



Assignment4

课程: 数据库原理与应用

姓名: 雷翔

学号: 2053932

时间: 2023 年 6 月

Q1. 13.14 标准的缓冲区管理器假定每个块的大小和读取代价都是相同的。设想一个缓冲区管理器使用对象引用率而不是 LRU，对象引用率是指一个对象在此前的 n 秒内被访问的频率。假设我们要在缓冲区中存储变长和读取代价可变 (例如网页，它的读取代价取决于它从哪个站点被获取) 的对象。试建议缓冲区管理器可以如何选择要从缓冲区中移出哪个块。

Ans.

维护一个优先队列，优先级取决于以下三个因素：

对象引用率 X、对象大小 Y、对象读取代价 Z。

记 $T = k_1 \times X + k_2 \times Y + k_3 \times Z$ ，其中 k_1, k_2, k_3 为常数。

每次缓冲区满时，将优先级最低的对象，即 T 最小的对象移出缓冲区。

Q2. 14.3 用下面的码值集合建立一棵 B^+ 树:

(2, 3, 5, 7, 11, 17, 19, 23, 29, 31)

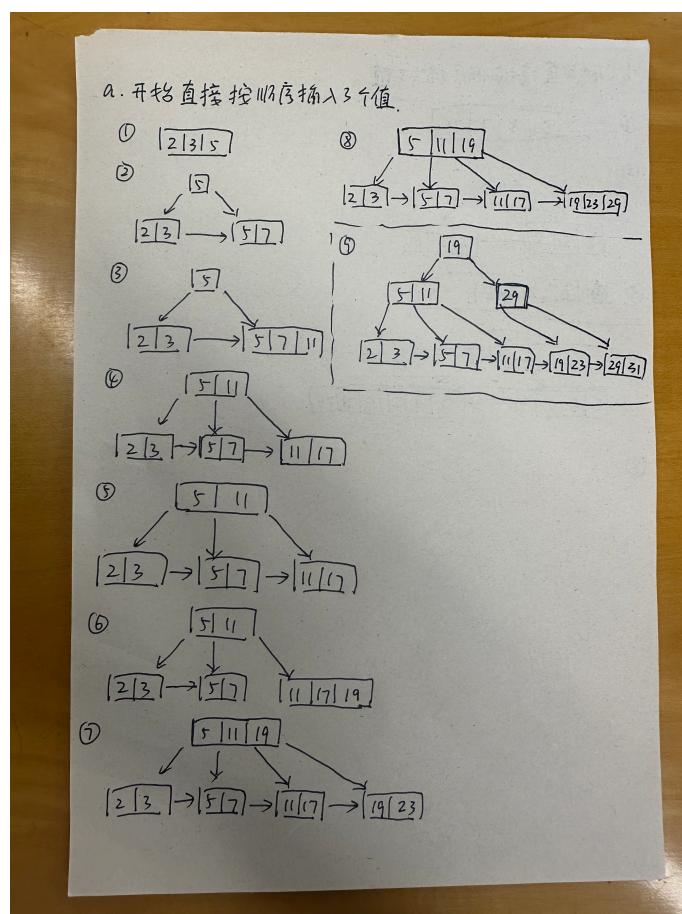
假设树初始为空，按升序添加这些值。当一个节点所能容纳的指针数是下列情况时，请分别构造 B^+ 树:

- a. 4 b. 6 c. 8

Ans.

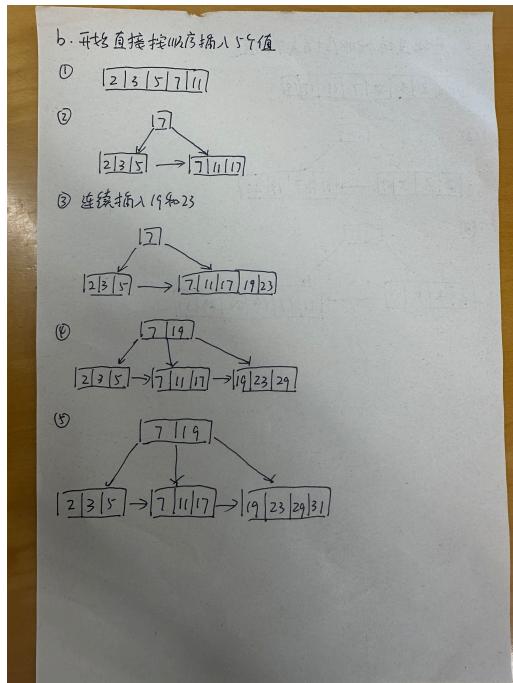
a.

$n = 4$, 每个叶节点最多有 $n - 1 = 3$ 个值, 最少有 $\lceil (n - 1)/2 \rceil - 1 = 2$ 个值。



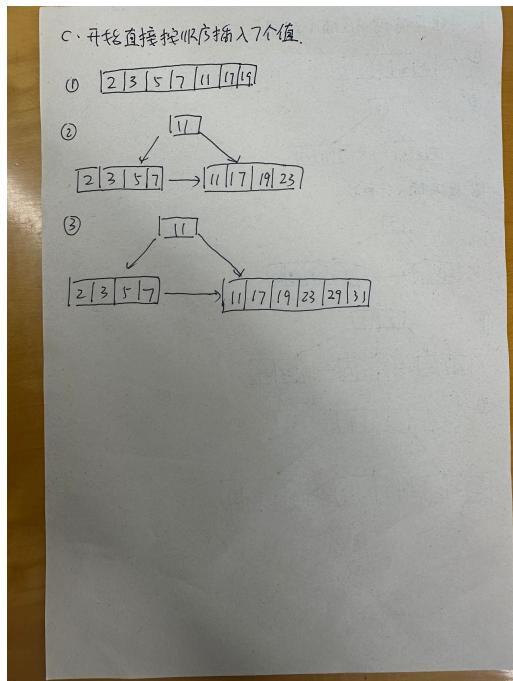
b.

$n = 6$, 每个叶节点最多有 $n - 1 = 5$ 个值, 最少有 $\lceil (n - 1)/2 \rceil - 1 = 3$ 个值。



c.

$n = 8$, 每个叶节点最多有 $n - 1 = 7$ 个值, 最少有 $\lceil (n - 1)/2 \rceil - 1 = 4$ 个值。



Q3. 14.13 请考虑如图 14-1 所示的 *instructor* 关系。

- 在 salary 属性上构建一个位图索引，把 salary 的值分为 4 个区间：小于 50000 的，50000 到 60000 以下的，60000 到 70000 以下的，70000 及以上的。
- 请考虑一个查询：查找在金融系中工资为 80000 或更高的所有教师。概述回答该查询的步骤，并给出为回答这个查询而构建的最终位图和中间位图。

Ans.

10101	Srinivasan	Comp. Sci.	65000	
12121	Wu	Finance	90000	
15151	Mozart	Music	40000	
22222	Einstein	Physics	95000	
32343	El Said	History	60000	
33456	Gold	Physics	87000	
45565	Katz	Comp. Sci.	75000	
58583	Califieri	History	62000	
76543	Singh	Finance	80000	
76766	Crick	Biology	72000	
83821	Brandt	Comp. Sci.	92000	
98345	Kim	Elec. Eng.	80000	

图 11-1 *instructor* 记录的顺序文件

a.

记 L_1 为小于 50000 的, L_2 为 50000 到 60000 以下的, L_3 为 60000 到 70000 以下的, L_4 为 70000 及以上的。

L_1	0 0 1 0 0 0 0 0 0 0 0 0
L_2	0 0 0 0 0 0 0 0 0 0 0 0
L_3	1 0 0 0 1 0 0 1 0 0 0 0
L_4	0 1 0 1 0 1 1 0 1 1 1 1

表 1: salary 的位图索引

b.

记 A_1 为金融系, A_2 为非金融系。

A_1	0 1 0 0 0 0 0 0 1 0 0 0
A_2	1 0 1 1 1 1 1 0 1 1 1

表 2: dept_name 的位图索引

记 B_1 为工资低于 80000 的, B_2 为工资高于 80000 的。

B_1	1 0 1 0 1 0 1 1 0 1 0 0
B_2	0 1 0 1 0 1 0 0 1 0 1 1

表 3: salary 的位图索引

取 dept_name 为 A_1 的位图和 salary 为 B_2 位图, 并执行这两个位图的交, 得到最终位图。

$A_1 \cap B_2$	0 1 0 0 0 0 0 0 1 0 0 0
----------------	-------------------------

表 4: 最终位图

所以, 金融系中工资为 80000 或更高的所有教师有 2 个, 分别是 Wu 和 Singh。

Q4. 17.15 请考虑以下两个事务:

T_{13} :

```
read(A);
read(B);
if A = 0 then B := B + 1;
write(B);
```

T_{14} :

```
read(B);
read(A);
if B = 0 then A := A + 1;
write(A);
```

令一致性需求为 $A = 0 \vee B = 0$, 初值是 $A=B=0$ 。

- 请说明包括这两个事务的每一个串行执行都保持了数据库的一致性。
- 请给出 T_{13} 和 T_{14} 的一个并行执行, 它产生了不可串行化的调度。
- 存在产生可串行化调度的 T_{13} 和 T_{14} 的并行执行吗?

Ans. a.

一致性需求是 A 、 B 至少有一个为 0, 初始条件为 $A=B=0$

假设两个事务以先 T_{13} 后 T_{14} 的次序执行, 执行顺序如下表所示:

T_{13}	T_{14}
read(A)	
read(B)	
if A = 0 then B := B + 1	
write(B)	
	read(B)
	read(A)
	if B = 0 then A := A + 1
	write(A)

执行完成后, $A=0$ 、 $B=1$, 保持数据库的一致性。

同理，假设两个事务以先 T_{14} 后 T_{13} 的次序执行，执行顺序如下表所示：

T_{13}	T_{14}
	read(B)
	read(A)
	if B = 0 then A := A + 1
	write(A)
read(A)	
read(B)	
if A = 0 then B := B + 1	
	write(B)

执行完成后，A=1、B=0，保持数据库的一致性。

b.

T_{13}	T_{14}
read(A)	
read(B)	
	read(B)
	read(A)
if A = 0 then B := B + 1	
	if B = 0 then A := A + 1
write(B)	
	write(A)

执行完成后，A=1、B=1，没有保持数据库的一致性，产生了不可串行化的调度。

c.

一旦 T_{13} 和 T_{14} 并行执行，那么必然导致先进行所有的读取操作，才会进行写操作，因为要并行，所以大体存在两种情况：

情况一：先进行 T_{13} ，在 T_{13} 进行写操作前， T_{14} 进行读。

情况二：先进行 T_{14} ，在 T_{14} 进行写操作前， T_{13} 进行读。

无论哪种情况，因为写操作还没有完成，读到的都是 0，所以两个 if then 语句均会执行，最后的结果就是 A=1、B=1，不满足一致性

所以不存在产生可串行化调度的 T_{13} 和 T_{14} 的并行执行。