National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Advance Database Concepts**	**Course Code:**	**CS4064**
Program:	**BS (Computer Science)**	**Semester:**	**Spring 2025**
Out Date:	**21-Mar-2025**	**Total Marks:**	
Due Date:	**Thu 27-Mar-2025** ***(Start of class)***	**Weight:**	
		Page(s):	**2**
Assignment:	**3 (Indexing Structures) - Solution**		

Instructions:

- Use any valid assumption where needed.
- You are required to submit the hard copy of your assignment at the start of your class.
- For any queries, please contact your TA.

Take the following assumptions for the block size and order file size to solve the questions:

Q1. Block Size **B= 4096 b**ytes and File Records (fixed length and un-spanned) **r= 10b**illion
Q2. Block Size **B= 8192 b**ytes and File Records (fixed length and un-spanned) **r= 100m**illion

A block pointer (P) is 5 bytes long and a record pointer (P_R) is 6 bytes long. Record length (R) is 120 bytes long and assume that the size of each field is 10 bytes long.

a. Suppose that the file is *ordered* by the key field OrderNo and we want to construct a *primary* index on OrderNo. Calculate (i) the index blocking factor bfr_i (which is also the index fan-out *fa);* (ii) the number of first-level index entries and the number of first-level index blocks; (iii) the number of levels needed if we make it into a multilevel index; (iv) the total number of blocks required by the multilevel index; and (v) the number of block accesses needed to search for and retrieve a record from the file given its OrderNo value using the primary index.

b. Suppose that the file is not *ordered* by the key field OrderNo and we want to construct a *secondary* index on OrderNo. Repeat the previous (part a) for the secondary index and compare it with the primary index.

c. Suppose that the file is not *ordered* by the non-key field CustID and we want to construct a *secondary* index on CustID, with an extra level of indirection that stores record pointers. Assume there are 50,000 distinct values of CustID and that the Order records are evenly distributed among these values. Calculate (i) the index blocking factor bfr, (which is also the index fan-out *fa);* (ii) the number of blocks needed by the level of indirection that stores record pointers; (iii) the number of first level index entries and the number of first-level index blocks; (iv) the number of levels needed if we make it into a multilevel index; (v) the total number of blocks required by the multilevel index and the blocks used in the extra level of indirection; and (vi) the approximate number of block accesses needed to search for and retrieve all records in the file that have a specific CustID value, using the index.

d. Suppose that the file is *ordered* by the non-key field CustID and we want to construct a *clustering index* on CustID that uses block anchors (every new value of CustID starts at the beginning of a new block). Assume there are 50,000 distinct values of CustID and that the Order records are evenly distributed among these values. Calculate (i) the index blocking factor bfr, (which is also the index fan-out *fa);* (ii) the number of first-level index entries and the number of first-level index blocks; (iii) the number of levels needed if we make it into a multilevel index; (iv) the total number of blocks required by the multilevel index; and (v) the number of block accesses needed to search for and retrieve all records in the file that have a specific CustID value, using the clustering index (assume that multiple blocks in a cluster are contiguous).

e. Suppose the file is not ordered by the key field OrderNo and we want to construct a B^+-tree access structure (index) on OrderNo. Calculate (i) the orders p and p leaf of the B^+-tree; (ii) the number of leaf-level blocks needed if blocks are approximately 70% full (rounded up for convenience); (iii) the number of levels needed if internal nodes are also 70% full

(rounded up for convenience); (iv) the total number of blocks required by the B^+-tree; and (v) the number of block accesses needed to search for and retrieve a record from the file--given its OrderNo value--using the B^+-tree.

f. Repeat (part e), but for a B-tree *rather than for a* B^+-tree. Compare your results with the B^+-tree.

Q1 & Q2. Answer:

Q NO-1

B = 4096 bytes
r = 10 billion = 10000000000
P = 5 bytes
Pr = 6 bytes
R = 120 bytes
size of each field = 10 bytes

(a) File ordered by Key OrderNO
Primary index on OrderNO

(i) Bfri = ?

As index block of primary index has index field and block pointer so size of index entry :-

$R_i = 10 + 5 = 15$ bytes

$$Bfr_i = \left\lfloor \frac{B}{R_i} \right\rfloor = \frac{4096}{15}$$

$$\boxed{Bfr_i = 273}$$

(ii) Number of first level index entries and number of first level index blocks

For primary index,
number of first level index entries is number of blocks in file.
So,

$$b = \left\lceil \frac{r}{Bfr} \right\rceil \qquad Bfr = ?$$

$$Bfr = \left\lfloor \frac{B}{R} \right\rfloor = \frac{4096}{120} = 34$$

$$b = \left\lceil \frac{10\,000\,000\,000}{34} \right\rceil = 294117648 \text{ blocks}$$

So, number of 1st level index entries = 294117648 entries

And number of 1st level index blocks $= \left\lceil \frac{r_i}{Bfr_i} \right\rceil$

$$= \left\lceil \frac{294117648}{273} \right\rceil$$

no. of 1st level index blocks = 1077355 blocks

(iii) No. of levels for multi level index

level 1 blocks = 1077355
level 2 used as primary index for level 1
So, $r_i = b_1$
level 2 blocks $= \left\lceil \frac{1077355}{273} \right\rceil = 3947$ blocks

Level 3 blocks = $\frac{3947}{273}$ = 15

Level 4 blocks = $\frac{15}{273}$ = 1 (root)
↓
top level

So
total levels = 4

(iv) Total no. of blocks for multilevel index

Total blocks = $b_1 + b_2 + b_3 + b_4$
= 1077355 + 3947 + 15 + 1

Total blocks = 1081318 blocks

(v) No. of block accesses using PI

No. of block access using PI = X + 1 = 4 + 1
X ⇒ levels

No. of block access = 5

(b) Unordered file:
Secondary index on key field OrderNo.

(i) Bfri

$Bfri = \left\lfloor \frac{B}{Ri} \right\rfloor$

As index record has field and block pointer

So,

$Ri = 10 + 5 = 15$ bytes

$Bfri = \left\lfloor \frac{4096}{15} \right\rfloor = 273$

(ii) In secondary index (unordered file and index on unique key), so

$ri = r$

Number of first level index entries = number of records in file

Number of 1st level index entries = 1000000000

And

Blocks $= \left\lceil \frac{ri}{Bfri} \right\rceil = \frac{1000000000}{273}$

Number of 1st level index blocks = 36630037 blocks

(iii) No. of levels :-

level 2 blocks $= \left\lceil \frac{36630037}{273} \right\rceil = 134176$

⇓

primary index for level 1 so $ri = b_1$

level 3 blocks $= \left\lceil \frac{134176}{273} \right\rceil = 492$

level 4 blocks $= \left\lceil \frac{492}{273} \right\rceil = 2$

Level 5 blocks $= \left\lceil \frac{2}{273} \right\rceil = 1$ (root)
⇓ top level

So total levels = 5

(iv) Total blocks

$= 36630037 + 134176 + 492 + 2 + 1$

Total blocks = 36764708 blocks

(V) No. of block access using SI

As SI on unique attribute
So,
block access = X + 1 = 5 + 1
No. of block accesses = 6
⇒ So no. of levels increased as compared to PI

(c) Unordered File
Secondary index on non key CustID
level of indirection for record pointers.

50,000 distinct values of custID & even distribution

So each distinct value has records $= \frac{10000000000}{50000} = 200,000$

(i) Bfr$_i$

R_i = 15 bytes

$Bfr_i = \left\lfloor \frac{B}{R_i} \right\rfloor = 273$

(ii) No. of blocks for level of indirection to store record pointers

Against each distinct value of custID, we have 200000 pointers (record), so total size of record pointers for each value.

$= 200,000 \times 6 = 1200000$ bytes

So for 1 value of custID, no. of blocks of record pointers $= \lceil \frac{1200000}{4096} \rceil$

$= 292.97$

level of indirection = 293 blocks for 1 custID

So total blocks of level of indirection $= 293 \times 50000 = 14650000$ blocks

(iii) No. of 1st level index entries

$r_i = d$

No. of 1st level index entries = 50,000

No. of 1st level index blocks $= \lceil \frac{50,000}{Bfr_i} \rceil$

$= 184$ blocks

(iv) No. of levels = ?

level 1 blocks = 184

level 2 blocks $= \lceil \frac{184}{273} \rceil = 1$ (top level)

Total levels = 2

(V) Total blocks

= 184 + 1 (index blocks)

Total blocks of index = 185 blocks

Total blocks for level of indirection = 14650000 blocks

Total no. of blocks of multilevel index & level of indirection = 185 + 14650000
= 14650185 blocks

(vi) No. of block access using SI

For each distinct value, 200,000 records, if we assume all records are in different blocks as file is unordered and $s < b$

selectivity = s = 200,000 ⇒ for each ID

Block access = $x + s +$ level of indirection
= 2 + 293 + 200,000

No. of block access = 200,295

(d) Ordered file
Cluster index on non key custID
Block anchoring
50,000 distinct values of custID

each custID has $= \left\lceil \frac{10000000000}{50000} \right\rceil = 200,000$ records

(i) Bfri

$R_i = 10 + 5 = 15$ bytes

$Bfr_i = \left\lfloor \frac{B}{R_i} \right\rfloor = \left\lfloor \frac{4096}{15} \right\rfloor = 273$

(ii)

Number of 1st level index entries $= d = 50,000$

Number of 1st level index blocks $= \left\lceil \frac{50000}{273} \right\rceil = 184$

(iii) No. of levels.

level 1 blocks = 184
level 2 as primary index for level 1
So, level 2 blocks $= \left\lceil \frac{184}{273} \right\rceil = 1$

So 2 levels

(iv) Total blocks

Total blocks $= 184 + 1 = 185$

(V) No of block access using CI

Block access = $X + \left\lceil \frac{s}{Bfr} \right\rceil$

X = no of contiguous blocks for each CustID

$= 2 + \left\lceil \frac{200,000}{34} \right\rceil$

Block accesses = 5885 blocks

(e) Unordered file
B+ tree access index on Key OrderNo

(i) Order p & pleaf

Order p is max no of block pointers that can fit in one internal node.

$p \times P + (p-1)(\text{Keysize}) \leq B$
$p(5) + (p-1)(10) \leq 4096$
$5p + 10p - 10 \leq 4096$
$15p - 10 \leq 4096$
$15p \leq 4106$
$p \leq 273.73$

$\boxed{p = 273}$ => fan out (fo) of internal node

order pleaf is max number of Key Data pointer pair that can fit in one leaf node.

$P_{leaf}(Key + P_R) + P \leq 4096$

$P_{leaf}(10+6) + 5 \leq 4096$

$16\,P_{leaf} + 5 \leq 4096$

$16\,P_{leaf} \leq 4091$

$P_{leaf} \leq 255.69$

$\boxed{P_{leaf} = 255}$ => fanout (fo) of leaf

(ii) No. of leaf level blocks needed if blocks are 70% full (round up)

$\lceil 0.7 \times 255 \rceil = 179$

So leaf level blocks = $\frac{r_i}{179}$

$r_i = r$ as: unordered file & index on key attribute. (all record in leaves)

leaf level blocks = $\left\lceil \frac{10,000,000}{179} \right\rceil$

leaf level blocks = 55865922

(iii) No. of levels if internal nodes are also 70% full (round up)

$\lceil 0.7 \times 273 \rceil = 192$

level 1 blocks = 55865922
(leaf)

level 2 blocks = $\lceil \frac{55865922}{192} \rceil$ = 290969

level 3 blocks = $\lceil \frac{290969}{192} \rceil$ = 1516

level 4 blocks = $\lceil \frac{1516}{192} \rceil$ = 8

level 5 blocks = $\lceil \frac{8}{192} \rceil$ = 1 (root)

So total 5 levels.
Another way = $\log_{fo}(b_1)$

$= \lceil \log_{255}(55865922) \rceil + 1$

$= 4 + 1 = 5$ levels.

(iv) Total number of blocks

$= 55865922 + 290969 + 1516 + 8 + 1$

Total blocks = 56158416

(v) No. of block access.
Block access = $X + 1 = 5 + 1 = 6$

(f) B Tree
unordered file
BTree index on Key OrderNO

(i) Orderp & pleaf
in B tree, structure of all nodes is same so orderp = order pleaf

Orderp :-

$$P \times P + (P-1)(Key + PR) \leq 4096$$
$$P \times 5 + (P-1)(16) \leq 4096$$

$$5P + 16P - 16 \leq 4096$$
$$21P \leq 4112$$

$$P \leq 195.80$$

$$\boxed{P = 195} \Rightarrow \text{orderp \& pleaf}$$

(ii) No. of leaf level blocks if 70% full

$$\text{leaf level blocks} = \left\lceil \frac{10\,000\,000\,000}{195} \right\rceil = 51282052 \text{ blocks}$$

(round up)

$$\lceil 0.7 \times 195 \rceil = 137$$

$$\text{leaf level blocks} = \left\lceil \frac{10\,000\,000\,000}{137} \right\rceil$$
$$= 72992701$$

(iii) No. of levels if internal nodes are also 70% full.

$\lceil 0.7 \times 195 \rceil = 137$

Level 2 blocks $= \left\lceil \frac{729927O1}{137} \right\rceil$

Level 2 blocks = 532794
Level 3 blocks = 3890
Level 4 blocks = 29
Level 5 blocks = 1 (root)

5 levels.

Another way:-

$\log_{60}(b_1) = \lceil \log_{137}(729927O1) \rceil + 1 = 4 + 1$

= 5 levels

(iv) Total blocks.

= 729927O1 + 532794 + 3890 + 29 + 1
Total blocks = 735294I5 blocks

(V) No. of block access

= X + 1 = 5 + 1 = 6

in B Tree capacity of node decreases as compared B+ tree as in B tree it was 137 & B+ tree was 179 for leaf which can lead to more levels in B tree & hence searching is difficult as compared to B+ tree. And more blocks req. in Btree

Q2. $B = 8192$ bytes

$r = 100$ million $= 100\,000\,000$

$P = 5$ bytes

$P_R = 6$ bytes

$R = 120$ bytes

size of each field = 10 bytes

(a) Ordered file

Primary index on Key Order No.

(i) Bfri

$R_i = 10 + 5$ bytes $= 15$ bytes

$$Bfri = \left\lfloor \frac{B}{R_i} \right\rfloor = \left\lfloor \frac{8192}{15} \right\rfloor = 546$$

(ii)

Number of first level index entries $= b$

$$b = \left\lceil \frac{r}{bfr} \right\rceil$$

$$bfr = \left\lfloor \frac{B}{R} \right\rfloor = 68$$

$$b = \left\lceil \frac{100\,000\,000}{68} \right\rceil = 1470589$$

So,

no. of entries in 1st level $= 1470589$

No. of blocks $= \left\lceil \frac{1470589}{546} \right\rceil$

No. of 1st level index blocks $= 2694$

(iii) No. of levels

level 2 blocks $= \left\lceil \frac{2694}{546} \right\rceil = 5$

level 3 blocks $= 1$

So 3 levels

Another way $= \lceil \log_{bfri}(r_1) \rceil = \log_{546}(14705589)$

$= 3$ levels

(iv) Total blocks

$= 2694 + 5 + 1$

Total blocks $= 2700$

(v) No. of block access $= x + 1 = 3 + 1$

$= 4$ block access

(b) Unordered file
SI on key OrderNO.

(i) Bfri

$Bfri = \left\lfloor \frac{B}{Ri} \right\rfloor = \frac{8192}{15} = 546$

(ii) No. of 1st level index entries $= r = 100\,000000$

No. of 1st level index blocks $= \left\lceil \frac{100\,000\,000}{546} \right\rceil = 183151$

(iii) No. of levels

Level 2 blocks $= \left\lceil \frac{183151}{546} \right\rceil = 336$

Level 3 blocks $= \left\lceil \frac{336}{546} \right\rceil = 1$

3 levels

Another way $= \left\lceil \log_{546}(100\,000\,000) \right\rceil = 3$ levels

(iv) Total blocks

$= 183151 + 336 + 1$

Total Blocks $= 183488$

(v) No. of block access

Block access $= X + 1 = 3 + 1 = 4$

$\Downarrow$

as unique orderNo.

So using SI on orderNo requires alot of blocks for multilevel index as compared to PI

(c) Unordered file

SI on non Key CustID

Level of indirection

50000 distinct CustID

So each custID has = 2000 records/pointer records

(i) Bfri

$$Bfr_i = \left\lfloor \frac{B}{R_i} \right\rfloor = \frac{8192}{15} = 546$$

(ii) No. of blocks for level of indirection

For one CustID, total = 2000 × 6 = 12000 bytes record pointers size

So no. of blocks of 1 CustID $= \left\lceil \frac{12000}{8192} \right\rceil = 2$

So total no. of blocks for level of in Direction $= 50,000 \times 2 = 100000$

(iii) No. of 1st level index entries $= d = 50,000$

No. of 1st level index blocks $= \left\lceil \frac{50000}{546} \right\rceil = 92$

(iv) No. of levels

level 2 blocks $= \left\lceil \frac{92}{546} \right\rceil = 1$

2 levels

or

$\lceil \log_{596}(50000) \rceil = 2$ levels

(v) Total blocks

$= 92 + 1$

Total no. of blocks for multilevel index $= 93$

Total blocks for indirection level $= 100000$

total blocks $= 100093$

(vi) No. of block access

Block access = X + level of indirection + s of 1 custID

$= 2 + 2 + 2000$

Block access $= 2004$

(d) Ordered file

CI on non key custID

50,000 distinct custID

each custID (distinct) has = 2000 records.

(i) Bfri

$$Bfri = \left\lfloor \frac{B}{Ri} \right\rfloor = 546$$

(ii) No. of 1st level index entries = d = 50,000

No. of 1st level index blocks $= \left\lceil \frac{50000}{546} \right\rceil = 92$

(iii) No. of levels

level 2 blocks $= \left\lceil \frac{92}{546} \right\rceil = 1$

2 levels

or

$\left\lceil \log_{546}(50{,}000) \right\rceil = 2$ levels

(iv) Total blocks

Total blocks = 93

(v) No. of block access

block access $= X + \left\lceil \frac{s}{bfr} \right\rceil$

$= 2 + \left\lceil \frac{2000}{68} \right\rceil$

$= 2 + 30$

Block access = 32

(c) Unordered file
B+ tree index on Order No (Key)

(i) Order p & p_{leaf}.

$p \times P + (p-1)\,Key \leq 8192$
$5p + (p-1)\,10 \leq 8192$
$5p + 10p - 10 \leq 8192$
$15p \leq 8202$
$p \leq 546.8$
$\boxed{p = 546}$ => order p or fan out of internal node

$P_{leaf}(Key + P_R) + P \leq 8192$

$P_{leaf}(16) + 5 \leq 8192$

$16\,P_{leaf} \leq 8187$
$P_{leaf} \leq 511.68$

$\boxed{P_{leaf} = 511}$ => order P_{leaf} or fan out of leaf nodes.

(ii) No. of leaf level blocks if 70% full (round up)

$\lceil 0.7 \times 511 \rceil = 358$

No. of leaf level blocks $= \left\lceil \frac{100\,000\,000}{358} \right\rceil = 279330$

(iii) No. of levels if internal nodes are 70% full

$\lceil 0.7 \times 546 \rceil = 383$

Level 2 blocks $= \left\lceil \frac{279330}{383} \right\rceil = 730$

Level 3 blocks $= \left\lceil \frac{730}{383} \right\rceil = 2$

Level 4 blocks = 1

4 levels

Or $\lceil \log_{358}(279330) \rceil + 1 = 3 + 1 = 4$ levels

(iv) Total blocks
$= 279330 + 730 + 2 + 1$

Total blocks = 280063

(V) No. of block access

Block access = $X + 1 = 4 + 1 = 5$

(b) Unordered file
B Tree index on Key order No.

(i) Order p & pleaf

$pP + (p-1)(Key + P_R) \leq 8192$
$5p + (p-1)\,16 \leq 8192$
$5p + 16p - 16 \leq 8192$
$21p \leq 8208$
$p \leq 390.86$

$p = 390$ (p = Pleaf)

(ii) No. of leaf level blocks if 70% full

$\lceil 0.7 \times 390 \rceil = 273$

No. of leaf level blocks $= \left\lceil \frac{100\,000\,000}{273} \right\rceil = 366301$

(iii) No. of levels if internal nodes 70% full

level 2 blocks $= \left\lceil \frac{366301}{273} \right\rceil = 1342$

level 3 blocks $= \left\lceil \frac{1342}{273} \right\rceil = 5$

level 4 blocks $= 1$ (root)

4 levels

or

$\lceil \log_{273}(366301) \rceil + 1 = 3 + 1 = 4$ levels

(iv) Total blocks

$= 366301 + 1342 + 5 + 1$

Total blocks $= 367649$

(V) No. of block access

Block access $= x + 1 = 4 + 1 = 5$

In B tree capacity of node decreases so more blocks are required for index using B tree

Q3. Consider a DBMS that has the following characteristics:

- 2KB fixed-size blocks
- 6-bytes block pointer and 7-bytes record pointer
- 50-bytes block headers

We want to build an index on a search key that is 10 bytes long. Calculate the maximum number of records we can index with a

a. 4 Level B^+- tree index (including the root level)
b. 4 Level B-tree index (including the root level)

Answer:

a. Available capacity of block= 2048- 50= 1998

order of p: p * 6 + (p-1) * 10 < 1998, which gives us order p = (1998 + 10)/16 = 125

order of p_{leaf}: p * (10+7) + 6) < 1998, which gives us order p_{leaf} = (1998 - 6)/17 = 117

Level#	Nodes	Key-Entries	Pointers
1 (root)	1	124	125
2	125	125*124	125*125
3	125*125	125*125*124	125*125*125
4 (leaf)	125*125*125	125*125*125*117 *(data records pointers)*	

Max number of records we can index: 125*125*125*117= 228,515,625

b. Available capacity of block= 2048- 50= 1998

order of p: p * 6 + (p-1) * (10+7) < 1998, which gives us order p = (1998 -17)/23 = 86

Level#	Nodes	Key-Entries	Pointers
1 (root)	1	85	86
2	86	86*85	86*86
3	86*86	86*86*85	86*86*86
4 (leaf)	86*86*86	86*86*86*85	

Max number of records we can index: 85 + (86*85) + (86*86*85) + (86*86*86*85) = 54,700,815

Q4. Assume an un-ordered relation Customer *(CustID, Rating, Age)* is given with CustID as a key attribute and the data types of all attributes are integer. Size of the relation is 10,000 data blocks. Assume there is a four-level of B^+-tree index exist on CustID attribute. Moreover, one node of the B^+-tree is stored in one block on the disk. Take any valid assumption where needed.

Estimate the number of block fetches needed to compute the following queries:

a. SELECT * FROM Customer WHERE CustID= 100;
b. SELECT age FROM Customer WHERE CustID= 200 AND Rating= 5;
c. SELECT * FROM Customer WHERE CustID= 100 OR CustID= 200;
d. SELECT CustID, age FROM Customer WHERE Rating= 5;
e. SELECT * FROM Customer WHERE CustID>= 100;
f. SELECT COUNT(*) FROM Customer WHERE CustID>= 100;

Answer:

a. Number of block fetches = X+1 = 5
b. Number of block fetches = X+1 = 5
c. Number of block fetches = 2(X+1) = 10
d. Number of block fetches = b = 10,000
e. Number of block fetches = X + b = 10,004 (Assume all values are greater than 100 as a worst case)
f. Number of block fetches = X + b1 -1 (here b1 is the total number of leaf level blocks in B^+-tree index)