Database Management System: Assignment 6

Total Marks : 20

July 20, 2022

Question 1

Choose the incorrect option(s) about indexing from the following.

Marks: 2 **MSQ**

a) There can be many clustering indices in a database file.

b) There can be at most one secondary index in a database file.

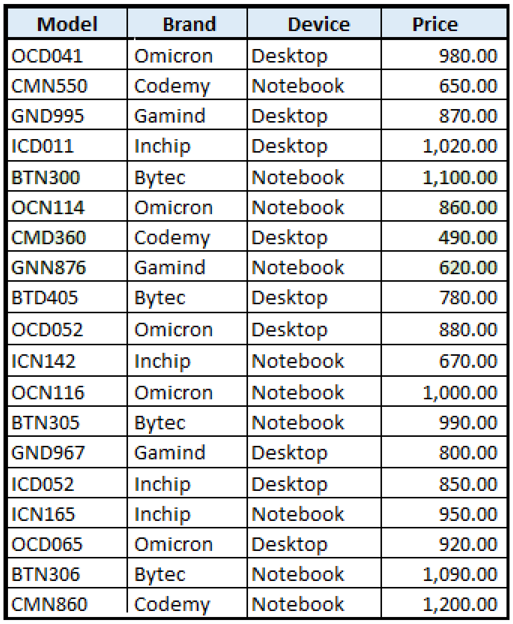
c) It is preferable to use a dense index when the file is not sorted on the indexed field.

d) Dense index is preferred over the sparse index when the size of a file is small.

Answer: a), b)
Explanation: It is preferable to use a dense index instead of a sparse index when the file is not sorted on the indexed field or when the index file is small compared to the size of memory.
There can be only one clustering index while there can be many secondary indices.
Hence, options a) and b) are the answer.

Question 2

Consider the following instance of a `Gadgets` table:



Model	Brand	Device	Price
OCD041	Omicron	Desktop	980.00
CMN550	Codemy	Notebook	650.00
GND995	Gamind	Desktop	870.00
ICD011	Inchip	Desktop	1,020.00
BTN300	Bytec	Notebook	1,100.00
OCN114	Omicron	Notebook	860.00
CMD360	Codemy	Desktop	490.00
GNN876	Gamind	Notebook	620.00
BTD405	Bytec	Desktop	780.00
OCD052	Omicron	Desktop	880.00
ICN142	Inchip	Notebook	670.00
OCN116	Omicron	Notebook	1,000.00
BTN305	Bytec	Notebook	990.00
GND967	Gamind	Desktop	800.00
ICD052	Inchip	Desktop	850.00
ICN165	Inchip	Notebook	950.00
OCD065	Omicron	Desktop	920.00
BTN306	Bytec	Notebook	1,090.00
CMN860	Codemy	Notebook	1,200.00

If we want to create an index on `Model` attributes, which of the following indexing can be selected? *Marks: 2* **MCQ**

a) Primary and Dense indexing.

b) Secondary and Dense indexing.

c) Secondary and Sparse indexing.

d) Primary and Sparse indexing.

Answer: b)
Explanation: **Primary index:** in a sequentially ordered file, the index whose search key specifies the sequential order of the file.
Secondary index: an index whose search key specifies an order different from the sequential order of the file.
Dense index: the index record appears for every search-key value in the file.
Sparse Index: contains index records for only some search-key values.
The values given in attribute `Model`, are not sorted, hence it is not possible to build primary and sparse indexing.
So, only Secondary and Dense indexing can be possible. Hence, option b) is correct.

Question 3

Suppose that one memory block in a disk can store either 32 key pointers or 12 whole records. If a database file contains 1056 records, how many memory blocks do we need to store the data file and the dense index? *Marks: 2* **MCQ**

a) 22

b) 33

c) 88

d) 121

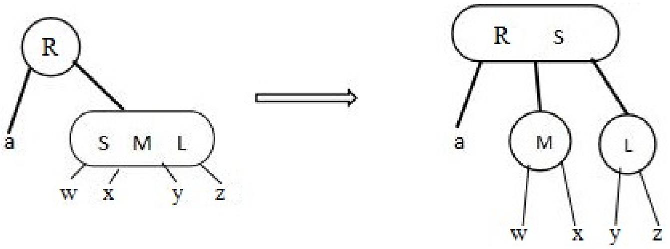
Answer: d)
Explanation: Number of blocks needed to store data file with 1056 records = 1056/12 = 88
Number of blocks needed to store dense index file = 1056/32= 33
Total blocks required = 88 + 33 = 121 blocks. So, option (d) is correct.

Question 4

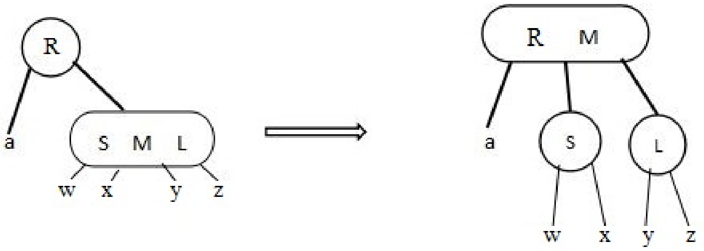
Which of the following cell splitting strategies are possible in a 2-3-4 tree?

Marks: 2 **MCQ**

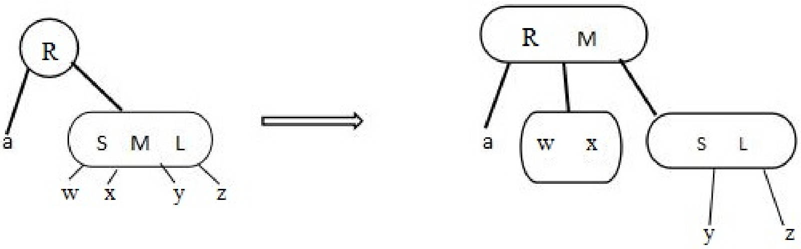
a)



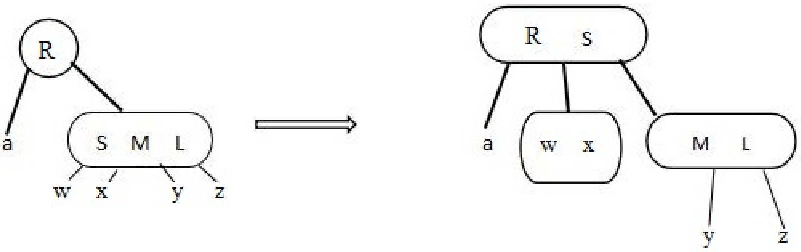
b)



c)



d)



Answer: b)

Explanation: A `2-3-4` tree must contain 2 data and 3 links. As per the rules of splitting, if an extra node appears at the leaf (beyond node's capacity), then the leaf node splits in a way where the data item in the middle moves up to the immediate parent node. The data item which was at its immediate left becomes its left child(when it becomes a parent node) and the data item which was at its right would become its right child. This process continues unless the tree becomes stable.

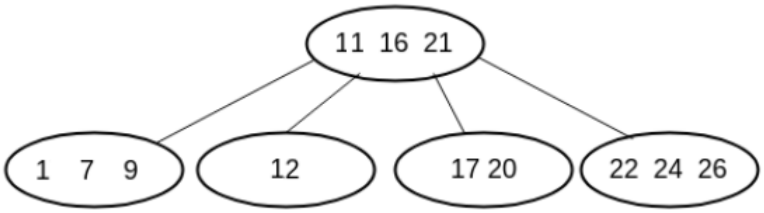
In option a) before splitting, `M` was to the right of `S` and `x` also was to the right pointer of `S`. But after splitting `M` and `x` both are on the left of `S`, which is incorrect.

Option c) and option d) are incorrect because `w` and `x` were not the direct children of `R` before splitting.

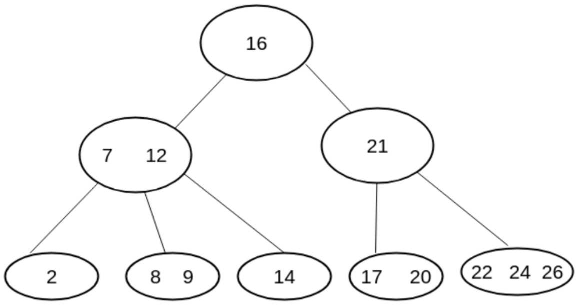
Option b) is correct as the relationship between parent-child node is preserved.

Question 5

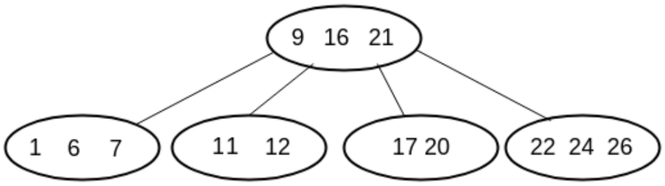
Consider the following 2-3-4 tree in which each data item is an integer.



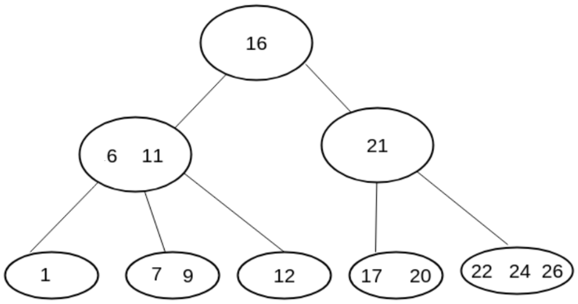
What will be the result of inserting 6 in the above tree? Marks: 2 **MCQ**



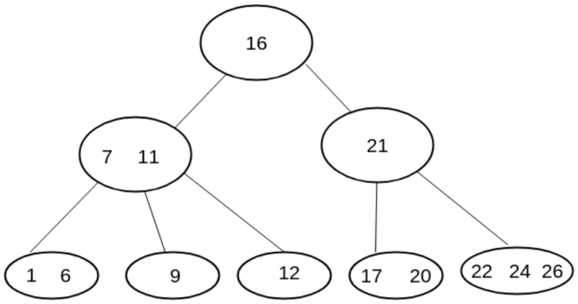
a)



b)



c)



d)

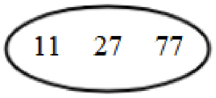
Answer: d)
Explanation: In 2-3-4 tree insertion, we start from root and traverse till the leaf node where key is to be inserted. While traversing, if we find a node which is full, we split it. When we insert 6, we find root itself is full, so we split it. When we come down to leftmost leaf, we find that the leaf is also full, so we split the leaf also.

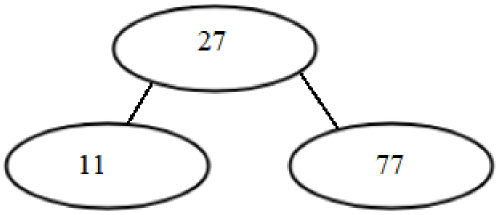
Refer Module 27 slide 9-26

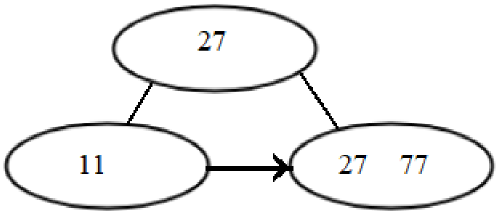
Question 6

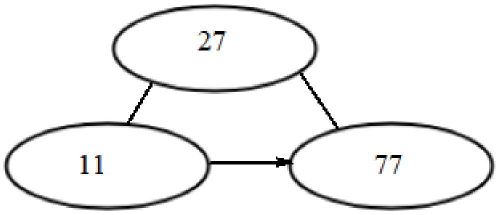
Identify the correct B^+-tree of order 3, generated by inserting 27, 11 and 77.

Marks: 2 **MCQ**

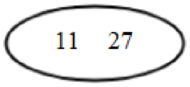
a) 

b) 

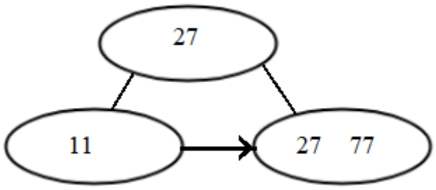
c) 

d) 

Answer: c)
Explanation: In a B^+-tree of order 3, every non-root node can have maximum of 2 keys. After insertion of 27, and 11 the tree will be:



After insertion of 77, the node will split and the new tree will be:



Question 7

In an open hashing system, collision is resolved using linear probing. In the given system, the hash table is of size eight starting with index `zero`. The hash function is:`(3x + 2) mod 7` where `x` is the value of the `key`. Assuming that the hash table is initially empty, which of the following depicts the contents of the hash table when a sequence `10, 23, 37, 41, 45, 53` is inserted into the table?

Marks: 2 **MCQ**

a) 37, 23, _, 53, 10, 45, 41,_

b) 53, 23, 37, _, 10, _, 41, 45

c) 53, 23, 37, _, 10, 45, 41,_

d) _, 53, 23, 37, 10, 45, 41,_

Answer: c)
Explanation: In open hashing, the set of buckets is fixed, and there are no overflow chains. If a bucket is full, the system inserts records in some other bucket in the initial set of buckets. One policy is to use the next bucket (in cyclic order) that has space; this policy is called linear probing.

(3x+2) mod 7 =4 when x=10
_ , _, _, _, 10, _,_, _

(3x+2) mod 7 =1 when x=23
_ , 23, _, _, 10, _,_, _

(3x+2) mod 7 =1 when x=37 → collision occurs at index position 1. As next available index position is 2, hence it will be occupied
_ , 23, 37, _, 10, _,_, _

(3x+2) mod 7 =6 when x=41
_ , 23, 37, _, 10, _,41, _

(3x+2) mod 7 =4 when x=45 → collision occurs at index position 4. As next available index position is 5, hence it will be occupied
_ , 23, 37, _, 10, 45, 41, _

(3x+2) mod 7 =0 when x=53
53 , 23, 37, _, 10, 45, 41, _

Question 8

Consider the table `Flower`:

FID	NAME	SEASON
131	Rose	Summer
149	Jasmine	Rainy
252	Lotus	Autumn
256	Daliya	Winter
265	Orchid	Summer
369	Jiniya	Winter
378	Lily	Rainy

Consider the bitmap index for the `SEASON` field, with S1, S2, S3, and S4 representing different `SEASON`. Select the correct option for which the bitmap index value of S1, S2 will be as follows:

Marks: 2 **MCQ**

S1	0	1	0	0	0	0	1

S2	0	0	0	1	0	1	0

a) Summer and Winter

b) Summer and Autumn

c) Rainy and Winter

d) Rainy and Autumn

Answer: c)
Explanation: if we put true (1) and false (0) for the given value of the `SEASON` field we get for Rainy –S1, and Winter – S2.
So, option (c) is correct.

Question 9

If the index entries are inserted in sorted order into a `B+ tree`, choose the correct option(s) from the following statements. *Marks: 2* **MCQ**

a) The new entries get directed to last leaf node if the index entries are inserted in ascending order.

b) When last leaf node gets filled, it will split into three.

c) If the leaf node splits when inserting in ascending order, the right node is left untouched and the insertions take place on the parent node.

d) Except for the first and last leaf, the occupancy of nodes is about 100 percent if insertion happens in descending order.

Answer: a)
Explanation: If the index entries are inserted in ascending order, the new entries get directed to the last leaf node.
When this leaf node gets filled, it is split into two.
This makes the occupancy of the leaf nodes about 50 percent, except for the last leaf.
If keys that are inserted are sorted in descending order, the above situation would still occur, but symmetrically, with the right node of a split never getting touched again, and occupancy would again be 50 percent for all nodes other than the first leaf.

Hence, only option a) is correct.

Question 10

How many maximum numbers of keys, a `B-tree` of order 4 can store? The height of the tree is 5 and the root node is assumed to be in height zero.

Marks: 2 **MCQ**

a) 8191

b) 4095

c) 2047

d) 1023

Answer: b)
Explanation: A `B-tree` of order `m` and height `h` will have the maximum number of keys when all nodes are completely filled.
A height-0, `B-tree` of order `4` can store 3 keys.
A height-1, `B-tree` of order `4` can have 4 children. So, can have 4x3 + 3 keys.
Similarly, maximum number of keys in a B-tree of height h = $(m^{h+1}–1)$ keys
Here, m=4 and h=5. Hence, maximum keys = $(4^{5+1}–1)$
So, $4^6 - 1$ = 4096 - 1 = 4095
Hence, option b) is correct.