

### 1、Page Eviction Policy

运用最近最少使用（LRU）算法来进行页面驱逐，优先驱逐最近最少使用的页面。使用名为 LRUList 的链表来存储最近使用的页面，并用名为 pageId2Loc 的哈希表来映射页面 ID 到其在内存中的位置。当访问一个页面时，我将其添加到 LRUList 的尾部。

### 2、B+树 Search

从根节点开始递归搜索，直到到达叶节点并返回该节点。

### 3、B+树中 Insert

首先，递归搜索插入新节点的叶节点。如果叶节点没有满，就直接插入。否则，分裂叶节点，将节点上存储的右半部分页面移动到新的叶节点，并在父节点中添加一个新的条目。

### 4、B+树 delete

如果一个叶节点的填充度少于一半，则它从其一个兄弟节点中获取元组或与一个兄弟节点合并。相应更新父节点的条目。