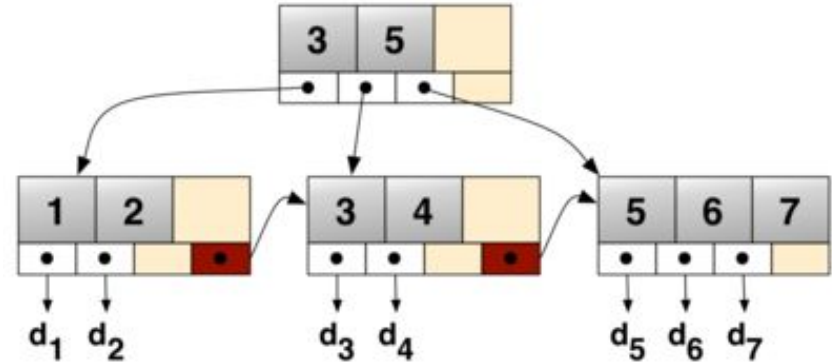


B+ Tree

Structure

- Inner - indexing
- Leaf - data
- Root
- Keys & pointers
- Order (m) - max num of direct child nodes
- Leafs are linked



Properties:

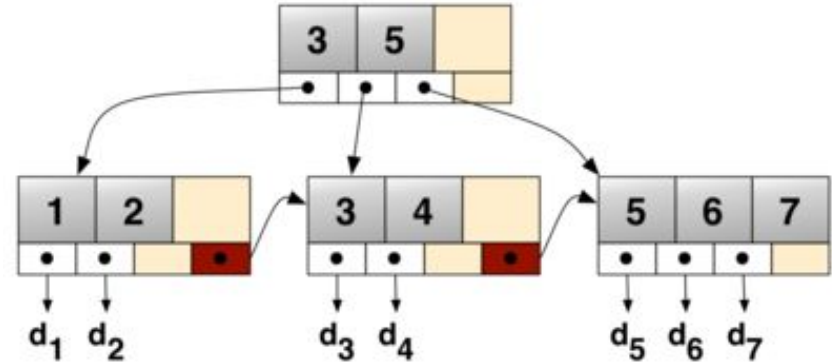
- K - max number of keys ($m - 1$)
- Min keys num:
 - Leaf $\Rightarrow \text{ceil}(K/2)$
 - Inner $\Rightarrow \text{floor}(K/2)$
- Keys can be repeated in inner nodes

Characteristics

- Balanced
- Ordered
- Sequential Access (range queries)
- High fan-out (reduced height)
- Disk-oriented
- Same search complexity
- Supports bulk-loading

Applications:

- DB indexing
- File system
- OS
- NoSQL DBs



Search

$K < \text{keys}[0] \Rightarrow \text{go child}[0]$

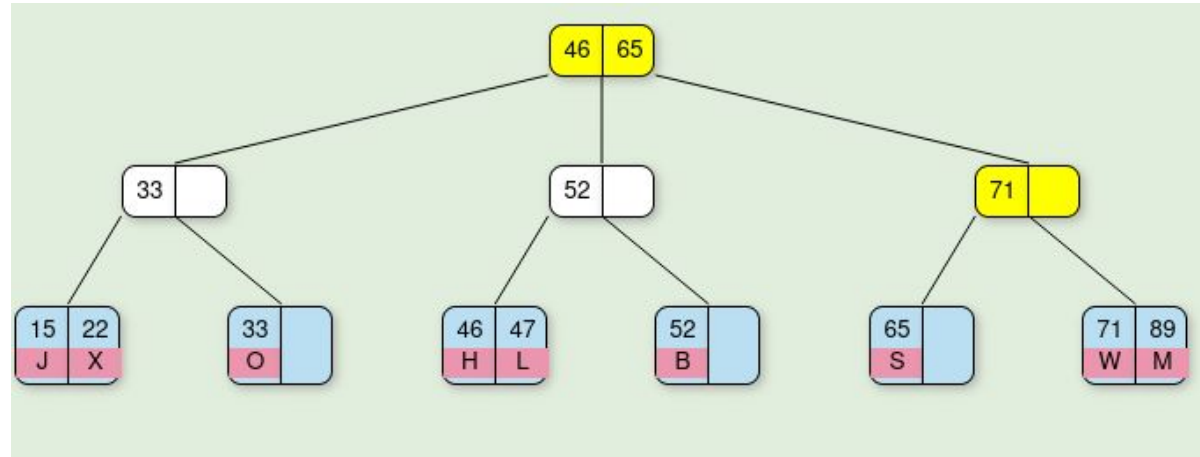
$\text{keys}[i - 1] \leq K < \text{keys}[i] \Rightarrow \text{go child}[i]$

$\text{Keys}[n - 1] \leq K \Rightarrow \text{go child}[n]$

Example: search 65

Notes:

- $O(\log_2(m-1) * \log_m n)$



Insertion

- Find leaf
- Insert at correct position
- No overflow ($n < \text{order}$) \Rightarrow END
- Else ($n == \text{order}$) \Rightarrow split
 - Old node $[0, m)$
 - New node $[m, n)$
 - Promote middle key to parent (for inner node - key is kicked out)
 - Repeat until a parent found that need no split
- Root split - treat as an empty parent

Notes:

- Tree grows up - to root
- Change parent during children split

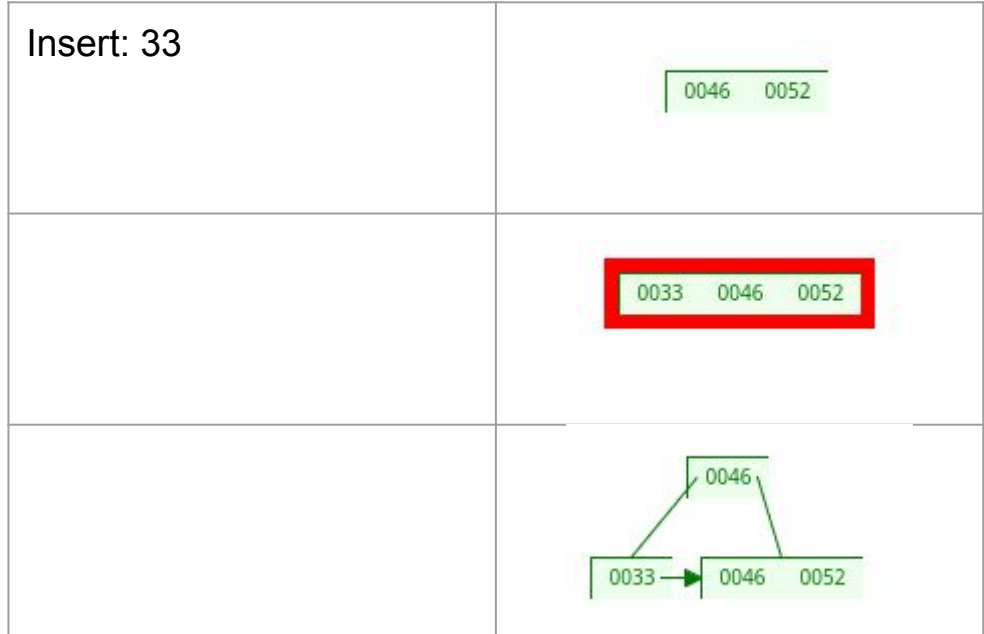
Tree order: 3	
Insert 52	
Insert 46	

Insertion (leaf split)

- Find leaf
- Insert at correct position
- No overflow ($n < \text{order}$) \Rightarrow END
- Else ($n == \text{order}$) \Rightarrow split
 - Old node $[0, m)$
 - New node $[m, n)$
 - Promote middle key to parent (for inner node - key is kicked out)
 - Repeat until a parent found that need no split
- Root split - treat as an empty parent

Notes:

- Tree grows up - to root
- Change parent during children split



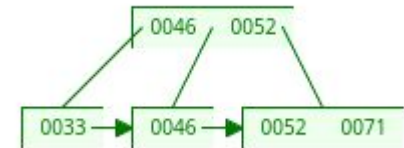
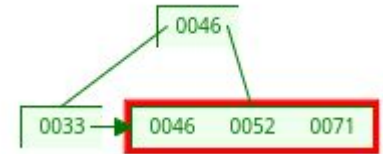
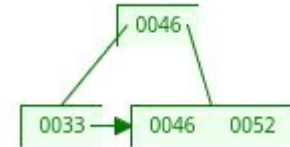
Insertion (leaf split)

- Find leaf
- Insert at correct position
- No overflow ($n < \text{order}$) \Rightarrow END
- Else ($n == \text{order}$) \Rightarrow split
 - Old node $[0, m)$
 - New node $[m, n)$
 - Promote middle key to parent (for inner node - key is kicked out)
 - Repeat until a parent found that need no split
- Root split - treat as an empty parent

Notes:

- Tree grows up - to root
- Change parent during children split

Insert: 71



Insert (inner split)

Insert: 65

