B^+ Trees

A close relative of B trees.

B-Tree

- **Definition:** A **B-tree** of order d is an (a,b)-tree such that

  $a = \lceil d/2 \rceil$ and $b = d$

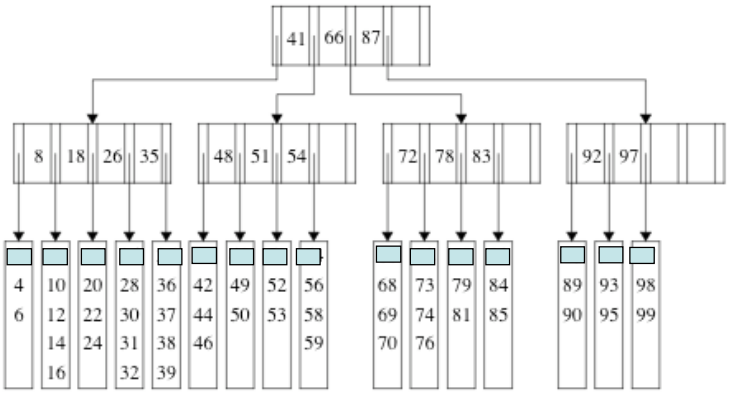
  Examples:
  - A (2,3)-tree is a B-tree of order 3,
  - A (2,4)-tree is a B-tree of order 4,
  - A (3,5)-tree is a B-tree of order 5, etc.
- **Definition of *d*:** Given a disk, choose d such that

d-1 key-item pairs together with d references to children

fit into a single disk block

- **Observations:**
  - Every node in a B-tree is at least half full
  - Underflow with fusion in a B-tree always creates a node that is full and thus occupies exactly one block

B Tree



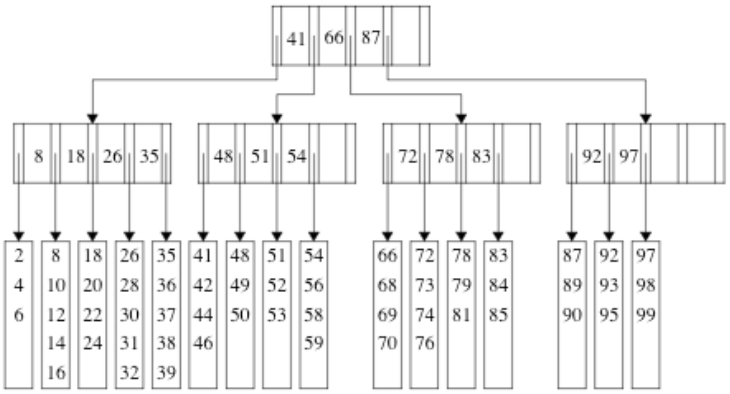
B+ Tree (from text)

A B-tree of order M is an M-ary tree with the following properties:

1. The data items are stored at leaves.
2. The nonleaf nodes store up to $M - 1$ keys to guide the searching; key i represents the smallest key in subtree $i + 1$.
3. The root is either a leaf or has between 2 and M children.
4. All nonleaf nodes (except the root) have between $\lceil M/2 \rceil$ and M children.
5. All leaves are at the same depth and have between $\lceil L/2 \rceil$ and L children, for some L.

Note: B+ trees are often defined with the leaves linked So that the entire Dictionary can be accessed easily in order.

B+ Tree



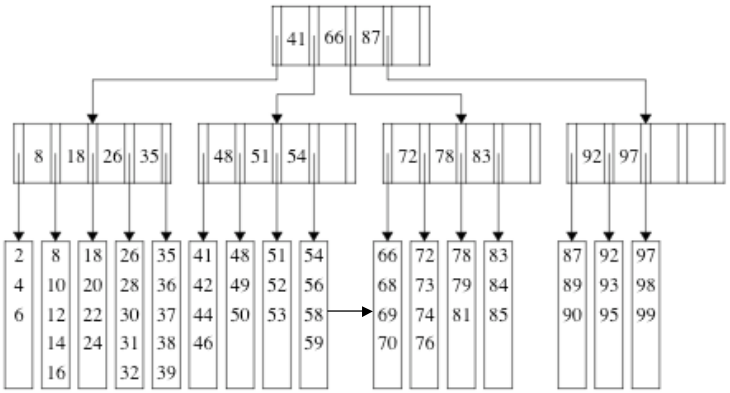
B+ Tree (from text)

A B-tree of order M is an M-ary tree with the following properties:

1. The data items are stored at leaves.
2. The nonleaf nodes store up to $M - 1$ keys to guide the searching; key i represents the smallest key in subtree $i + 1$.
3. The root is either a leaf or has between 2 and M children.
4. All nonleaf nodes (except the root) have between $\lceil M/2 \rceil$ and M children.
5. All leaves are at the same depth and have between $\lceil L/2 \rceil$ and L children, for some L.

Note: B+ trees are often defined with the leaves linked So that the entire Dictionary can be accessed easily in order.

B^+ Tree



Note: Although all leaves are linked only one arrow is shown.