

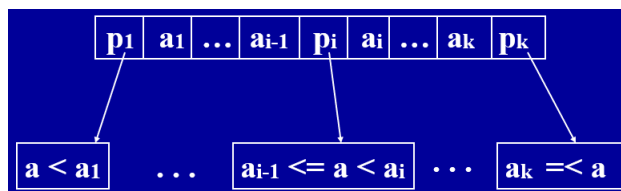
数据库系统 期末速通教程

8. 索引

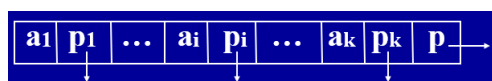
8.1 B^+ 树

[定义 8.1.1] B^+ 树是一棵平衡搜索树(根节点到叶子节点的距离都相等), 每个节点存一页.

(1) 树中节点形如:



(2) 叶子节点形如:



(3) 所有叶子节点按 a_i 值非降序排列连成单链表.

[注]

(1) 固定 B^+ 树的 K 值和 L 值.

① 除根节点外, 每个非叶子节点有 $\left[\left\lceil \frac{K}{2} \right\rceil, K \right]$ 个数据, 根节点可有少于 $\left\lceil \frac{K}{2} \right\rceil$ 个数据.

② 每个叶子节点有 $\left[\left\lceil \frac{L}{2} \right\rceil, L \right]$ 个数据.

(2) 若有 N 个数据, 则查找的时间复杂度为 $O\left(\log_{\frac{K}{2}} N\right)$.

(3) B^+ 树与哈希的对比:

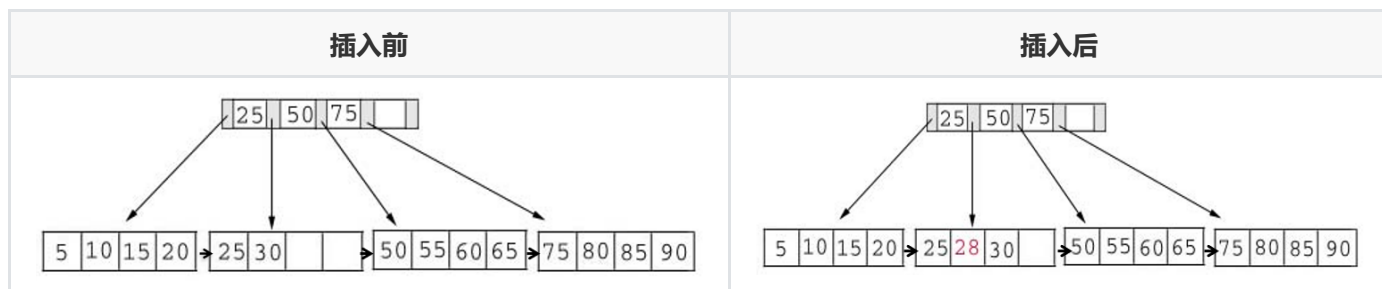
① 等值查找时, 效率相当.

② 哈希查找只能做等值查找, B^+ 树可做等值查找和范围查找.

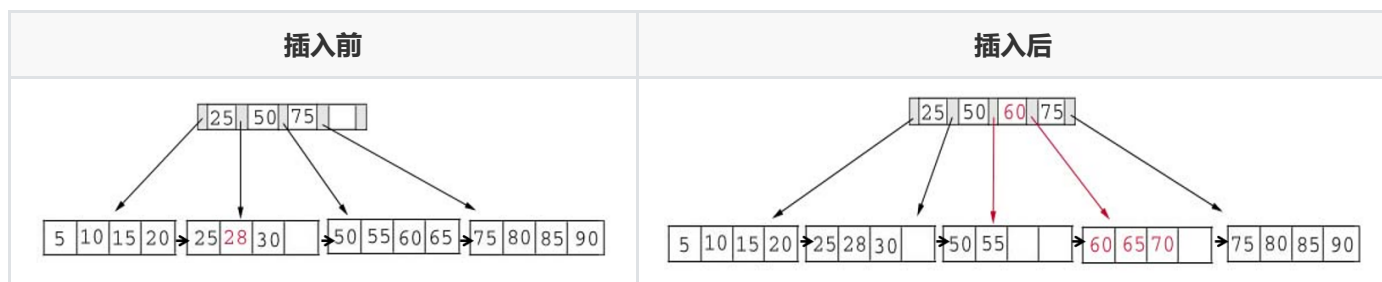
(4) B^+ 树的一般高度为 2 或 3.

[例 8.1.1] 考察如下图所示的 $K = L = 4$ 的 B^+ 树.

(1) 插入数据 28 : 有空间直接插入.

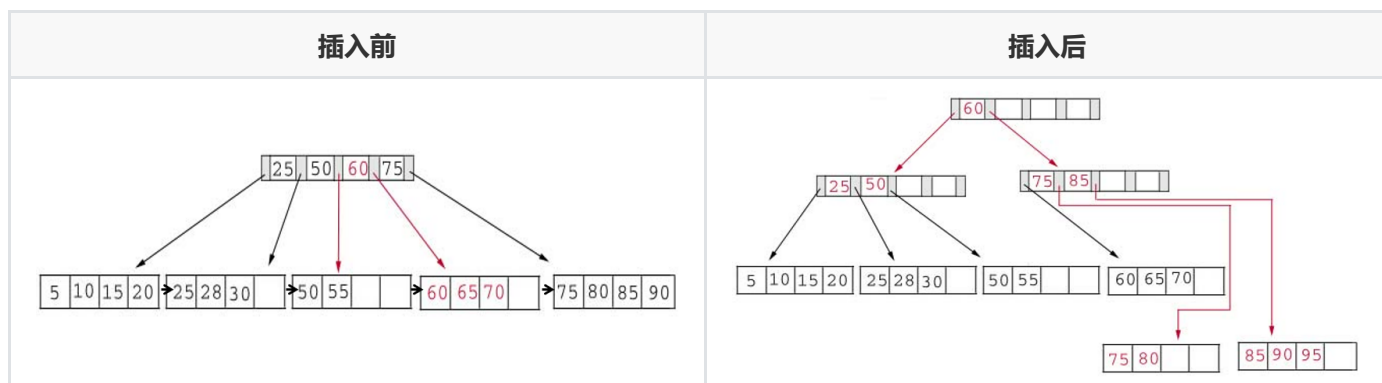


(2) 插入数据 70 : 70 应插入左图中的第 3 个叶子节点. 取 50, 55, 60, 65, 70 的中位数 60 作索引.

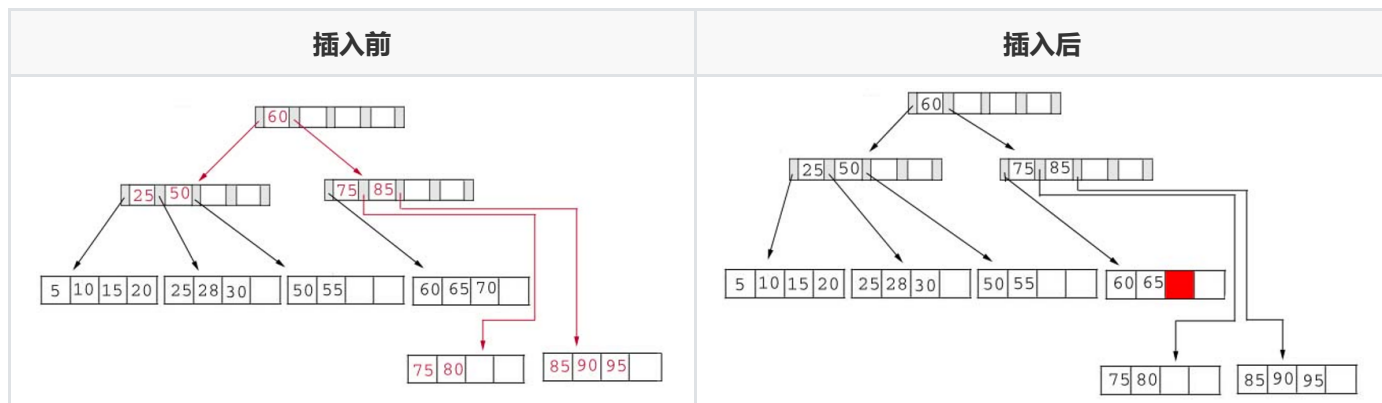


(3) 插入数据 95 : 95 应插入左图中的第 4 个叶子节点. 取 75, 80, 85, 90, 95 的中位数 85 作索引.

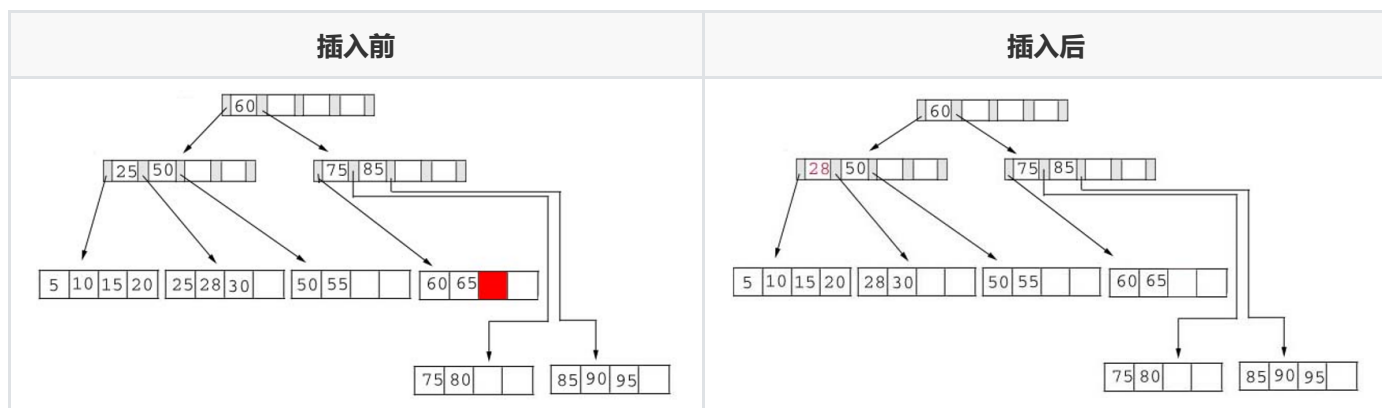
根节点的索引已满, 取 25, 50, 60, 75, 85 的中位数 60 作索引, 根节点分裂.



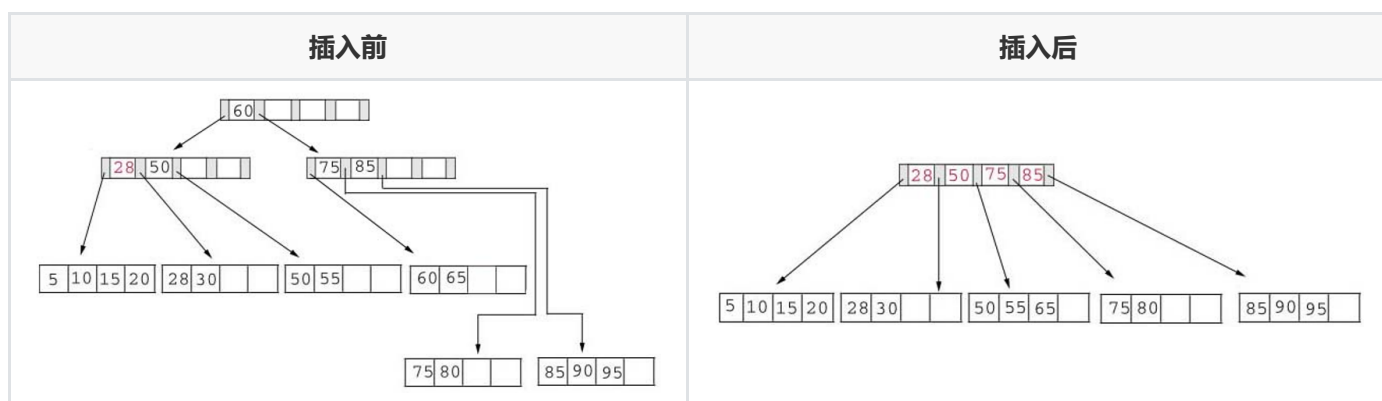
(4) 删除数据 70 : 删除后满足每个非根节点至少半满, 且 70 非索引, 则直接删.



(5) 删除数据 25 : 删除后满足每个非根节点至少半满, 但 25 是索引, 取 25 所在的叶子节点的第 2 个元素 28 作索引.



(6) 删除数据 60 : 删除后不满足每个非根节点至少半满, 需合并节点, 且 60 是索引, 需合并索引.



[例 8.1.2] 设员工表有 $1e5$ 个 100 B 的数据. B^+ 树的每个数据 10 B, 每个指针 4 B, 每个页大小 2 KB.

(1) 每个数据项 (a, p) 14 B, 每页存放 $\left\lfloor \frac{2 \times 1024 \text{ B}}{14 \text{ B}} \right\rfloor = 146$ 个数据项.

(2) B^+ 树的第一层需 $\left\lceil \frac{1e5}{146} \right\rceil = 685$ 页, 第二层需 $\left\lceil \frac{685}{146} \right\rceil = 5$ 页, 第三层需 $\left\lceil \frac{5}{146} \right\rceil = 1$ 页.

(3) 取员工表的一行访存 4 次, 即访问 B^+ 树的三层得到指向该行数据的地址, 再根据该地址取该数据.

8.2 索引

[定义 8.2.1]

(1) [主索引, 聚集索引, Primary Index, Clustered Index] 若关系中的元组按 A 值升序或降序存放, 则 A 上的 B^+ 树索引为主索引或聚集索引.

(2) [次索引, 非聚集索引, Secondary Index, Non-clustered Index] 若关系中的元素不按 A 值升序或降序存放, 则称 A 上的 B^+ 树索引为次索引或非聚集索引.

[注]

(1) 主索引与主键无必然关系, 即主键未必建主索引.

(2) 次索引常用于 CK 或重复值较少的属性.

(3) 一个关系的主索引唯一, 次索引不唯一.

(4) 主索引与次索引的速度比较:

① 单值查找时, 效率相同.

② 范围查找时, 主索引常比次索引效率高.