B-Trees and B+Trees

B-Tree

A B-tree of order m is a type of n-ary trees with some particularly nice properties:

- Every node has at most m children
- Every non-leaf node, except the root, has at least $m/2$ children
- The root node, if it is not a leaf node, has at least 2 children
- A non-leaf node with c children, contains $c - 1$ search key values, which act as separators or *discriminators*, to guide seaches down appropriate sub-trees
- All leaf nodes appear in the same level
- B-Trees are always height balanced
- Update and search is $O(\log n)$

B-Tree

In fact, different authors define the B-Tree in slightly different ways.

This does not really matter, because no-one really uses basic B-Trees:

- When used as an in-memory data structure, they have no significant advantage over AVL-trees or other height balanced binary trees like 2-3 trees or red-black trees, and are a bit more complex to implement.

Instead, a variant of the B-Tree called a B+Tree is the main-stay of databases and the most comment *external* data structure in use today.

We will not consider the basic B-Tree further but concentrate on the B+Tree.