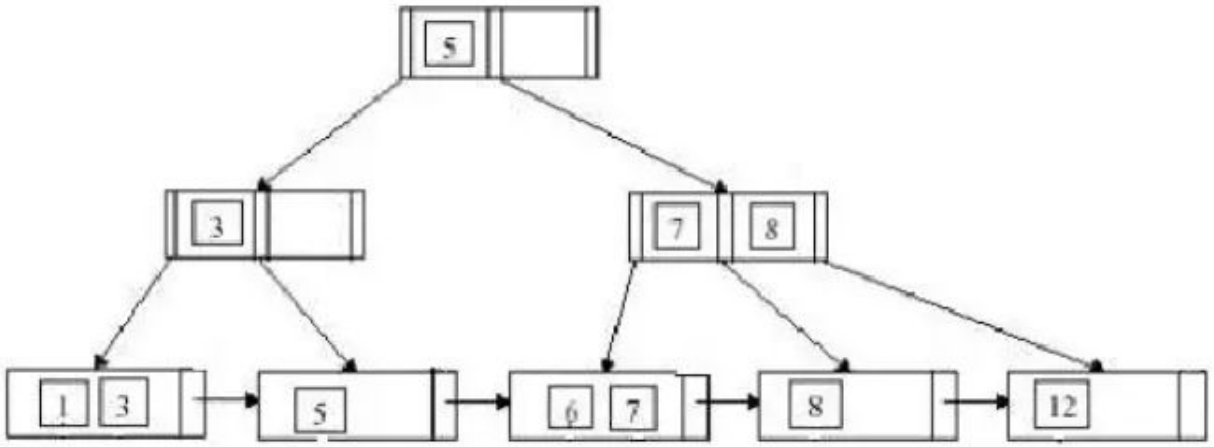
B+ Tree Deletion

- Initial Tree



B+ Tree Deletion

- Delete Value 9
- Since the node is not less than half full, the tree is correct.



B+ Tree Deletion

- Deleting value 8
- The node is less then half full, the values are redistributed from the node on the left because it is full.
- The parent node is adjusted to reflect the change.