

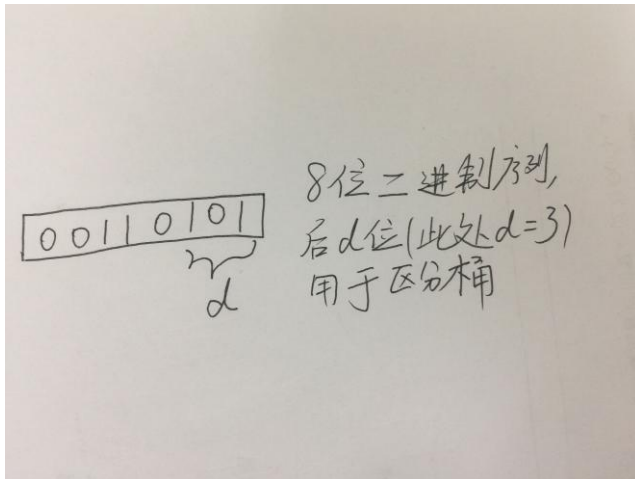
作业五：物理存储与查询优化（截止时间：4月25日晚12点）

请在邮件主题中加入以下字符串之一：第五次作业，第5次作业，作业五，作业5，homework5，否则可能收不到自动回复。

1. 考虑一个可扩展的 Hash 结构，其每个桶最多可以容纳两个记录。

考虑按顺序插入 8,16,4,3,11,12 后的索引结构，且初始索引为空。

且用低位表示桶的数目，即：全局深度为 d 时考虑的是 Hash 函数的最后 d 位。如：

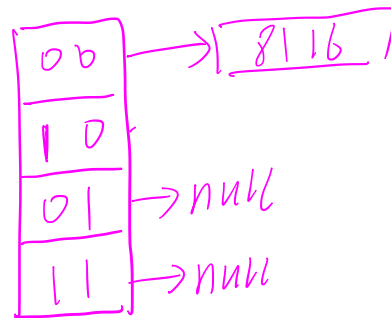
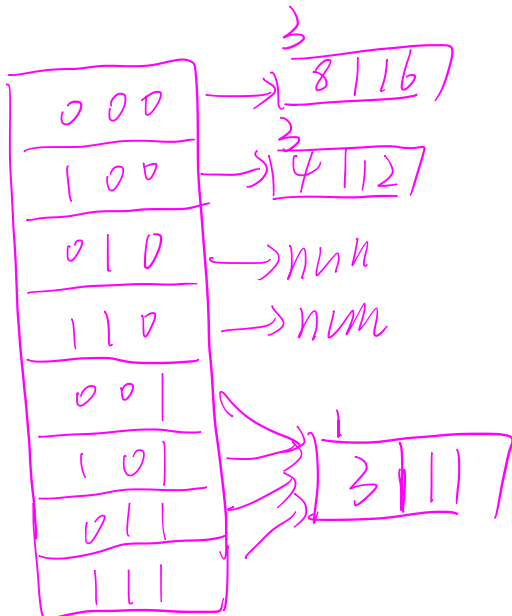


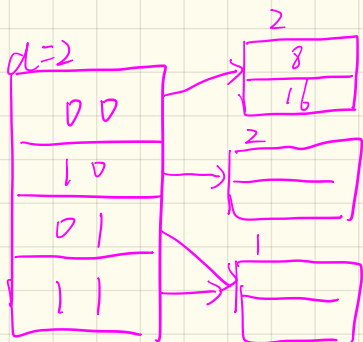
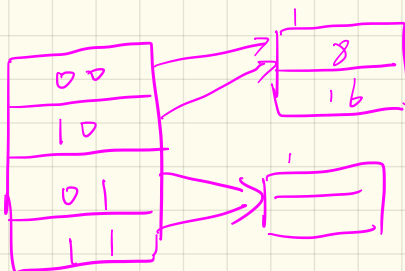
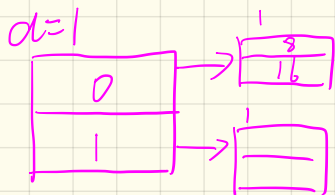
1000
10000
0100
011
1100

(1) 此时索引全局深度是多少？画出此时索引结构。

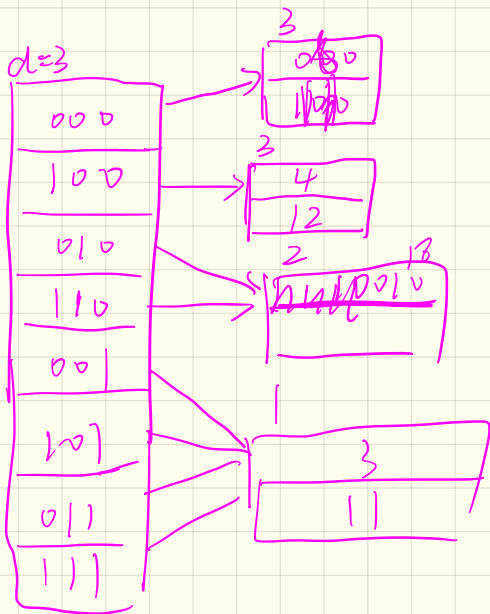
(2) 在上题索引基础上插入 18。包括 18 的桶的深度是多少？画出此时的索引结构。

(3) 考虑如下图所示可扩展 Hash 结构：

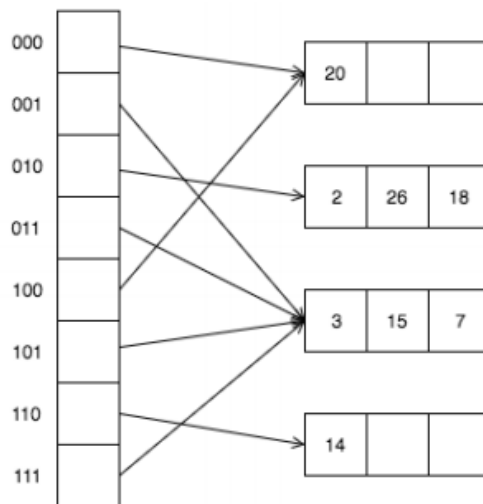




$$\begin{array}{r} 12 \\ 1100 \\ \hline \end{array}$$



$$\begin{array}{r} 18 \\ 10010 \\ \hline \end{array}$$

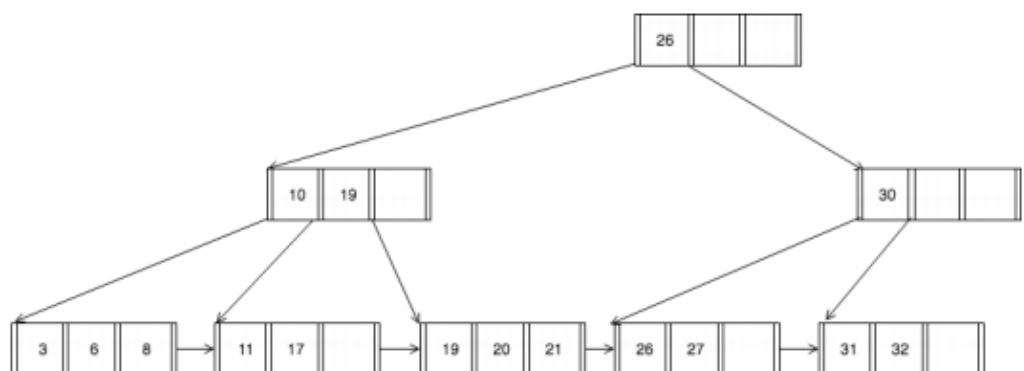


向其中按顺序插入 28,30,4,8,34。哪一个值的插入引发第一次桶的分裂？哪一个值的插入导致全局深度第一次增加？

2.

(1) 把 1-10 插入到阶数为 4 和为 8 的空 B+ 树中，并分别画出插入的最终结果。假设右子树中的值大于等于索引值，而左子树小于索引值。

(2) 考虑如下 B+ 树：



假设：右子树中的值大于等于索引值，而左子树小于索引值。

当出现下溢，且可以从左右两个兄弟节点借元素，则优先选择右边的兄弟节点。

下面四题全部是以上图为基础，它们之间没有联系。

(a) 增加元素 10; (b) 增加元素 10、18; (c) 删除元素 11;

(d) 删除元素 31。

画出四种情况后的 B+树。

3. 考虑关系 $R1(A,B,C)$, $R2(C,D,E)$, 和 $R3(E,F)$.

(1) 假设 $R1$ 、 $R2$ 、 $R3$ 的主键分别是 A 、 C 、 E 。 $R1$ 有 1000 个元组， $R2$ 有 1500 个元组， $R3$ 有 750 个元组，估计 $R1 \bowtie R2 \bowtie R3$ 的大小，并给出一个这个连接操作的实现策略（简述）。

(2) 假定没有主键， $V(C,R1)=900$, $V(C,R2)=1100$, $V(E,R2)=50$, $V(E,R3)=100$ 。假设 $R1$ 有 1000 个元组， $R2$ 有 1500 个元组， $R3$ 有 750 个元组，估计 $R1 \bowtie R2 \bowtie R3$ 的大小，并给出一个这个连接操作的实现策略（简述）。

4. 图书馆数据库有如下3个关系模式：

$B(B\#, Title, Author, Publisher)$

$B\#$

$S(S\#, Name, Department)$

$(S\# - B\#)$

$L(S\#, B\#, Date)$

其中 B 为图书信息表： $B\#$ 为图书编号， $Title$ 为书名， $Author$ 为作者， $Publisher$ 为出版社； S 为学生信息表： $S\#$ 为学号， $Name$ 为学生姓名， $Department$ 为学院名； L 为借阅信息表： $S\#$ 为借阅人学号， $B\#$ 为被借阅图书编号， $Date$ 为借阅日期。

用户有一查询语句：

Select Name

From B, S, L

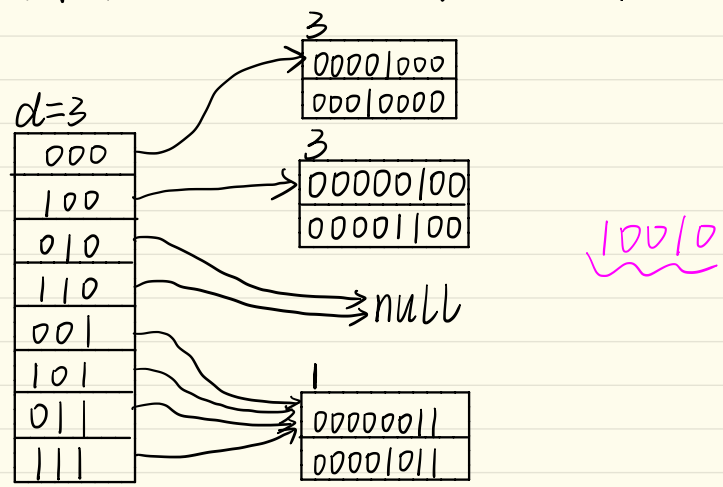
Where L.S#=S.S# and L. B#=B.B# and Title="Jane Eyre"

检索借阅了书名为“Jane Eyre”的学生姓名。

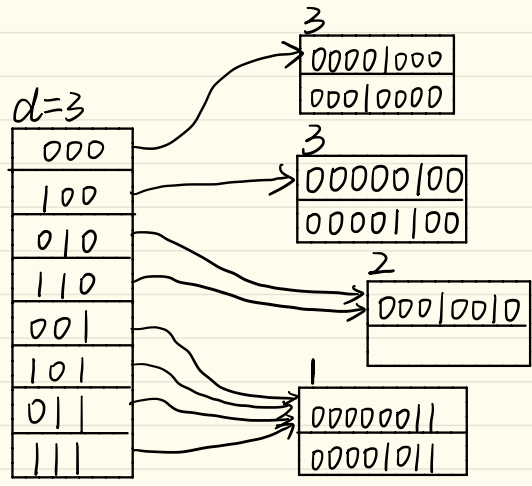
求解下列问题：

- a. 写出以上SQL 语句所对应的关系代数表达式，并画出对应的逻辑查询计划树。使用启发式查询优化算法，对该逻辑查询计划树进行优化，并画出优化后的逻辑查询计划树。
- b. 设L表有10000条元组，B表有2000条元组，S表中有1000条元组，L表中满足借阅书名为“Jane Eyre”的元组数为50，计算优化前与优化后的查询计划中每一步所产生的中间结果大小。（要求写出具体计算过程）

1. (1) 此时索引全局深度为3, 索引结构如下:

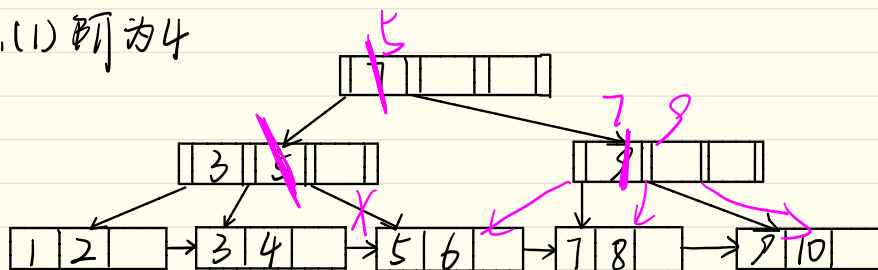


(2) 插入18, 其桶深度为2, 此时索引结构:

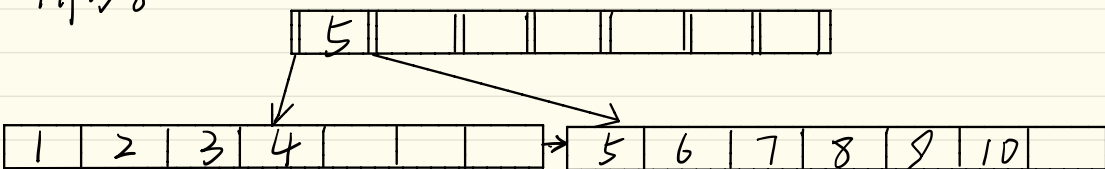


(3) 插入8时, 引发第一次桶的分裂, 插入34时, 导致全局深度第一次增加.

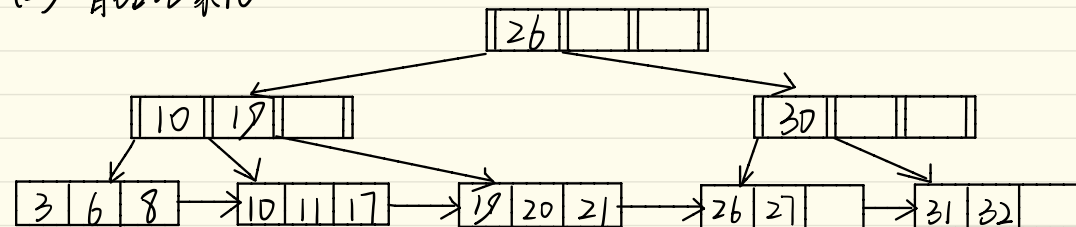
2.11) 树为4



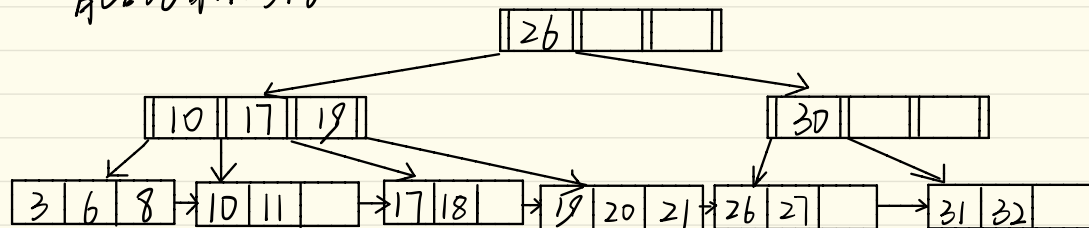
树为8



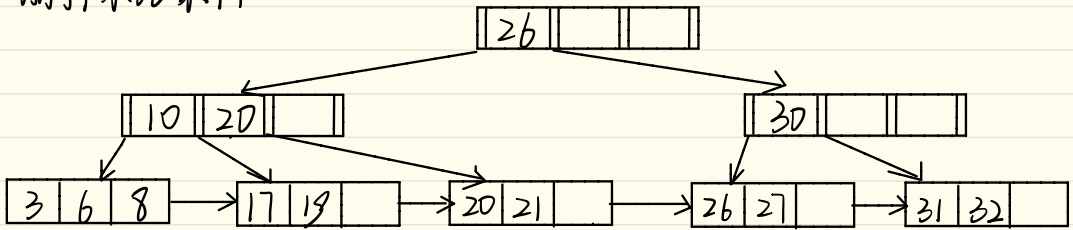
(2) 增加元素10



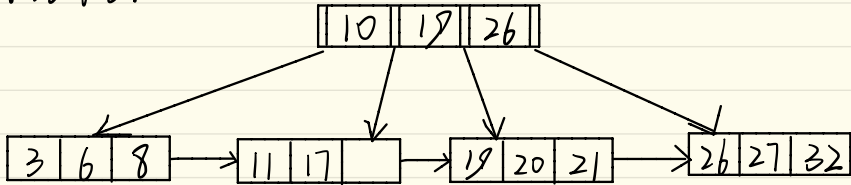
增加元素10, 18



删除元素11



删除元素31



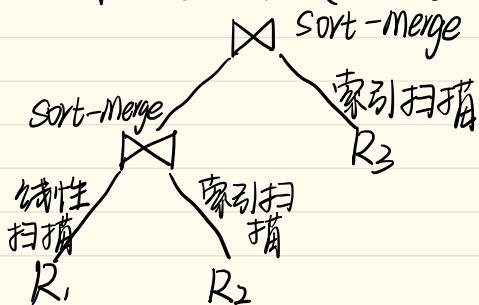
3. (1) A, C, E 分别为 R_1, R_2, R_3 的主键, 有 $V(R_2, C) = T(R_2) = 1500$
 $V(R_3, E) = T(R_3) = 750$, 连接结果大小 =

$$\frac{V(R_2, C) V(R_3, E)}{V(R_2, C) V(R_3, E)} = 1000$$

实现策略: 先使用动态规划的方法来选择连接顺序

	R_1	R_2	R_3		R_1, R_2	R_2, R_3		R_1, R_2, R_3
大小	1000	1500	750	大小	1000	1500	大小	1000
代价	0	0	0	代价	0	0	代价	1000
最佳顺序	R_1	R_2	R_3	顺序	$R_1 \bowtie R_2$	$R_2 \bowtie R_3$	顺序	$(R_1 \bowtie R_2) \bowtie R_3$

确定连接顺序为 $(R_1 \bowtie R_2) \bowtie R_3$, 每一步使用的方法 =



(2) R_1 与 R_2 连接结果大小为

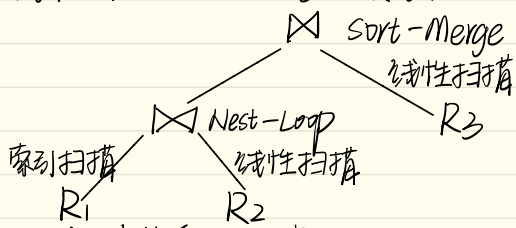
$$\frac{T(R_1)T(R_2)}{\max\{V(C, R_1), V(C, R_2)\}} = 1364$$

再与 R_3 连接, 结果大小为

$$\frac{T(R_1 \bowtie R_2)T(R_3)}{\max\{V(E, R_2), V(E, R_3)\}} = 10227$$

实现策略 = 先动态规划确定连接顺序

	R_1	R_2	R_3		$R_1 \bowtie R_2$	$R_2 \bowtie R_3$		$R_1 \bowtie R_2 \bowtie R_3$
大小	1000	1500	750	大小	1364	11250	大小	10227
代价	0	0	0	代价	0	0	代价	1364
顺序	R_1	R_2	R_3	顺序	$R_1 \bowtie R_2$	$R_2 \bowtie R_3$	顺序	$(R_1 \bowtie R_2) \bowtie R_3$



4. 对应的关系代数表达式 =

$$\pi_{Name}(\sigma_{Title="Jane Eyre"}((L \bowtie S) \bowtie B))$$

逻辑查询计划树

