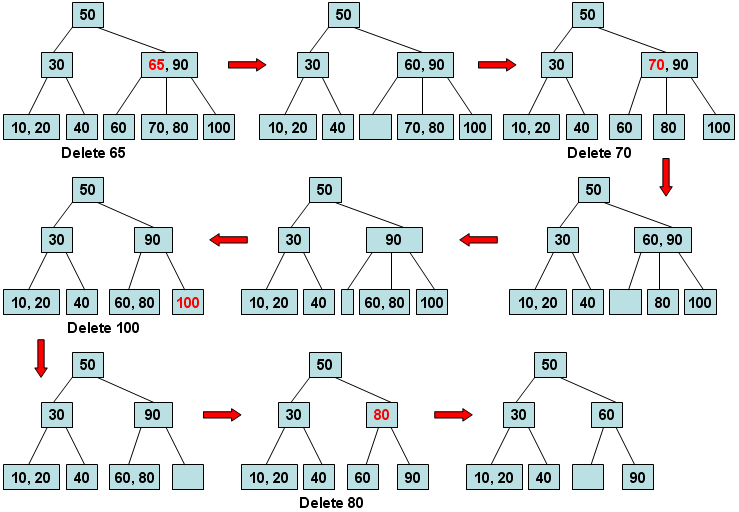
**B TREE**

In computer science, a **B-tree** is a self-balancing tree data structure that keeps data sorted and allows searches, sequential access, insertions, and deletions in logarithmic time. The B-tree is a generalization of a binary search tree in that a node can have more than two children. Unlike self-balancing binary search trees, the B-tree is optimized for systems that read and write large blocks of data. B-trees are a good example of a data structure for external memory. It is commonly used in databases and file systems.

A B-tree of order *m* is a tree which satisfies the following properties:

1. Every node has at most *m* children.
2. Every non-leaf node (except root) has at least ⌈*m*⁄2⌉ children.
3. The root has at least two children if it is not a leaf node.
4. A non-leaf node with *k* children contains *k*−1 keys.
5. All leaves appear in the same level



**B+ TREE**

A B+ tree is a data structure often used in the implementation of database indexes.  Each node of the tree contains an ordered list of keys and pointers to lower level nodes in the tree.  These pointers can be thought of as being between each of the keys.  To search for or insert an element into the tree, one loads up the root node, finds the adjacent keys that the searched-for value is between, and follows the corresponding pointer to the next node in the tree. Recursion eventually leads to the desired value or the conclusion that the value is not present.

