01. When would you want to use a B-tree instead of B^+ tree?

I. When sequential access to key values is never required.

II. When sequential access to key values is required.

III. When both sequential and direct access is required.

IV. When direct access is required without sequential access.

(a) III only

(b) II only

(c) I only

(d) IV only

(d) 1⇒ Use B-tree instead of B^+-tree.

(d) (IV) - When direct access is required without sequential access.

Because B-tree use direct access
⇒ B^+-tree use sequential access.

02. What is the maximum height of B-tree with order-m and having N-keys?

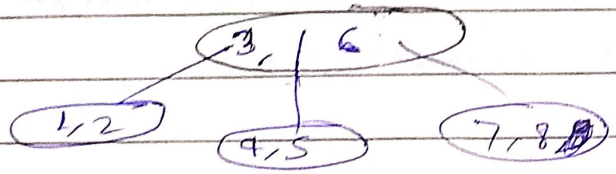
(a) $[\log_{\lceil m/2 \rceil} N]$

(b) $[\log_{(m/2)} N+1]$

(c) $[\log_{(m/2)-1} N]$

(d) none

(a) 2⇒ maximum hight of B-tree - with order = m
key = N



order = 4 = m
max key = 3 = N

(a) $\lceil \log_{\lceil m/2 \rceil} N \rceil$ = $\log_{\lceil 4/2 \rceil} 3 = \log_2 2 = 1$

03. Insertion of data into B-tree may cause

(a) height increase

(b) no change in height & no change in no. of nodes

(c) splitting the node

(d) all the above

(d) (3) Insertion in B-tree

(a) height increase

(b) no change in height & no change in no. of nodes

(c) spliting the node

(d) All

04. In B^+ trees deletion causes the height will _____.

I. no change

II. decreased by 1

III. decreased by 2

(a) I only

(b) II only

(c) I & II

(d) I, II & III

(c) 4⇒ In B^+-tree deletion causes the height will _____.

I) no change II) dec. by 1 III) dec. by 2

(c) I & II

05. An index record appears for every search key value in a file and a pointer to the record. This is example for

(a) sparse index

(b) dense index

(c) primary key index

(d) secondary key index

(b) 5=> An index record appears for every search key value in a file and a pointer to the record. This is example for -

(b) A dense index has an index entry for every search key value.

06. Assume that you have built a dense primary B+ tree index on a heap file containing 20,000 records. The key field for this B+ tree index is a 40 byte string and it is a candidate key. Pointers (i.e. record ids and page ids) are (at most) 10 byte values. The size of one disk page is 1,000 bytes . The index was built in a bottom-up fashion using the bulk-loading algorithm, and the nodes were filled up as much as possible. Then there are _____ number levels in the tree.

(a) 4 levels

(b) 5 levels

(c) 3 levels

(d) 6 levels

(a) 6-9

Size of index = 40 B (String) + 10 B (pointer)

= 50 B

Blocking factor = $\frac{1000}{50}$ = 20 records / block

No. of 1st level index record = 20,000

No. of 1st level block = $\left\lceil \frac{20000}{20} \right\rceil$ = 1000 blocks.

2nd level block = $\left\lceil \frac{1000}{20} \right\rceil$ = 50 blocks

3rd ——— block = $\left\lceil \frac{50}{20} \right\rceil$ = 3

4th ——— = $\left\lceil \frac{3}{20} \right\rceil$ = $\lceil 0.1 \rceil$ = 1

(a) 4 level

Teacher's Signature .....

07. Consider a disk with block size B = 512 bytes. A block pointer is P = 6 bytes long and a record P_R = 7 bytes long. A file has 30000 employee records of fixed length, each record size is fixed of 50 bytes. Find out the number of file blocks 'b' required, assuming an unspanned organization.

(a) 2800 blocks

(b) 2800 <b< 2900 blocks

(c) 3000 blocks

(d) None

(c) 7⇒ Block size = 512 B

Block pointer ~~(P)~~ = 6 B

P_R = 7 B

record size = 50 B

file = 30000 records.

blocks = ?

Blocking factor = $\left\lfloor \frac{512}{50} \right\rfloor$ = 10

No. of blocks = $\frac{30000}{10}$ = 30000 block

(c) 3000 blocks

08. Consider B tree where size of the node is 512 bytes.

Search key field = 9 bytes, block pointer = 6 bytes, data pointer = 7 bytes.

Find the order of the tree and number of data entries in level 2 if fill factor of the page is 0.69.

(a) 16, 3840

(b) 16, 61440

(c) 16, 15

(d) 16, 16

(a) Ans

Node - size = 512 B, Key = 9B, P_B = 6B

P_R = 7B, fan of page = 0.69, level = 2

order of tree = ?, number of data entries = ?

$$n P_B + (n-1)(K+P_R) \leq \text{Block size}$$

$$6n + (n-1) \times (9+7) \leq 512$$

$$6n + 16n - 16 \leq 512$$

$$n \leq 528/24$$

$$n \leq 22$$

69% full order = $22 \times 0.69 \approx 16$

No. of index record = 3840

Teacher's Signature .....

09. Pick up the ***correct*** statements of the following:

P: All conflict serializable schedules are also view serializable schedules.

Q: B+ tree will have more redundancy than B tree.

R: Weak entity is always having total participation.

S: R1(x), W2(y) are conflict operations.

T: Read and write locks are released only after commit in rigorous two phase locking protocol

U: ALL and ANY conditions will return TRUE if inner query returns an empty set.

(a) P, Q, R, T

(b) P, R, S, T, U

(c) P, R, T

(d) P, Q, T

(a) Q.8 Correct —

(P) All conflict serializable schedules are also view serializable schedules.

(Q) B^+- tree will have more redundancy than B-tree.

(R) Weak entity is always having total participation.

(T) Read & write locks are released only after commit in rigorous 2PL phase locking protocol.

(a) P, Q, R, T.

10. Given that the block size is 512 bytes, search field is 9 bytes, record pointer is 7 bytes and a block pointer is 6 bytes long, what are the orders of internal and leaf nodes of B^+-tree respectively?

(a) 31, 34

(b) 34, 31

(c) 32, 33

(d) 34, 32

(b) 10⇒ Block-Size = 512 B, search field = 9B

$P_R = 7B$, $P_B = 6B$,

order of internal & leaf node (B^+-tree)

Internal Node	Leaf Node
$n \times 6 + (n-1) \times 9 \leq 512$	$n \times (9+7) + 6 \leq 512$
$15n \leq 521$	$16n \leq 506$
$n = 34$	$n = 31$

Teacher's Signature .....

11. Consider a disk with block size = 1024 bytes, A block pointer 6 bytes and record pointer 8 bytes. The file has 10000 employee records. Every record is fixed length of 100 bytes. Its key is SSN which is of 2 bytes. The file is not ordered by the key field (SSN) and want to construct a secondary index on SSN. Find out the no. of blocks required in first level index.

(a) 100 blocks
(b) 99 blocks
(c) 117 blocks
(d) 116 blocks

(b) Ans ⇒ Block-size = 1024 B, P_B = 6B, P_R = 8B

Field = 10000 emp. record

record length = 100 B.

Key = 2B, block = ? (1st level)

index record size = 2 + 8 = 10B

Blocking factor = $\lfloor 1024/10 \rfloor = 102$

No. of 1st level blocks = $\lceil \frac{10000}{102} \rceil = 99$

block = 99

12. For the above file, if we make it into multilevel index, find out the no. of total blocks required.

(a) 102 blocks

(b) 100 blocks

(c) 119 blocks

(d) 118 blocks

(b) Ans ⇒ Total blocks = 99 (From Q1-(i))

2nd level = $\left\lceil \frac{99}{102} \right\rceil$ = 1

Total = 99 + 1 = 100

13. Consider the file consists of 30,000 fixed length records of size 100 bytes. Each disk block size is 1024 bytes and block pointer size is 6 bytes. Suppose we construct a secondary index on key field of length 9 bytes long then find out the no. of block needed for index file?

(a) 441 (b) 434

(c) 435 (d) 442

(d) 13⇒ Size of index = 9+8 = 15B

Blocking factor = $\frac{1024}{15}$ = 68

No. of blocks = $\frac{30000}{68}$ = 442

14. Which of the following is a false statement?

(a) Insertion of a search key in B^+-Tree may cause height to be increased one level.

(b) Deletion of a search key in B-Tree may cause height to be decreased by one level.

(c) Insertion of a search key into a B-Tree may or may not increase the height.

(d) Deletion of a search key will not decrease the height of a B^+-Tree as B^+ Tree allows duplicate values.

(d) Ans => False Statement —

(d). Deletion of a search key ~~the se~~ will not decrease the height of a B^+-tree & B^+ Tree allows duplicate values.

15. Suppose we have an ordered file with 30,000 records stored on a disk with block size 1024 bytes stored based on key field of length 9 bytes and Block pointer of 6 bytes. File records are of fixed length and are unsnapped, with record length 100 bytes. If we construct an index on the above key field, find out the total number of blocks required in index file?

(a) 44 blocks (b) 45 blocks

(c) 441 blocks (d) 442 blocks

(b) 15)

Blocking factor = 1024 / 100 = 10

No. of data blocks = 30000 / 10 = 3000

Size of index record = 9 + 6 = 15 B

Blocking factor = $\frac{1024}{15}$ = 68

1st level blocks = $\frac{3000}{68}$

= 45

16. Consider a block size 512 bytes, search key field 9 bytes long and block pointer size 6 bytes long. If a level 3 B^+ tree is constructed considering root as level 0, find out the maximum number of entries in the B^+ tree ?

(a) $(34)^2$ * 33 entries

(b) $(34)^3$ * 33 entries

(c) $(35)^2$ * entries

(d) $(35)^3$ * 34 entries

(b) 16 →

Block size = 512 B

Search key = 9 B , P_B = 6 B

level = 3 , root → level - 0

$$\text{order} = \left\lfloor \frac{512}{15} \right\rfloor = 34$$

No. of children - Leaf = $(34)^3$

Total entries = $(34)^3 * 33$

17. Consider a disk with block size 512 bytes. A block pointer 6 bytes long, record pointer 7 bytes long. A file has 30,000 employee

records of fixed length 100 bytes and with a key field 9 bytes long.

Suppose that the file is ordered by the key field SSN and we want to construct a multilevel index on SSN field. Find out the number of block accesses needed to search for and retrieve the record from file for given SSN value using the index?

(a) 3 blocks
(b) 4 blocks
(c) 5 blocks
(d) more than 5 blocks

(b) (7⇒)

Block-size = 512 B, P_B = 6 B, P_R = 7 B

file = 30000 emp. fixed length = 100 B

key field = 9 B, No. of blocks access = 5

level = 3

block access = 4

18. How many clustered indexes a table can have

(a) Any

(b) Can't say

(c) Depends on columns in the table

(d) None of the above

(d) 18⇒ How many clustered indexes a table can have —

⇒ Cluster index created on non-key and file is physically order on that non-key

(d) None

Statement for Linked answer 19 and 20 is given below.

Consider the following description:

To construct the B+ tree use the following values. Consider the database block size as 512 bytes, block pointer as 6 bytes, record pointer as 7 bytes and search key value as 9 bytes.

19. What is the order of the B+ tree internal node.

(a) 23 (b) 32 (c) 34 (d) 44

20. What will be the number of data entries possible in level 2 of the B+ tree, if fill factor of the node is 0.8

(a) 17, 954 (b) 16, 954

(c) 18, 954 (d) 17, 496

Common 19×20

Block size $\Rightarrow$ 512

$P_R = 6$, $P_R = 7$, $K = 9$

(c) 19 $\Rightarrow$ order of B^+-tree (internal node)

$$6n + (n-1) \times 9 \leq 512$$

$$n = 39$$

(d) 20 $\Rightarrow$ in level-2, No. of data entry (0-8)

order of leaf = 31

of 80% filled order of leaf = $31 \times 0.8 = 24$

Internal = 27

(d) Total = $(27)^2 \times 24 = 17,496$

21. What is the minimum space utilization for a B+ tree index with the exception of the root?

(a) 0 %
(b) 100%
(c) 50 %
(d) None of these

(c) 2|3 minimum space utilization for B^+-tree index with exception of root ?

(c) Each node in B^+-tree to be filled minimum 50%

22. What is the difference between a primary index and a secondary index?

(a) There can be only one secondary index while there can be many primary indices

(b) There can be many secondary indices while there can be many primary indices

(c) Primary is dense index, but secondary is sparse index

(d) There can be only one primary index while there can be many secondary Indices

Q2 ⇒ Primary index & Secondary index (Difference)

(1) There can be only one primary index while there can be many secondary

23. Select * from student where marks > 50 and marks < 80, pick the correct index to have optimized results.

(a) Clustered B^+ tree index is best suited

(b) Non Clustered B^+ tree index is best suited

(c) Both are having equal impact

(d) Both are not useful

(q) 23⇒ SELECT * FROM students where marks > 50 and mark < 80

(a) clustered B^+ - tree is best suited.

24. Database consists 30,000 records and each record length is 100 bytes. Consider block size =1024 bytes. Find the number of searches required without primary index if search key field value = 9 bytes and block pointer = 6 bytes.

(a) $\log_2 30000$ (b) $\log_2 3000$

(c) $\log_2 4$ (d) $\log_2 3045$

(b) Q4 =>

No. of data blocks = 3000

Binary search used & block access

$= \log_2 3000$

b