

Describe any design decisions you made.

2.1

使用 map, 驱逐策略选择先进先出策略, 驱逐最先被占用的 page

2.2 查找

* 根据查找 PID 的类型

- * BtreePageId.LEAF : 找到, 返回结果. 递归结束.

- * BtreePageId.INTERNAL :

 - * 如果 Field 为 null 左递归

 - * 否则从左向右依次寻找 下一次递归位置

- * other type : 非法情况

2.3 插入 实现 split 函数

* splitLeafPage

- * 将叶子结点尽可能均匀分成两个部分, 新节点拥有左边的 Tuple

- * 维护叶子结点的链表结构

- * 为父亲结点添加 npage 索引

- * updateParentPointers

* splitInternalPage

- * 将叶子结点尽可能均匀分成两个部分, 新节点拥有左边的 Tuple

- * 将 page(后半部分)的第一个结点取出添加到其父亲上

- * 为父亲结点添加 npage 索引

- * updateParentPointers

2.4 删除

* stealFromLeafPage

- * 计算个数

- * 确定 steal 的方向

- * 更新父亲结点

* stealFromLeftInternalPage

- * 同上

- * reverseIterator()
- * stealFromRightInternalPage
 - * 同上
- * reverseIterator()
- * mergeLeafPages
 - * 向左边叶子结点添加右边叶子结点的 tuple
 - * 删除父亲指向右边结点的 Entry
- * mergeInternalPages
 - * 同上

Discuss and justify any changes you made to the API.

添加一些必要的 private 成员

Describe any missing or incomplete elements of your code.

缺少一些非法性检测,需要保证,使用者完全按照接口使用.

Describe how long you spent on the lab, and whether there was anything you found particularly difficult or confusing.

花费 大概 8 个小时
出现问题调试过程真的好累