DB Exercise Class

2014.6

Assignment Analysis

- **Normalization Assignment**
- **Query Language Assignment**
- **SQL Assignment**
- **Assignment for Chapter 8-11**
- **Assignment for Chapter 12-17**

Normalization Assignment

1. Suppose you are given a relation R with four attributes ABCD. Assume that the following FDs are the only dependencies that hold for R.

We have the FD set: F = {AB→A, AB→C, AB→D, C→A, D→B}

(1) Compute the attribute closure AB+ and A+ with respect to F with detailed explanation.

Solution:

AB^+ = ABCD， A^+=A。

AB^+内一定包含AB。因为AB→C，所以AB^+内包含C。因为AB→D，所以AB^+内包含D，综上所述AB^+ = ABCD。

A^+=A，因为从A无法推出其他属性，A^+只包含A，所以A^+=A。

Normalization Assignment

1. Suppose you are given a relation R with four attributes ABCD. Assume that the following FDs are the only dependencies that hold for R.

We have the FD set: F = {AB→A, AB→C, AB→D, C→A, D→B}

(2) Identify the candidate key(s) for R. Briefly explains the reason.

Solution:

$A^+ = A$， $B^+= B$， $C^+=CA$， $D^+=DB$，

$AB^+=ABCD$， $AC^+=AC$， $AD^+=ABCD$， $BC^+=ABCD$， $BD^+=BD$， $CD^+=ABCD$

所以R的候选码有AB， AD， BC， CD。

Normalization Assignment

1. Suppose you are given a relation R with four attributes ABCD. Assume that the following FDs are the only dependencies that hold for R.

We have the FD set: F = {AB→A, AB→C, AB→D, C→A, D→B}

(3) Find out the trivial functional dependencies in F and prove that they are trivial FDs.

Solution:

AB→A是平凡函数依赖。

因为在AB→A中，函数依赖右边只有一个属性A，而且A是左边的属性AB的子集，所以它是平凡函数依赖。

Normalization Assignment

1. Suppose you are given a relation R with four attributes ABCD. Assume that the following FDs are the only dependencies that hold for R.

We have the FD set: F = {AB→A, AB→C, AB→D, C→A, D→B}

(4) Write down a minimal cover for the set F. Briefly explain the reason.

Solution:

①将FD中的所有依赖右边化为单一元素，**FD = { AB→A, AB→C, AB→D, C→A, D→B }**已经满足。

②去掉**FD**中的所有依赖左边的冗余属性。

在依赖**AB→A**中，因为**A^+=A**，其中包含**A**，所以**B**是冗余的，可去除。**FD = { A→A, AB→C, AB→D, C→A, D→B }**

③ 去掉**FD**中所有冗余依赖关系。具体做法为从**FD**中去掉某关系，如去掉**X→Y**，然后在**FD**中求**X^+**，如果**Y**在**X^+**中，则表明**X→Y**是多余的。需要去掉。

(a) 如果**FD**去掉**A→A**，**FD**将等于**{AB→C, AB→D, C→A, D→B }**，在**FD**中求**A^+=A**，**A**在**A^+**中，所以**A→A**是多余的，可去掉**A→A**。所以**FD={AB→C, AB→D, C→A, D→B }**

(b) 接着如果**FD**去掉**AB→C**，**FD**将等于**{AB→D, C→A, D→B }**，在**FD**中求**AB^+=ABD,AB^+**不包含**C**，所以**AB→C**不是多余的，不能去掉。

(c)同理，**AB→D, C→A, D→B**都不是多余，故不能去掉。

所以最小函数依赖集为**{ AB→C, AB→D, C→A, D→B }**。

Normalization Assignment

2. Suppose you are given a relation R with six attributes ABCDEG. Assume that the following FDs are the only dependencies that hold for R.

We have the FD set: F = {A→B, B→C, AD→G, D→E}

Identify the best normal form that R satisfies (1NF, 2NF, 3NF and BCNF). If R is not in BCNF, decompose it into a set of BCNF relations that preserve the dependencies.

Solution:

1NF。因为R的码为AD，A→B违反了部分依赖，所以R甚至不符合2NF。所以它只符合1NF。

BCNF分解过程：

（1）违反函数依赖的有A→B，D→E，A→C(A→B, B→C推出的)

按A→B分解成AB，ACDEG；ACDEG按D→E分解成DE，ACDG；ACDG按A→C分解成AC，ADG

所以最后结果是AB，DE，AC，ADG （分解顺序不同也都是这个结果）

（2）违反函数依赖的有A→B，D→E，B→C

按B→C分解成BC，ABDEG；ABDEG按A→B分解成AB，ADEG；ADEG按D→E分解成DE，ADG

所以最后结果是BC，AB，DE，ADG

综上所述，BCNF分解为AB， DE，AC，ADG 或者BC，AB，DE，ADG。

Query Language Assignment

4.3

(5)Find the sids of suppliers who supply every part.

Solution:

(5) RA: $(\pi_{\mathrm{sid},pid} Catalog)/(\pi_{pid} Parts)$

TRC: $\{\mathrm{T} \mid \exists \mathrm{T1} \in \mathrm{Catalog}(\forall \mathrm{X} \in \mathrm{Parts}(\exists \mathrm{T2} \in \mathrm{Catalog}(\mathrm{T2.pid} = \mathrm{X.pid} \wedge \mathrm{T2.sid} = \mathrm{T1.sid})) \wedge \mathrm{T.sid} = \mathrm{T1.sid})\}$

Query Language Assignment

4.3

(11) Find the *pids* of the most expensive parts supplied by suppliers named Yosemite Sham.

Solution:

(11)RA: $\rho(R1, \pi_{sid}\sigma_{sname='Yosemite\ Sham'}\mathrm{Suppliers})$

$\rho(R2, R1 \bowtie \mathrm{Catalog})$

$\rho(R3, R2)$

$\rho\left(R4(1 \rightarrow sid, 2 \rightarrow pid, 3 \rightarrow cost), \sigma_{R2.cost<R3.cost}(R2 \times R3)\right)$

$\pi_{pid}(R2 - \pi_{sid,pid,cost}R4)$

TRC: $\{T \mid \exists T1 \in \mathrm{Catalog}(\exists X \in \mathrm{Suppliers}(X.sname =' Yosemite\ Sham' \wedge X.sid = T1.sid) \wedge \neg(\exists S \in \mathrm{Suppliers}(S.sname =' Yosemite\ Sham' \wedge \exists Z \in \mathrm{Catalog}(Z.sid = S.sid \wedge Z.cost > T1.cost))) \wedge T.pid = T1.pid)\}$

SQL Assignment

2. Courses(<u>cid</u>,cname,hour)
   Students(<u>sid</u>,sname,email,grade)
   Teachers(<u>tid</u>,tname,email,salary)
   Choices(<u>no</u>,sid,tid,cid,score)

（8）请将教授db课程的老师的工资增加1000元。

Solution:

```
UPDATE Teachers T
SET     T.salary = T.salary + 1000
WHERE  T.tid IN ( SELECT  A.tid
FROM    Choices A, Courses C
WHERE   A.cid = C.cid AND C.cname = 'db')
```

SQL Assignment

2. Courses(<u>cid</u>,cname,hour)

   Students(<u>sid</u>,sname,email,grade)

   Teachers(<u>tid</u>,tname,email,salary)

   Choices(<u>no</u>,sid,tid,cid,score)

（12）请找出分数多于90分的学生人数超过20人的课程名

Solution:

```
SELECT   C.cname
FROM     Courses C
WHERE    C.cid IN (SELECT    A.cid
                   FROM      Choices A
                   WHERE     A.score > 90
                   GROUP BY  A.cid
                   HAVING    COUNT(*)>20)
```

默认每个学生只选择一门相同cid的课程。

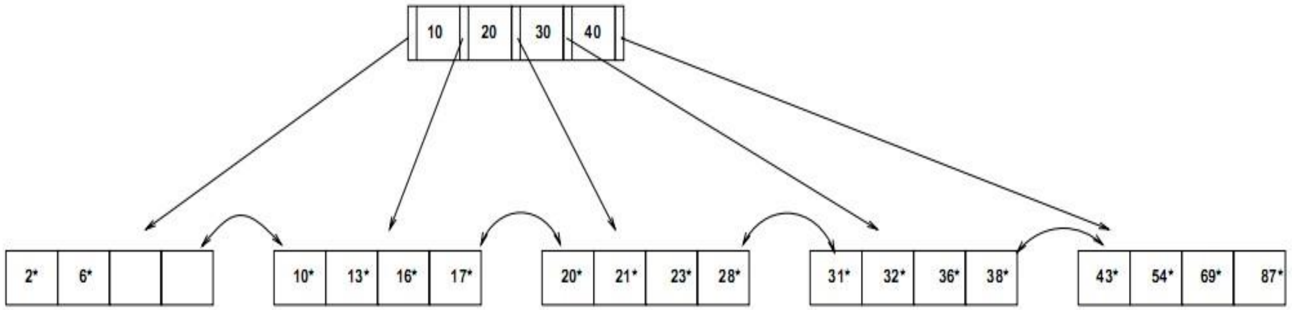
Assignment for Chapter 8-11

10.4 Suppose that a page can contain at most four data values and that all data values are integers. Using only B+ trees of order 2, give examples of each of the following:

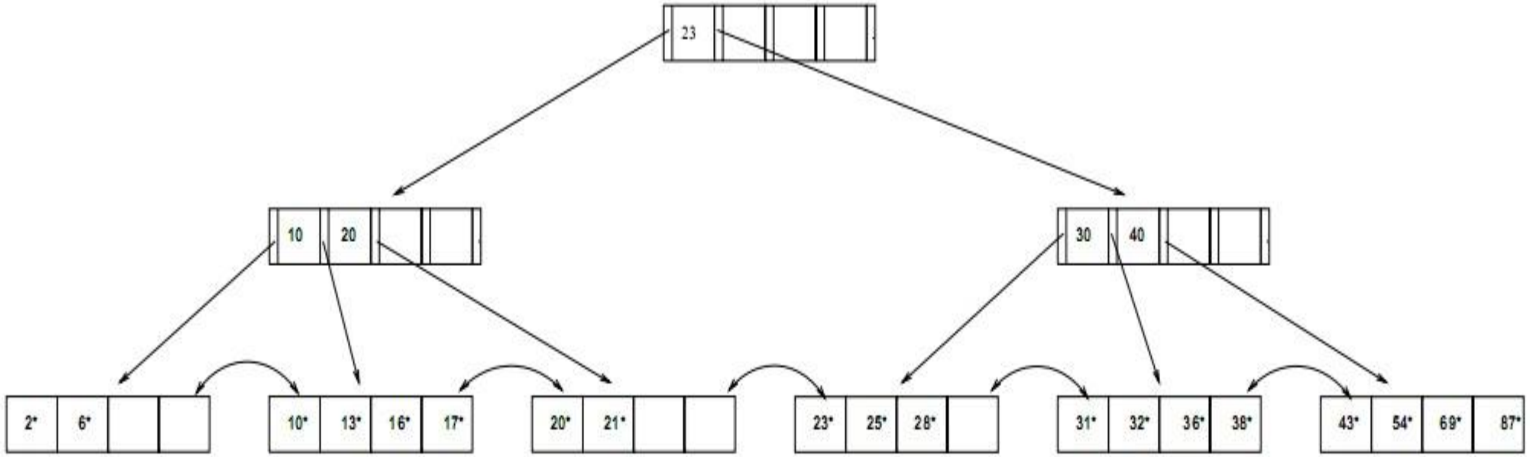
(1)A B+ tree whose height changes from 2 to 3 when the value 25 is inserted. Show your structure before and after the insertion.

Assignment for Chapter 8-11

Solution:



B+ tree before inserting 25



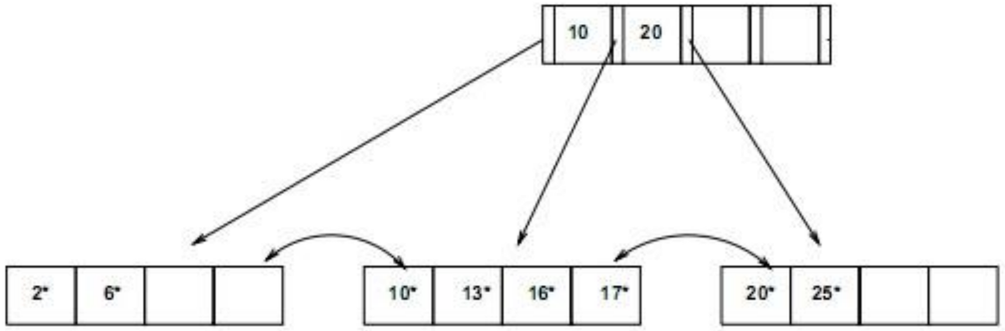
B+ tree after inserting 25

Assignment for Chapter 8-11

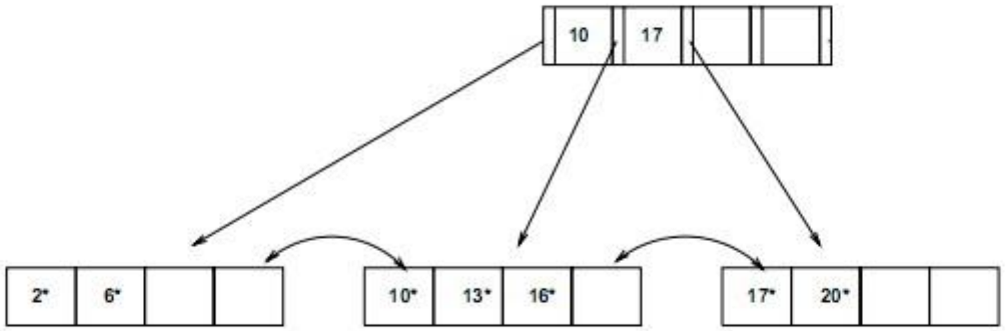
10.4

(2)A B+ tree in which the deletion of the value 25 leads to a redistribution. Show your structure before and after the deletion.

Solution:



B+ tree before deleting 25



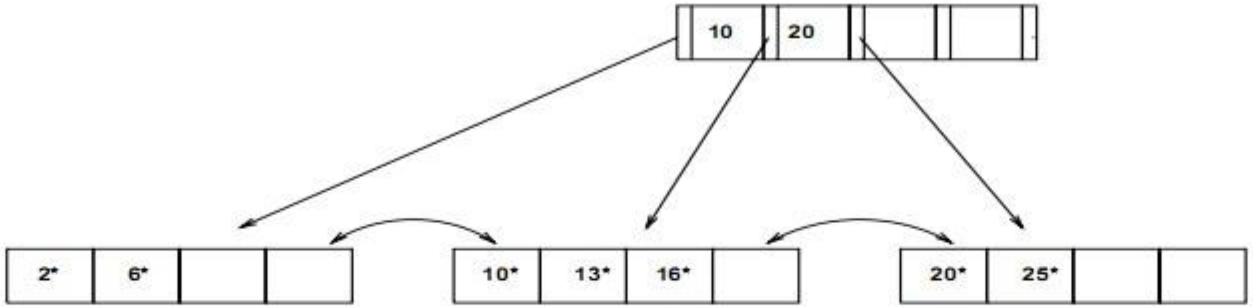
B+ tree after deleting 25

Assignment for Chapter 8-11

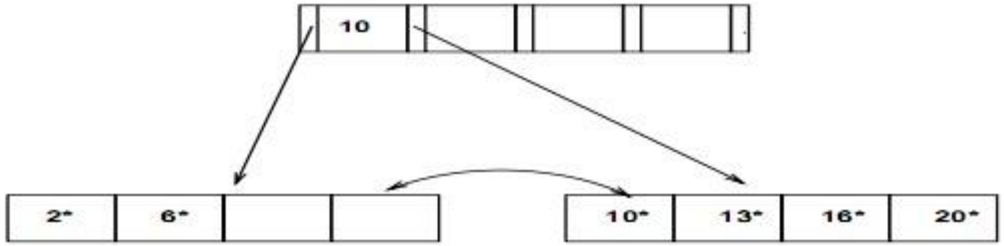
10.4

(3)A B+ tree in which the deletion of the value 25 causes a merge of two nodes but without altering the height of the tree.

Solution:



B+ tree before deleting 25



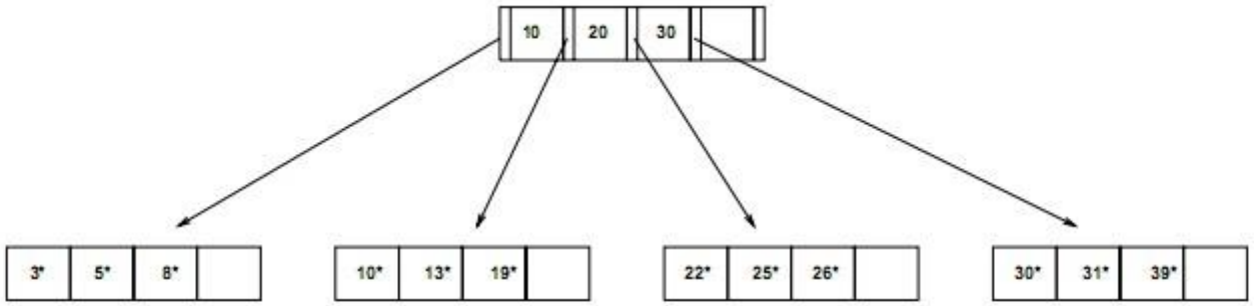
B+ tree after deleting 25

Assignment for Chapter 8-11

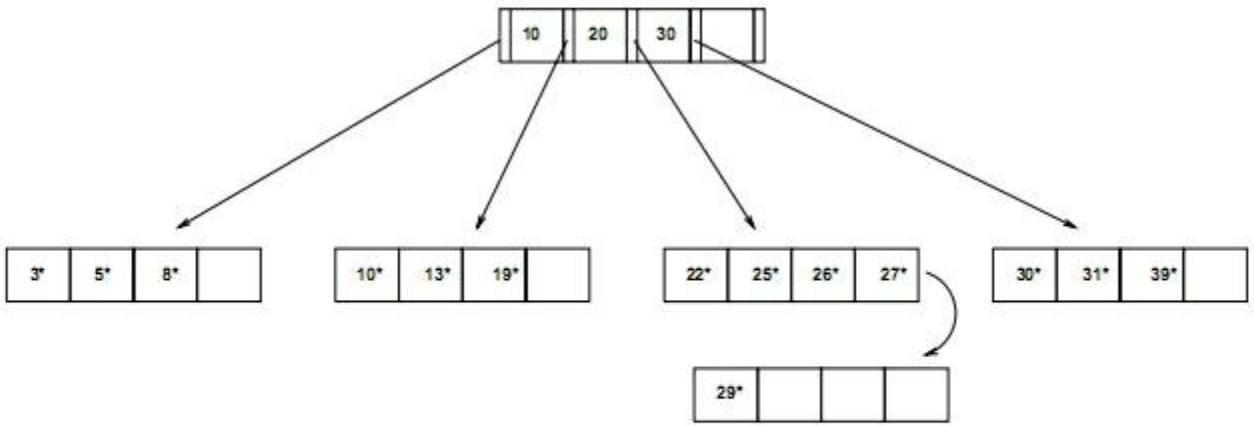
10.4

(4)An ISAM structure with four buckets, none of which has an overflow page. Further, every bucket has space for exactly one more entry. Show your structure before and after inserting two additional values, chosen so that an overflow page is created.

Solution:



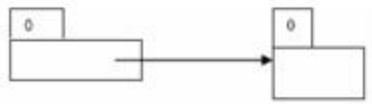
ISAM tree before inserting 27,29



ISAM tree after inserting 27,29

Assignment for Chapter 8-11

4. Suppose that we are using extendable hashing index that contains the following search-key values K: 1,4,5,7,8,2,20. Assuming the search-key values arrive in the given order (i.e. 1 being the first coming key and 20 being the last one). The initial configuration of the structure and the binary form (assuming the choice of bits starting from Least Significant Bit (LSB)) of all keys are given in the diagram below.



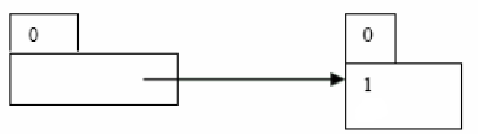
1	4	5	7	8	2	20
001	100	101	111	000	010	100

Show the extendable hash structure for each insertion of the above key values file if the hash function is *h(x) = x mod 8* and buckets can hold two keys.

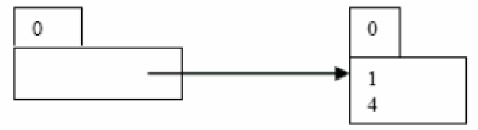
Assignment for Chapter 8-11

Solution:

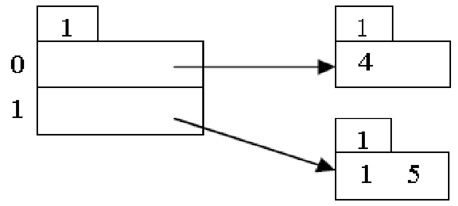
① Insert 1:



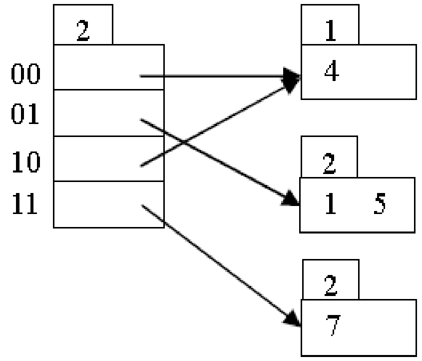
② Insert 4:



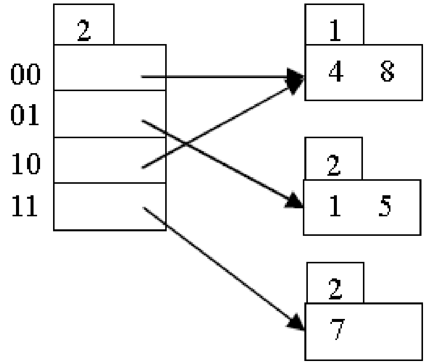
③ Insert 5:



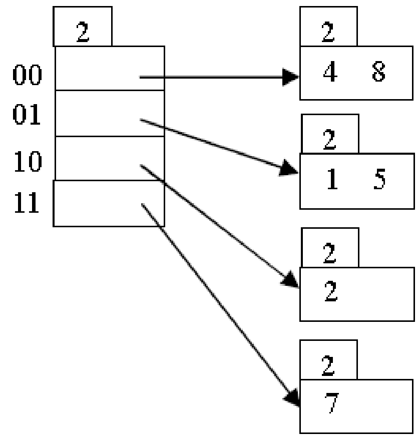
④ Insert 7:



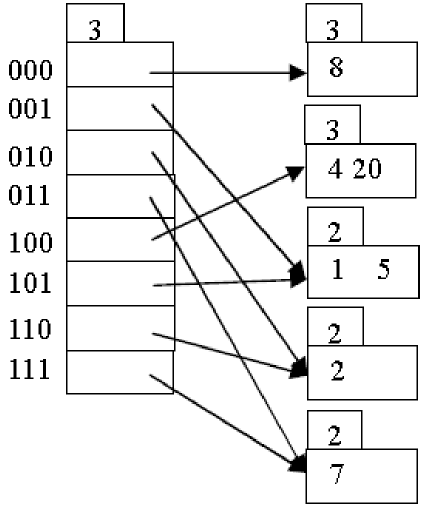
⑤ Insert 8:



⑥ Insert 2:



⑦ Insert 20:



Assignment for Chapter 12-17

Exercise14.4 Consider the join $R \bowtie_{R.a=S.b} S$, given the following information about the relations to be joined. The cost metric is the number of page I/Os unless otherwise noted, and the cost of writing out the result should be uniformly ignored.

Relation R contains 10,000 tuples and has 10 tuples per page. Relation S contains 2000 tuples and also has 10 tuples per page. Attribute *b* of relation S is the primary key for S. Both relations are stored as simple heap files. Neither relation has any indexes built on it.52 buffer pages are available.

Assignment for Chapter 12-17

(1) What is the cost of joining R and S using a page-oriented simple nested loops join? What is the minimum number of buffer pages required for this cost to remain unchanged?

Solution:

公式在中文课本338页倒数第三行，改进后的嵌套循环连接算法

R outer: 1000+1000*200 = 201000

S outer: 200+200*1000 = 200200

最小可能的缓冲区页数是3个，一个用于扫描读取R，一个用于扫描读取S，一个用于输出缓存。

Assignment for Chapter 12-17

(2)What is the cost of joining R and S using a block nested loops join? What is the minimum number of buffer pages required for this cost to remain unchanged?

Solution:

公式在中文课本339页倒数第二段。

R outer: $1000+200*\lceil 1000/(52-2)\rceil$

S outer: $200+1000*\lceil 200/(52-2)\rceil$

因为可用缓冲区的数目为52个，如果仅使用51个的话，扫描的次数会达到$\lceil 200/(51-2)\rceil = 5$次，大于原来的$\lceil 200/(52-2)\rceil = 4$次，所以要保持算法开销不变的话最小可能的缓冲区页数为5。

Assignment for Chapter 12-17

(3) What is the cost of joining R and S using a sort-merge join? What is the minimum number of buffer pages required for this cost to remain unchanged?

Solution:

排序归并算法我们一般不考虑改进。

公式：当B>$\sqrt{M}$时，3*（M+N）

当B<=$\sqrt{M}$时，4M+4N+M+N. 排序遍数计算详见中文书319页

本题中B>$\sqrt{M}$（52>$\sqrt{1000}$），3*（M+N）=3*（1000+200）=3600

最小可能的缓冲区页数为25.在第一遍排序中R大约被分解为20趟大小为50的数据页，S大约被分解为4趟大小为50的数据页。需要的这24个缓冲区页数再加上用于输出缓存的1个缓冲区页就是25个缓冲区页数。如果使用少于25个缓冲区页就不可能保持原一趟排序归并的开销。

Assignment for Chapter 12-17

(4) What is the cost of joining R and S using a hash join? What is the minimum number of buffer pages required for this cost to remain unchanged?

Solution:

公式详见中文书347页。

开销为3*(M+N)=3*(1000+200)=3600

需要的缓冲区最小数目为B>$\sqrt{f * M}$ （式中f表示哈希表与部分表相比，假定尺寸增大的因子，详细解释见中文书346到347页。）

Assignment for Chapter 12-17

5. Consider the schedule S that consists of four transactions as follows: S = < T2_X(A), T2_W(A), T1_S(B), T1_R(B), T2_X(B), T3_S(C), T3_R(C), T1_X(C), T4_X(B), T4_W(B), T3_S(A)>. The notation is self-explanatory. For example, T1_R(B) means that transaction T1 reads item B.

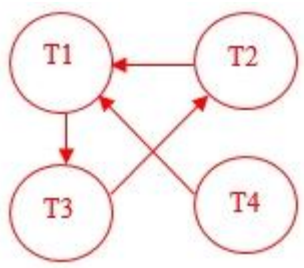
T1	T2	T3	T4
	X(A)		
	W(A)		
S(B)			
R(B)			
	X(B)		
		S(C)	
		R(C)	
X(C)			
			X(B)
			W(B)
		S(A)	

Show the waits-for graph and determine whether there is a deadlock. How to resolve the deadlock if it exists?

Assignment for Chapter 12-17

Solution:

waits-for graph:



There is a deadlock since a circle exists in the waits-for graph.

A deadlock is resolved by aborting a transaction that is on a cycle and releasing its locks.