

第二次作业答案：

习题 5.2

5.2 考虑图 5.21 所示的阶数 $m=4$ 的 B+树索引。

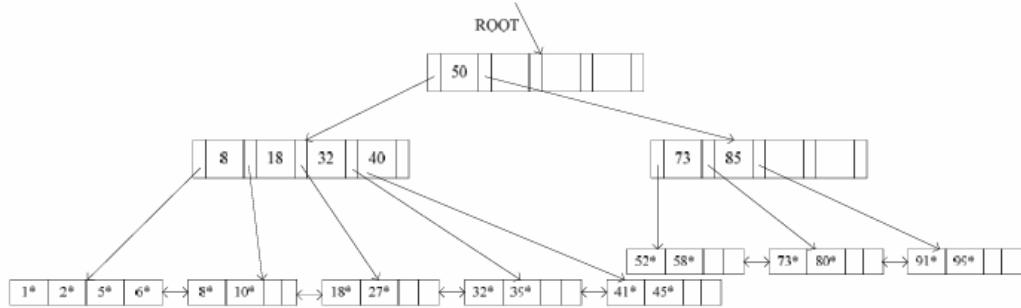


图 5.21 习题 5.2 附图

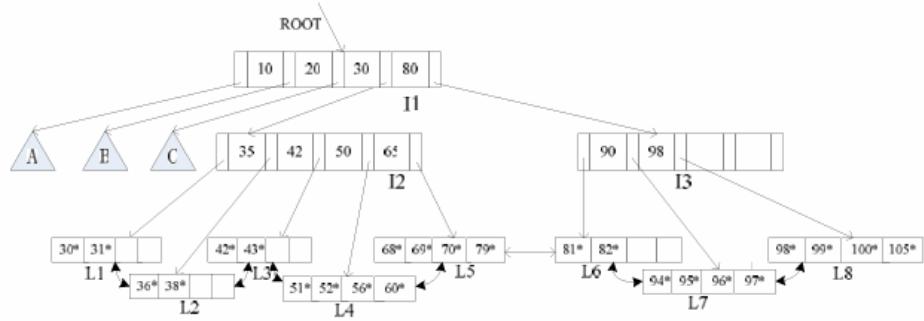
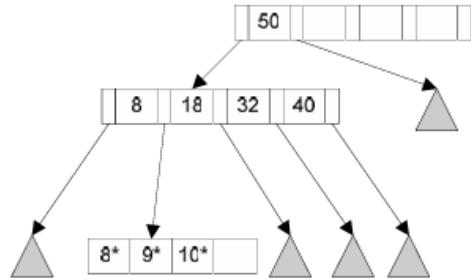


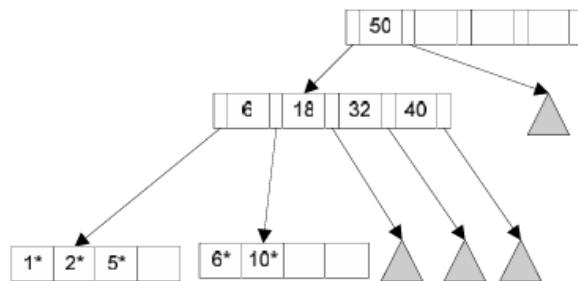
图 5.22 习题 5.3 附图

- (1) 标示插入数据项 9*之后的 B+树，并指出完成该插入需要读多少个页和写多少个页。
- (2) 给出在原树中删除数据项 8*之后的 B+树，并指出完成该操作需要读多少个页和写多少个页。
- (3) 给出在原树中先插入数据项 46*，然后再删除数据项 52*之后的 B+树。
- (4) 给出在原树中，依次删除 32*、39*、41*、45*和 73*之后的 B+树。

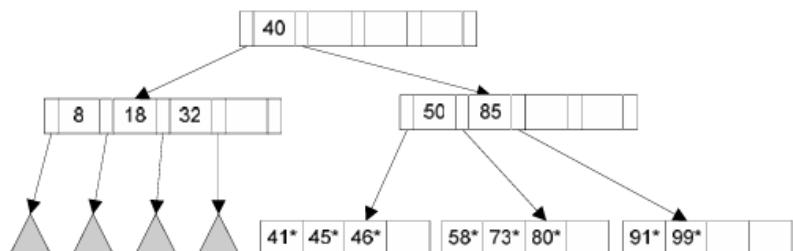
(1) 由图看出插入 9* 不需要分裂，直接插入即可。由于索引项即数据文件本身，从根结点到索引项读 3 个页。只有叶结点改过，所以写 1 个页。插入后的局部图如下：



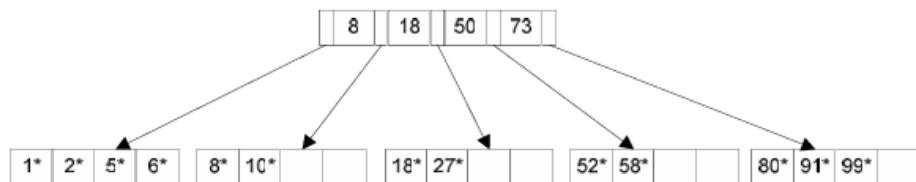
(2) 删除 8* 后要跟前一个索引项重组，从根结点到两个索引项要读 4 个页。操作完成后两个叶结点和一个中间结点是脏页，故要写出 3 个页。删除后的局部图：



(3) 可直接插入数据项 46*。删除数据项 52* 则要合并重组，操作完成后的局部图：



(4) 依次删除 32*、39*、41*、45*、73* 后的图：



习题 5.3

5.3 考虑图 5.22 所示 B+树索引：内节点可容纳 4 个键值和 5 个指针；叶节点中直接存储数据记录，可容纳 4 条记录且相邻叶节点之间用双链连接在一起。回答以下问题。

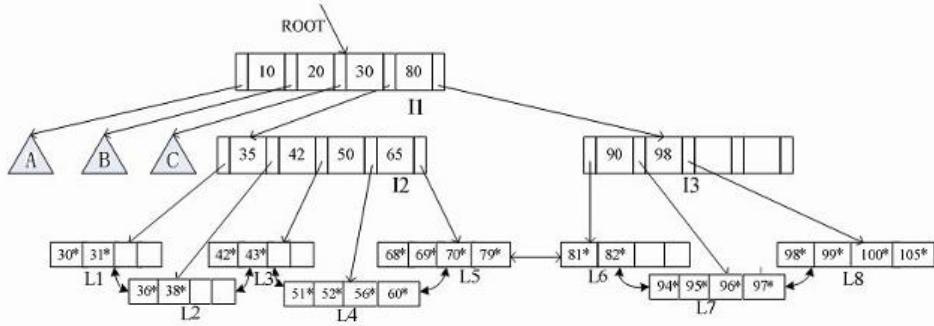
(1) 指出回答查询“取搜索键值大于 38 的所有记录”时，需要读取的有关节点。

(2) 给出插入 109*后的 B+树。

(3) 给出从原树中删除 81*后的 B+树。

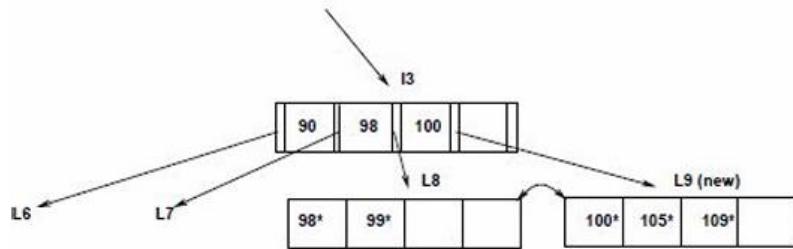
(4) 给出一个插入时会导致树高度增加的键值。

(5) 图中没有详细给出子树 A、B 和 C。你能推测出这些子树的内容和形状吗？

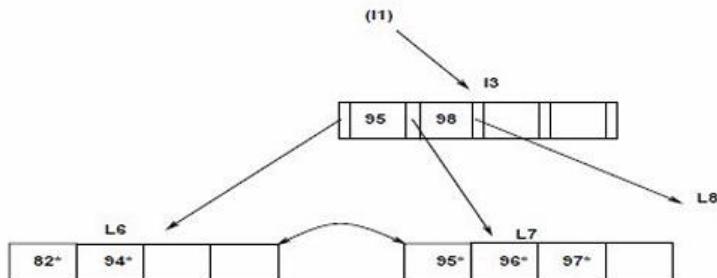


(1) 查询大于 38*的所有记录，要读取的节点有：I1,I2,L2,L3,L4,L5,L6,L7,L8

(2) 插入 109*后，原 L8 节点需要分裂，完成操作后的局部图：



(3) 删除 81*后，L6,L7 两个节点要重组，操作完成后的局部图如下：



(4) 插入任何[65,79]之间的搜索键值，都会分裂 L5 节点，而 I2 也是满的，向上分裂到根结点，根结点也是满的，就会导致高度增加一层。

(5) 关于子树 A、B、C，我们可推出以下几件事：1) 它们都是树高为 1 的子树，因为它们的相邻子树，即以 I₂、I₃ 为根节点的子树树高都是 1；2) 子树 A 包含的键值树肯定少于 10 个，子树 B 所包含的键值在范围 [10,20) 之间，子树 C 所包含的键值在范围 [20,30)

之间；3) 每个中间节点至少会含有 3 个键值和 3 个指针。