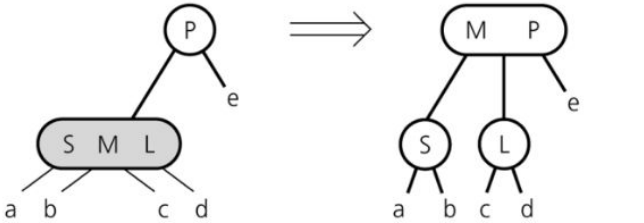
NOC24-CS75
Data Base Management System

Tutorial by: Adwait P. Parsodkar
Week - 6

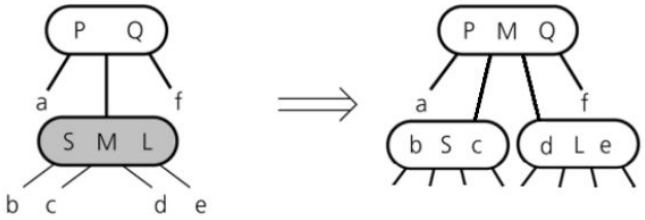
Which of the splitting rule is incorrect in respect of 2-3-4 Trees?

Question - 1

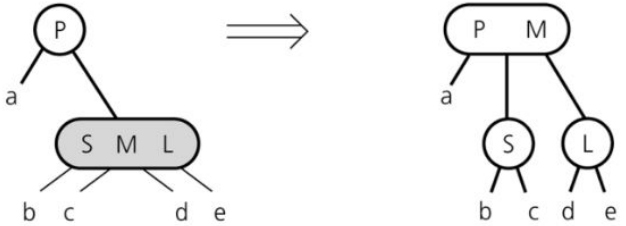
a)



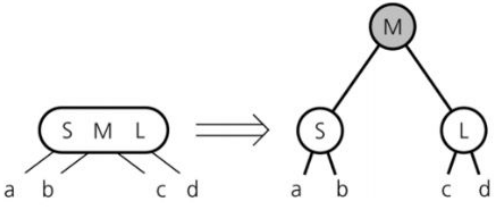
b)



c)



d)



Solution - 1

Solution: (b)

Explanation:

Refer to Module 27. Splittings corresponding to options (a), (c) and (d) are correct.

The labels “b”, “c”, “d”, and “e” are labeled for links to nodes. In option (b), links are inserted as values. So, (b) is incorrect.

Question - 2

Consider a file of 48000 records. Each record is 40 bytes long and its key field is of size 4 bytes. Size of the memory is 512 kilobytes. The disk block size is 512 bytes, and the size of a block address is 8 bytes. If we want to order the file on key field, which indexing technique gives better result for accessing a record?

a) Primary indexing

b) Secondary indexing

c) Clustering indexing

d) Multilevel indexing

Solution - 2

Solution: (d)

Explanation: [Reference: Module 26]

In this problem, (Key + block address) =12 bytes.

So, one block can store only = 512 / 12 = 42 (Key + block address) pair.

Size of all pairs (Key + block address) = 12 * 48000 = 576 KB

Hence, primary index does not fit in one memory block, and access becomes expensive.

As we want to order on key field only, secondary index is expensive and clustering indexing is not applicable.

Hence, Multilevel indexing is the best choice for this problem.

Question - 3

Identify the correct properties of a B+ tree of order n.

a) A node that is a root but not a leaf, has at least n/2 children.

b) A leaf node has between 1 and (n − 1) values.

c) A node that is a root and leaf, can have between 0 and (n − 1) values.

d) All paths from root to leaf are of the same length.

Solution - 3

Solution: (c), (d)

Explanation: [Reference: Module 28 slide 8]

Properties:

- All paths from root to leaf are of the same length
- Each node that is not a root or a leaf has between $\lceil n/2 \rceil$ and n children.
- A leaf node has between $\lceil (n-1)/2 \rceil$ and $n-1$ values.
- Special cases:
  - If the root is not a leaf, it has at least 2 children.
  - If the root is a leaf (that is, there are no other nodes in the tree), it can have between 0 and $(n-1)$ values.

Question - 4

A hash table of length 8 uses open addressing with hash function h(x) = x mod 6, and linear probing. The content of the table after inserting the six key values is shown. What will be the correct order of insertion of keys?

a) 22, 26, 19, 37, 45, 29

b) 26, 22, 37, 19, 45, 29

c) 45, 26, 22, 37, 19, 29

d) 29, 26, 22, 45, 37, 19

0	
1	37
2	26
3	19
4	22
5	45
6	29
7	

Solution - 4

Solution: (b)

Explanation:

In open hashing, the set of buckets is fixed, and there are no overflow chains. In linear probing, if a bucket is full, the system inserts record in the next free bucket.

X mod 6 = 2 when X=26. Bucket will be

		26					

X mod 6 = 4 when X=22. Bucket will be

		26		22			

X mod 6 = 1 when X=37. Bucket will be

	37	26		22			

X mod 6 = 1 when X=19, collision occurs at index position 1. As next available index position is 3. Bucket will be

	37	26	19	22			

X mod 6 = 3 when X=45, but the bucket is allocated, so the record will be inserted to the next free bucket. Allocation of the bucket will be

	37	26	19	22	45		

X mod 6 = 5 when X=29, but the bucket is allocated, so the record will be inserted to the next free bucket. Bucket will be

	37	26	19	22	45	29	

So, option b) is correct.

Question - 5

Suppose that one block in a disk can store either 6 records or 15 key pointers. If a database contains 1200 records, how many total number of blocks do we need to store the data file and the index file?

a) 280

b) 260

c) 200

d) 90

Solution - 5

Solution: a)

Explanation:

Number of blocks needed to store data file with n records = 1200/6 = 200

Number of blocks needed to store dense record index = 1200/15= 80

Total blocks required = 200 + 80 = 280 blocks.

So, option (a) is correct.

Question - 6

In a file system records are arranged with hashing using the hash function H(k) and a hash table of size n = 100. If the key k = 135642 is placed in location 21 of the hash table. Assume that there was no collision. Which of the following function is used as hash function H(k)?

$$H(135642) = 21$$

a) Round(key / 8) % 100

b) (Sum of digits of key) % 70

c) (Product of digits of key) % 50

d) (Key * 11) %35

Solution - 6

Solution: (b)

Explanation:

(Sum of digits of key) % 70 of Key k = 135642 will generate 21.

There are five records in a `CarPolicy` table.

Question - 7

PolicyId	HolderName	Age	Gender
121	Rama	45	M
212	Abdul	39	M
183	Jennifer	28	F
295	Maya	39	F
117	Dev	28	M

Identify the correct SQL query to create composite index on `Age` and `Gender`?

a)
```
CREATE composite INDEX cp_age_gen
ON CarPolicy (Gender, Age);
```

b)
```
CREATE INDEX cp_gen_age
ON TABLE CarPolicy (Age, Gender);
```

c)
```
CREATE INDEX cp_age_gen
ON CarPolicy (Gender, Age);
```

d)
```
CREATE INDEXES cp_age_gen
ON CarPolicy (Gender, Age);
```

Solution - 7

Solution: (c)

Explanation:

The general syntax for creating multi-column index is :

CREATE INDEX index_name

ON TABLE_NAME (COLUMN_NAME1, COLUMN_NAME2,.. COLUMN_NAMEN);

Hence, option (c) is correct.

Question - 8

Suppose, a system uses B+ tree indexing for storing its records. If the minimum size of one block is 780 bytes, the size of one key is 8 bytes, record pointer is 4 bytes, and one block pointer is 12 bytes long, what will be the order of a leaf node (maximum possible number of key value with record pointer pairs) ?

a) 65

b) 64

c) 97

d) 195

Solution - 8

Solution: b)

Explanation:

Record pointer size, r = 4 bytes

Key value size, V = 8 bytes

Disk block ptr, p = 12 bytes

One block size = 780 bytes

Order of leaf node= n

A leaf node in B+ tree contains at most n key values, at most n record pointers and one block pointer.

As one block size = n*V + n*r + p

Order of leaf node,
n = (one block size - p)/(V+r)
= (780 - 12)/(8+4) = 64

Question - 9

Consider the `HealthPolicy` Table.

HP_ID	NAME	JOB
1369	SMITH	ENGINEER
1499	ALLEN	HR
2521	WARD	HR
2566	JONES	MANAGER
2654	MARTIN	HR
3698	BLAKE	MANAGER
3788	SCOTT	ANALYST
4844	TURNER	HR
5876	ADAMS	ENGINEER
5900	JAMES	ENGINEER
6902	FORD	ANALYST
7934	MILLER	ENGINEER

Consider the bitmap index for the Job field, with J1, J2, J3, and J4 representing different jobs. If the bitmap index value of J1, J2 and J3 are as follows:

J1	0	0	0	0	0	0	1	0	0	0	1	0
J2	0	1	1	0	1	0	0	1	0	0	0	0
J3	1	0	0	0	0	0	0	0	1	1	0	1

Identify the correct option that must be represented by J4.

a) Analyst

b) HR

c) Engineer

d) Manager

Solution - 9

Solution: (d)

Explanation:

If we put true (1) and false (0) for the given value of the Job field we get J1 for Analyst, J2 for HR, and J3 for Engineer.

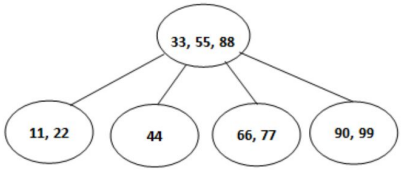
So, (d) is the correct option.

Question - 10

Consider the following 2-3-4 tree:

What is the result of inserting 90 in the above tree?

a)



b)

c)

d)

Solution - 10

Solution: (c)

Explanation:

When 90 will be inserted, the node containing (77, 88, 99) will be split. So, 88 will go to the root. Number of children at the root will increase.

Hence the tree in (b) will be formed.

Question - 11

Choose the incorrect option(s) about the indexing.

a) There can be many clustering indices.

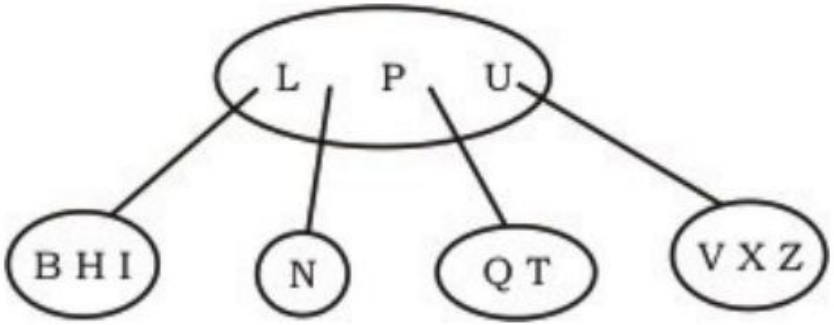
b) There can be at most one secondary index.

Solution - 11

Answer: a), b)

Question - 12

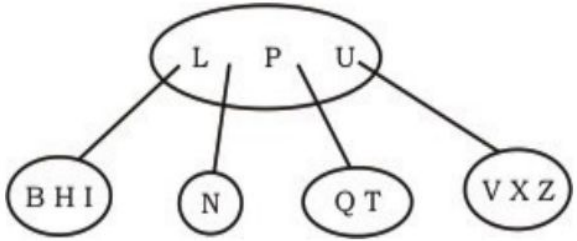
Consider the following 2-3-4 tree, in which each data item is inserted in the alphabetical order of letters.

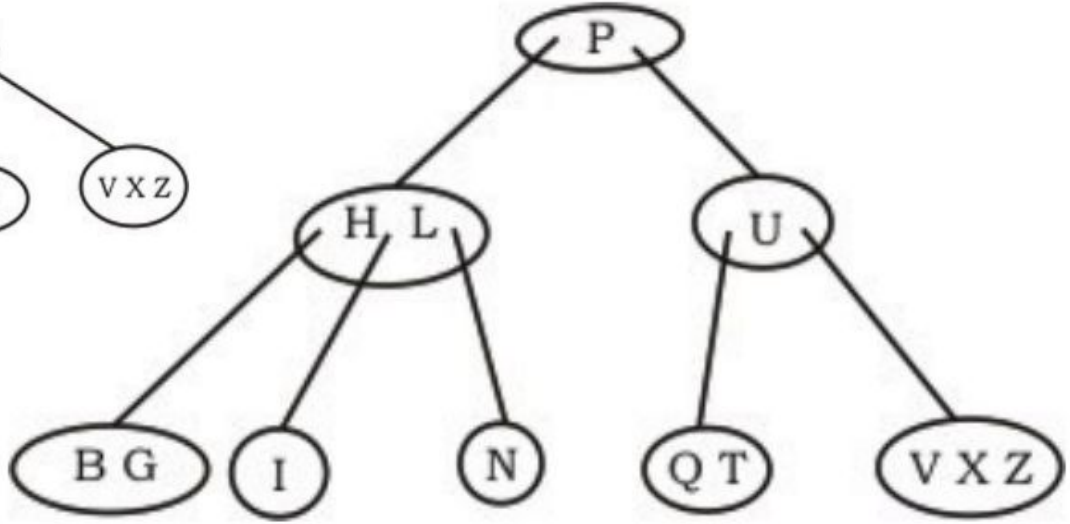


If we insert G into the above tree, what will be the total number of nodes in the resultant tree.

Solution - 12

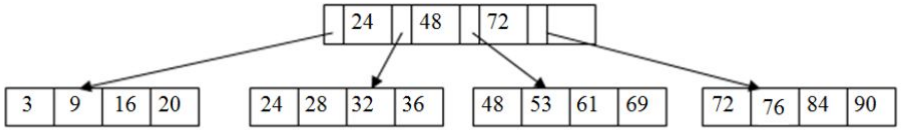
Explanation: The resulting tree after inserting G is shown below. It consists of 8 nodes.





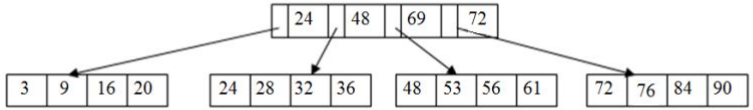
Question - 13

Consider the following B+ tree:

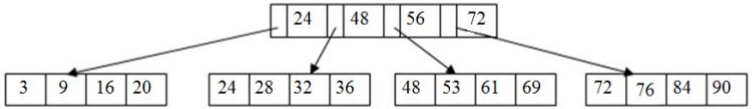


What will be the correct B+ tree after the insertion of key 56?

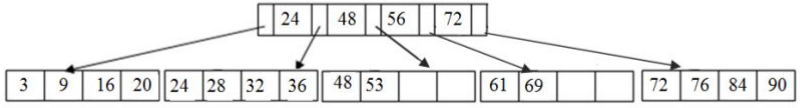
a)



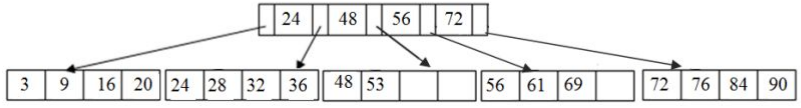
b)



c)



d)

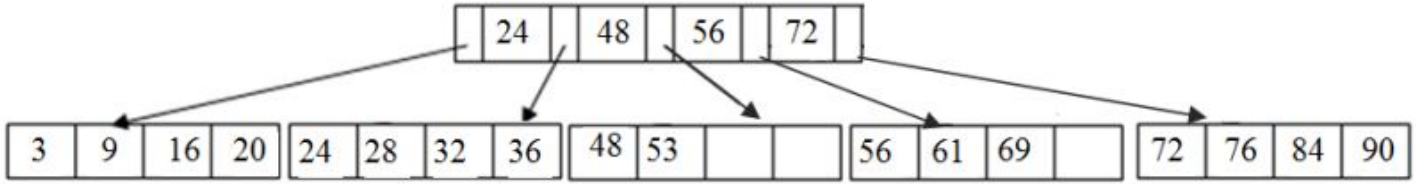


Solution - 13

Answer: d)

Explanation: It will go to 3rd leaf node after 53. As B+ tree is a balanced tree and that leaf node is already full, we cannot insert the record there directly.

The new leaf node should have values (48, 53, 56, 61, 69) and its current root value is 48. We need to split the leaf node from the middle. So, we have to group (48, 53) and (56, 61, 69) in 2 leaf nodes. If these two have to be leaf nodes, the intermediary node cannot branch from 48. It should have 56 added to it and then we can have pointers to the new leaf node. And the final tree will be:



Question - 14

Consider a file of 50000 records. Each record is 40 bytes long and its key field is of size 4 bytes. The size of the memory is 512 KB. The disk block size is 512 bytes, and the size of a block address is 8 bytes. If we want to order the file on key field, which indexing technique gives better results for accessing a record?

a) Primary indexing

b) Multilevel indexing

c) Secondary indexing

d) Clustering indexing

Solution - 14

Answer: b)
Explanation: In this problem, (Key + block address) =12 bytes.
So, one block can store only = 512 / 12 = 42 (Key + block address) pair.
Size of all pairs (Key + block address) = 12 * 50000 = **600 KB**
Hence, primary index does not fit in memory, and access becomes expensive.
As we want to order on key field only, secondary index is expensive and clustering indexing is not applicable.
Hence, Multilevel indexing is the best choice for this problem.
Refer to slide 26.

Question - 15

Consider a hash table with 8 slots. The hash function is $h(X) = X \bmod 8$. The collisions are resolved by chaining. Find out the maximum, minimum, and average chain lengths in the hash table. If the keys are inserted in the following order: 3, 18, 29, 25, 30, 43, 52, 67, 70.

a) 3, 0, and 1

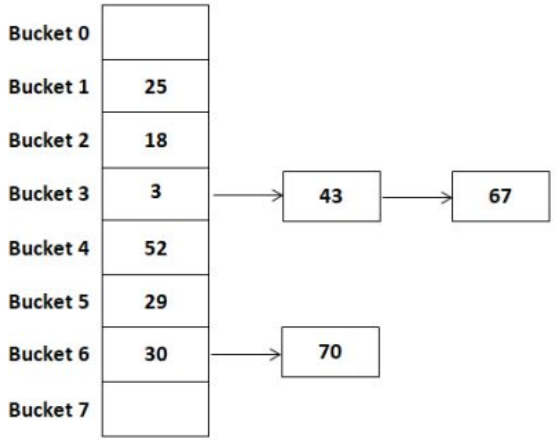
b) 3, 1, and 2

c) 4, 0, and 1

d) 4, 0, and 2

Solution - 15

Answer: a)
Explanation: After inserting the keys, the hash table looks like the following.



Hence, the maximum chain length = 3
The minimum chain length = 0
The average chain length = 1
Hence, option a) is the answer.

Question - 16

In a B^+ tree, size of a node is generally the same as that of a disk block. Suppose that the size of a block is 2 kilobytes. One index entry is 32 bytes long. What will be the height of the tree of a file with 1 million search key values?

a) 6

b) 5

c) 4

d) 3

Solution - 16

Answer: c)

Explanation: The height of the tree $= \lceil log_{\lceil n/2 \rceil}(K) \rceil$

n = 2 KB / 32 bytes = 64

K = 1000000

The height of the tree $= \lceil log_{\lceil 32 \rceil}(1000000) \rceil \approx 4$

Question - 17

There are five records in a database table.

Sid	Sname	Course	Teacher
S1	RAM	JAVA	AR
S2	MADHAB	DBMS	PPD
S4	Maya	JAVA	PB
S2	MADHAB	DBMS	SM
S3	Jennifer	PYTHON	SM

Which column is represented by the following bitmap index?

1	0	1	0	0
0	1	0	1	0
0	0	0	0	1

a) Sid

b) Sname

c) Course

d) Teacher

Solution - 17

Answer: c)

Explanation: If we create a bitmap indexing on **Course** column, we get the following:

1	0	1	0	0
0	1	0	1	0
0	0	0	0	1

Hence, option c) is the answer.

Question - 18

There are five records in a `Student` table.

Sid	Sname	Course	Teacher
S1	RAM	JAVA	AR
S2	MADHAB	DBMS	PPD
S4	Maya	JAVA	PB
S2	MADHAB	DBMS	SM
S3	Jennifer	PYTHON	SM

Identify the correct SQL query to create a bitmap index on `Teacher` attribute for the table `Student`.

a)
```
CREATE BITMAP INDEX bm_teacher
ON Student(Teacher);
```

b)
```
CREATE BITMAP INDEX bm_teacher
ON Teacher(Student);
```

c)
```
CREATE BITMAP bm_teacher
ON Teacher(Student);
```

d)
```
CREATE INDEX(BITMAP) bm_teacher
ON Student(Teacher);
```

Solution - 18

Answer: a)

Explanation: The general syntax for creating BITMAP index is :

```
CREATE BITMAP INDEX <index-name> on <relation-name> (<attribute-list>)
```

Hence, option (a) is correct.

Question - 19

Suppose that one block in a disk can store either 25 records or 40 key pointers. If a database contains 5000 records, how many blocks do we need to store the data file and the dense index?

a) 400

b) 325

c) 200

d) 125

Solution - 19

Answer: b)

Explanation:

Number of blocks needed to store data file with n records $= 5000/25 = 200$

Number of blocks needed to store dense file index $= 5000/40 = 125$

Total blocks required $= 200 + 125 = 325$ blocks.

So, option (b) is correct.

Question - 20

In a file system records are arranged with hashing using the hash function:
$H(x) = Round(x \div 7)\%100$, where x is the key. In which location, `key x= 42651` will be placed?

a) 93

b) 60

c) 51

d) 42

Solution - 20

Answer: a)

Explanation: The hash function is $H(x) = Round(x \div 7)\ \%\ 100$

`key k = 42651`

So, Location $= Round(42651 \div 7)\ \%\ 100$

$= Round(6093)\ \%\ 100$

$= 6093\ \%\ 100$

$= 93$

Hence, option a) is correct.