MEGA MOCK
CHALLENGE

gradeup

रा.इ.सू.प्रौ.सं
NIELIT

NIELIT

Scientist ‘B'
Technical Assistant 'A'

Computer Science Engineering

Mega Mock Challenge
(May 09- May 10 2020)

Questions & Solutions

1. x – = y+1; does the same as:

A. x= x –y +1 B. x= – x –y – 1

C. x= –x + y +1 D. x= x – y – 1

Ans. D

Sol. a-=b;→ a=a-b

x - = y+1; → x = x – (y+1) → x = x – y - 1

2. How many times Gradeup is printed?

```
#include<stdio.h>
int main()
{
int i = -5;
while (i <= 5)
{
if (i >= 0)
break;
else
{
i++;
continue;
}
printf("Gradeup");
}
return 0;
}
```

A. 10 times B. 5 times

C. Infinite times D. 0 times

Ans. D

Sol. The loop keeps incrementing i while it is smaller than 0. When i becomes 0, the loop breaks. So Gradeup is not printed at all.

3. What will be the output of following code-

```
#include <stdio.h>
int main()
{
int x = 3;
if (x == 2); x = 0;
if (x == 3) x++;
else x += 2;
printf("x = %d", x);
return 0;
}
```

A. x = 4 B. x = 2

C. Compiler Error D. x = 0

Ans. B

Sol. Value of x would be 2.

Note the semicolon after first if statement. x becomes 0 after the first if statement. So control goes to else part of second if else statement.

4. Which of the following data structures is best suited for heap sort?

A. Array B. Singly linked list

C. doubly linked list D. double ended queue

Ans. A

Sol. Since a binary heap is an almost complete binary tree, it can be easily represented as an array and array based representation is space efficient. If the parent node is stored at index i, the left child can be found at index 2i + 1 and right child at index 2i + 2.

5. Which of the following is/are True?

S1: Heap sort is an inplace sorting technique

S2: Heap sort is stable.

A. Only S1 B. Only S2

C. Both S1 & S2 D. None of these

Ans. A

Sol. • **heap sort** is considered to be an **in-place** algorithm because **no** extra memory is used to perform the **sort**

• **Heapsort** is **not stable** because operations on the **heap** can change the relative order of equal items. From here: When **sorting** (in ascending order) **heapsort** first peaks the largest element and put it in the last of the list.

6. Total running time of Huffman coding is (Hint: Consider using Minheap)

A. $O(n^2)$ B. $O(n \log n)$

C. $O(n^2 \log n)$ D. $O(n^3)$

Ans. B

Sol. To analyze the running time of Huffman‛s algorithm,

```
Huffman (c) Algorithm
n = |c| //No. of characters in the alphabet
2) Q = c // Build a Min-heap
3) For i =1 to n − 1
```

```
4) allocate a new node z
5) z.left = x = Extract – Min(Q)
6) z.right = y = Extract – Min(Q)
7) z.freq = x.freq + y.freq
8) Insert (Q, z)
9) return Extract min(Q) // Return the root of the tree. 5
```

In the above algorithm, line 2 takes O(n) time to build Minheap. The for loop execute exactly n – 1 times during which Extract – Min(Q) takes O(log n) time. Thus the for loop contributes O(n log n) to the running time.

$\therefore O(n) + O(n \log n) \simeq O(n \log n)$

7. Which of the following is not an application of Greedy approach?

(i) 0/1 Knapsack

(ii) Fractional Knapsack

(iii) All pair shortest path

(iv) Single Source shortest path

(v) Optimal Merge Patterns

A. Only (ii), (iv) & (v)

B. Only (ii), (iii) & (v)

C. Only (i) & (iii)

D. Only (i), (iii) & (iv)

Ans. C

Sol. 0/1 Knapsack and All pair shortest path uses dynamic programming approach.

8. Number of spanning trees for K_4 is:

A. 4

B. 8

C. 16

D. 32

Ans. C

Sol. No. of spanning trees in an n-vertex complete graph is n^{n-2}

$\therefore 4^{4-2} = 16$

9. Which of the following is True about Spanning Tree?

(i) A spanning tree is a connected graph.

(ii) Number of edges in a spanning tree with n-vertices is n – 1.

(iii) Spanning tree contains cycles.

A. only (i) & (ii)

B. only (i) & (iii)

C. only (ii) & (iii)

D. All are True

Ans. A

Sol. A spanning tree is always connected. True

Number of edges in a spanning tree with n-vertices is n – 1. True

Spanning tree contains cycles. False

10. In a simple undirected graph with 'n' vertices maximum degree for each vertex is

A. n
B. n – 1
C. n (n – 1)
D. Infinite

Ans. B

Sol. In a simple undirected graph, self- loops and parallel edges are allowed. So, for each vertex, there can be a maximum of n – 1 edges forming a complete graph.

11. What is the minimum of the multiplications required to multiply the following three matrices.

A_{3x7} , B_{7x2} and C_{2x9}

A. 96
B. 97
C. 98
D. 378

Ans. A

Sol. To multiply AB we need $3 \times 7 \times 2 = 42$ multiplications.

$(AB)_{3x2} \cdot C_{2x9}$

Further multiplication with C requires $3 \times 2 \times 9 = 54$ Multiplications

$\therefore$ Total $54 + 42 = 96$ multiplications for (AB)C.

To multiply BC we need $7 \times 2 \times 9 = 126$ multiplications.

$A_{3x7}(BC)_{7x9}$

Further multiplications with A requires $3 \times 7 \times 9 = 189$ Multiplications

$\therefore$ Total $126 + 189 = 315$ multiplications for A(BC)

$\therefore$ Minimum multiplications required are 96.

12. Which of the following statements is/are true?

S1: Heap sort is an inplace sorting algorithm.

S2: Quick sort is a stable algorithm.

S3: Merge sort is a stable algorithm.

S4: Insertion sort is an inplace algorithm.

A. Only S2
B. Only S1 and S2
C. Only S1, S3 and S4
D. Only S2 and S4

Ans. C

Sol. Quick sort is not stable algorithm i.e. if the elements are repeated in the array, their order may or may not be preserved after the application of quick sort algorithm. Hence S2 is false.

13. A graph G is such that every vertex is connected to every other vertex via direct link. We need to find shortest path between two vertices using Bellman ford algorithm. How much time does it take?

A. $O(V^2 \log V)$
B. $O(V^2)$
C. $O(V^3)$
D. $(V+E) \log V$

Ans. C

Sol. In Bellman ford algorithm for every round 'E' edges are relaxed. Each edge relaxation takes constant time. In general there are (V – 1) rounds excluding the verification round.

Total time = (V – 1) × O(E) × O(1) = O(VE)

But for complete graph E = O(V^2)

∴ Total time = O(V^3)

14. How many passes of bubble sort are required to sort the following sequence (Pass is counted only when at least one swap is performed in the bubble sort pass)?

12, 15, 17, 11, 14, 13

A. 6

B. 5

C. 4

D. 3

Ans. C

Sol. Original sequence: 12, 15, 17, 11, 14, 13

After 1st pass: 12, 15, 11, 14, 13, 17

After 2nd pass: 12, 11, 14, 13, 15, 17

After 3rd pass: 12, 11, 13, 14, 15, 17

After 4th pass: 11, 12, 13, 14, 15, 17

15. Which of the following statements is/are true?

S1: Heap sort is based on the selection method.

S2: Insertion sort is best when the best cases of input are considered (for comparisons based algorithms)

A. Only S2

B. Only S2

C. Both S1 and S2

D. Neither S1 nor S2

Ans. C

Sol. The following are best case of the sorting algorithms

Insertion sort : O(n)

Quick sort : O(n logn)

Merge sort : O(n logn)

Bubble sort : O(n^2)

Heap sort : O(n logn)

As we can see insertion sort has most efficient best case.

In heap sorting algorithm we extract the elements one by one. First the smallest (or largest) item is located and it is separated from the rest, then the next smallest (or next largest) is selected and so on until all items are separated.

Hence both S1 and S2 are true.

16. Which of the following procedure is suitable to find the longest path from a given vertex to any other given vertex in a directed acyclic graph (weighted) with few negative edge weights.

A. Divide and conquer B. Greedy approach

C. Dynamic programming D. All of these

Ans. C

Sol. Shortest path or longest path computation is possible using dynamic programming for directed acyclic graphs with presence of negative edge weights.

17. The number of distinct BFS traversal possible on complete graph of 5 vertices are ________ [vertices are labelled as A, B, C, D and E].

A. 24 B. 120

C. 60 D. 5

Ans. B

Sol. For a complete binary tree with n vertices the number of possible BFS traversal are n!. Here n=5, there number of possible BFS traversals= 5! =120.

18. What is the best data structure to implement topological sort on directed graph?

A. Heap B. Queue

C. Stack D. Array

Ans. C

Sol. The best data structure used to implement topological sort is stack, since topological sort based on depth first traversal.

19. Which of the following input will give best case time for selection sort?

A. 1 2 3 4 5 6 7 8 9 10 B. 2 3 1 5 9 7 8 6 10

C. 10 9 8 7 6 5 4 3 2 1 D. All of above take same amount of time

Ans. D

Sol. Selection sort in worst and best case take same time i.e., $O(n^2)$. So all the input take same time.

20. What is the worst case time complexity of inserting a node in a doubly linked list?

A. O(nlogn) B. O(logn)

C. O(n) D. O(1)

Ans. C

Sol. In the worst case, the position to be inserted maybe at the end of the list, hence you have to traverse through the entire list to get to the correct position, hence O(n).

21. You are given pointers to first and last nodes of a singly linked list, which of the following operations are dependent on the length of the linked list?

A. Delete the first element

B. Insert a new element as a first element

C. Delete the last element of the list

D. Add a new element at the end of the list

Ans. C

Sol. Deletion of the first element of the list is done in O (1) time by deleting memory and changing the first pointer.

Insertion of an element as a first element can be done in O (1) time. We will create a node that holds data and points to the head of the given linked list. The head pointer was changed to a newly created node.

Deletion of the last element requires a pointer to the previous node of last, which can only be obtained by traversing the list. This requires the length of the linked list.

Adding a new element at the end of the list can be done in O (1) by changing the pointer of the last node to the newly created node and last is changed to a newly created node.

22. Which of the following sorting algorithms can be used to sort a random linked list with minimum time complexity?

A. Insertion Sort

B. Quick Sort

C. Heap Sort

D. Merge Sort

Ans. D

Sol. Both Merge sort and Insertion sort can be used for linked lists. The slow random-access performance of a linked list makes other algorithms (such as quicksort) perform poorly, and others (such as heapsort) completely impossible. Since worst case time complexity of Merge Sort is O(nLogn) and Insertion sort is O(n^2), merge sort is preferred.

23. Consider an implementation of unsorted singly linked list. Suppose it has its representation with a head pointer only.

Given the representation, which of the following operation can be implemented in O(1) time?

i) Insertion at the front of the linked list

ii) Insertion at the end of the linked list

iii) Deletion of the front node of the linked list

iv) Deletion of the last node of the linked list

A. I and II

B. I and III

C. I, II and III

D. I, II and IV

Ans. B

Sol. We know the head node in the given linked list. Insertion and deletion of elements at the front of the linked list completes in O (1) time whereas for insertion and deletion at the last node requires to traverse through every node in the linked list. Suppose there are n elements in a linked list, we need to traverse through each node. Hence time complexity becomes O(n).

24. A normal queue, if implemented using an array of size MAX_SIZE, gets full when

A. Rear = MAX_SIZE – 1

B. Front = (rear + 1)mod MAX_SIZE

C. Front = rear + 1

D. Rear = front

Ans. A

Sol. When Rear = MAX_SIZE – 1, there will be no space left for the elements to be added in queue. Thus queue becomes full.

25. A queue follows __________

A. FIFO (First In First Out) principle

B. LIFO (Last In First Out) principle

C. Ordered array

D. Linear tree

Ans. A

Sol. Element first added in queue will be deleted first which is FIFO principle.

26. Assume that the operators +,-, X are left associative and ^ is right associative. The order of precedence (from highest to lowest) is ^, X, +, -. The postfix expression for the infix expression a + b X c – d ^ e ^ f is

A. abc X+ def ^^ –

B. abc X+ de^f^ –

C. ab+c Xd – e ^f^

D. -+aXbc^ ^def

Ans. B

Sol. Given Infix Expression is a + b X c – d ^ e ^ f.

(a b c X +) (d ^ e ^ f). '–' is present in stack.

(a b c X + d e ^ f ^ -). Thus the final expression is (a b c X + d e ^ f ^ -).

27. The **Little ω** notation is

A. Symmetric

B. Reflexive

C. Transitive

D. B & C only

Ans. C

Sol. Little ω due to strict bound follows only the transitive property.

28. This algorithm scans the list by swapping the entries whenever pair of adjacent keys are out of desired order.

A. Insertion sort.

B. Bubble sort.

C. Shell sort.

D. Quick sort.

Ans. B

Sol. Bubble sort is a simple sorting algorithm. This sorting algorithm is comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order. This algorithm is not suitable for large data sets as its average and worst case complexity are of $O(n^2)$ where **n** is the number of items.

29. Which of the following can be replaced by if

A. switch

B. while

C. continue

D. for

Ans. A

Sol. A **switch statement** can replace multiple if checks. It gives a more descriptive way to compare a value with multiple variants.

30. Pseudocode is used for

A. Denoting the program Flow

B. Program code

C. For coding the program

D. To write program steps

Ans. D

Sol. **Pseudocode** is an informal way of programming description that does not require any strict programming language syntax or underlying technology considerations. It is used for creating an outline or a rough draft to write a program.

31. Structured Programming is

A. Dividing the program into different program modules

B. Using Structures in the program

C. Using classes in the program

D. Parallelly executes the code

Ans. A

Sol. **Structured Programming Approach**, as the word suggests, can be defined as a programming approach in which the program is made as a single structure. It means that the code will execute the instruction by instruction one after the other. It doesn’t support the possibility of jumping from one instruction to some other with the help of any statement like GOTO, etc. Therefore, the instructions in this approach will be executed in a serial and structured manner. The languages that support Structured programming approach are: C, C++, Java, etc.

32. Continue Statement

A. Without Executing remaining statements takes control back to starting loop

B. Take control outside the loop

C. Continues to program end

D. None

Ans. A

Sol. The **continue statement** is used inside loops. When a **continue statement** is encountered inside a loop, control jumps to the beginning of the loop for next iteration, skipping the execution of **statements** inside the body of loop for the current iteration.

33. The running time of an algorithm is given by

$T(n) = T(n - 1) + T(n - 2) - T(n - 3)$ if $n > 3$ n otherwise

What is the order of this algorithm?

A. O(logn)

B. O(n)

C. $O(n^2)$

D. $O(n^3)$

Ans. B

Sol. $T(n) = T(n - 1) + T(n - 2) - T(n - 3), n > 3$

$T(4) = T(3) + T(2) - T(1)$

$= 3 + 2 - 1 = 4$

$T(5) = T(4) + T(3) - T(2)$

$= 4 + 3 - 2 = 5$

By induction

$T(n) = T(n - 1) + T(n - 2) - T(n - 3)$

$= (n - 1) + (n - 2) - (n - 3) = n$

$T(n) = O(n)$

34. Which of the following is correct statement?

A. Intermediate result of Kruskal algorithm is always forest.

B. Intermediate result of Kruskal algorithm is may be forest.

C. Intermediate result of Prim's algorithm is may be forest.

D. Intermediate result of Prim's algorithm is always forest.

Ans. B

Sol. Intermediate result of Prim's algorithm is always connected but Kruskal's algorithm may produce forest.

35. Consider the following C program:

```
int main ()
{
int x = 20;
static int y = x;
if (x = = y)
printf (“Equal”);
else
printf (“Not Equal”);
return 0;
}
```

What will be the output for above C program?

A. Equal

B. Not Equal

C. Runtime error

D. Compile time error

Ans. D

Sol. In C, static variable can be initialized only using constant literals. They can’t be assigned like this, so it will generate compile time error.

36. Output of the following program?

```
# include <stdio.h>
void fun(int *ptr) 2
{
*ptr = 30;
}
int main()
{
int y = 20;
fun(&y);
printf("%d", y);
return 0;
}
```

A. 20 B. 30

C. Compiler Error D. Runtime Error

Ans. B

Sol. The function fun() expects a pointer ptr to an integer (or an address of an integer). It modifies the value at the address ptr. The dereference operator * is used to access the value at an address. In the statement ‘*ptr = 30’, value at address ptr is changed to 30. The address operator & is used to get the address of a variable of any data type. In the function call statement ‘fun(&y)’, address of y is passed so that y can be modified using its address.

37. The operator used to get value at address stored in a pointer variable is:

A. * B. &

C. && D. ||

Ans. B

Sol. is a dereferencing pointer. It is used to retrieve the value stored at some address in memory.

38. Is there any difference between following declarations?

S1: extern int fun();

S2: int fun();

A. Both are identical

B. No difference, except extern int fun(); is probably in another file

C. int fun(); is overrided with extern int fun();

D. None of these

Ans. B

Sol. extern int fun(); declaration in C is to indicate the existence of a global function and it is defined externally to the current module or in another file.

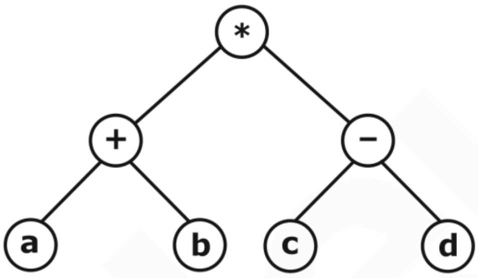
int fun(); declaration in C is to indicate the existence of a function inside the current module or in the same file.

39. The postfix equivalent of the tree whose prefix order is → * + ab – cd is:

A. ab + cd - *
B. ab cd + - *
C. ab + cd * -
D. ab + - cd *

Ans. A

Sol. Using the given prefix order we get the infix order as (a+b)*(c-d) . Now we can find the tree using the prefix and infix expression. The tree is shown below.



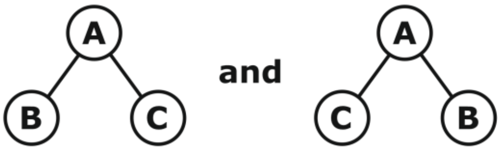
From the tree we can find the postfix expression.

40. The number of possible ordered trees with 3 nodes A, B, C is:

A. 16
B. 12
C. 6
D. 10

Ans. B

Sol. The tree maybe of depth 2 or 1. If depth is 2, we have 6 possible trees. This is because one of the 3 nodes A, B, C may be the root and the next level may be one of the remaining 2 nodes. If the depth is 1, the root maybe one of the 3 nodes A, B, C. Corresponding to a root, say A, 2 trees are possible as this.



This gives us 6 more possibilities.

41. Given a hash table with 4 keys and 10 slots, with simple uniform hashing. If collisions are resolved by chaining then the probability that first slot ends up empty?

A. 0.65
B. 0.54
C. 0.73
D. 0.34

Ans. A

Sol. Each key as 1/10 probability of hashing into first slot.

∴ Probability of not hashing into first slot is $1-\frac{1}{10}$.

∴ The probability that no key hashes into first slot = $\left(1-\frac{1}{10}\right)^4 = \left(\frac{9}{10}\right)^4 = 0.656$

42. Assume that a merge sort algorithm in the worst case takes 30 seconds for an input of size 64. Which of the following most closely approximates the maximum input size of a problem that can be solved in 6 minutes?

A. 256

B. 512

C. 1024

D. 2048

Ans. B

Sol. Time complexity of merge sort is Θ(nLogn)

c*64 Log64 is 30

c*64*6 is 30

c is 5/64

For time 6 minutes:

5/64*nLogn = 6*60

n Log n = 72*64 = 512 * 9

n = 512.

43. An entity set that does not have sufficient attributes to form a primary key is termed a ____________

A. Strong entity set

B. Variant set

C. Weak entity set

D. Variable set

Ans. C

Sol. An entity set that has a primary key is termed a strong entity set.

44. The entity relationship set is represented in E-R diagram as

A. Double diamonds

B. Undivided rectangles

C. Dashed lines

D. Diamond

Ans. D

Sol. Dashed lines link attributes of a relationship set to the relationship set.

45. Which one of the following uniquely identifies the elements in the relation?

A. Secondary Key

B. Primary key

C. Foreign key

D. Composite key

Ans. B

Sol. Primary key checks for not null and uniqueness constraint.

46. Data integrity constraints are used to:

A. Control who is allowed access to the data

B. Ensure that duplicate records are not entered into the table

C. Improve the quality of data entered for a specific property

D. Prevent users from changing the values stored in the table

Ans. C

Sol. The data entered will be in a particular cell (i.e., table column).

47. An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A.This is called as

A. One-to-many B. One-to-one

C. Many-to-many D. Many-to-one

Ans. B

Sol. Here one entity in one set is related to one one entity in other set.

48. An _________ is a set of entities of the same type that share the same properties, or attributes.

A. Entity set B. Attribute set

C. Relation set D. Entity model

Ans. A

Sol. An entity is a "thing" or "object" in the real world that is distinguishable from all other objects.

49. "Find all students who have taken all courses offered in the Biology department." The expressions that matches this sentence is :

A. $\ni t \varepsilon r (Q(t))$ B. $\forall t \varepsilon r (Q(t))$

C. $\neg t \varepsilon r (Q(t))$ D. $\sim t \varepsilon r (Q(t))$

Ans. B

Sol. ∀ is used denote "for all" in SQL.

50.$\{t \mid \ni s \varepsilon \text{ instructor } (t[name] = s[name]$

$\wedge \ni u \varepsilon \text{ department } (u[dept\ name] = s[dept\ name] \wedge u[building] = \text{"Watson"}))\}$

Which of the following best describes the query?

A. Finds the names of all instructors whose department is in the Watson building

B. Finds the names of all department is in the Watson building

C. Finds the name of the dapartment whose instructor and building is Watson

D. Returns the building name of all the departments

Ans. A

Sol. This query has two "there exists" clauses in our tuple-relational-calculus expression, connected by and (∧).

51. Which of the following is a fundamental operation in relational algebra?

A. Set intersection B. Natural join

C. Assignment D. None of the mentioned

Ans. D

Sol. The fundamental operations are select, project, union, set difference, Cartesian product, and rename.

52. Consider the following statements:

S1: The result of EXISTS is a Boolean value TRUE if the nested query result contains at least one tuple, or FALSE if the nested query result contains no tuples.

S2: The relations resulting from the set operations like UNION, EXCEPT and INTERSECT contain duplicate tuples.

Which of the following is correct?

A. Only S1 is true B. Only S2 is true

C. Both S1 and S2 are true D. Neither S1 nor S2 is true

Ans. A

Sol. The EXISTS function in SQL is used to check whether the result of a correlated nested query is empty (contains no tuples) or not. The result of EXISTS is a Boolean value TRUE if the nested query result contains at least one tuple, or FALSE if the nested query result contains no tuples.

There are mathematical set operations like set union (UNION), set difference (EXCEPT) and set intersection (INTERSECT) operations. The relations resulting from these set operations are sets of tuples; that is, duplicate tuples are eliminated from the result. These set operations apply only to union-compatible relations, so we must make sure that the two relations on which we apply the operation have the same attributes and that the attributes appear in the same order in both relations.

Therefore, out of the given statements, only statement S1 is true.

53. The property of a transaction that persists all the crashes is

A. Atomicity B. Durability

C. Isolation D. All of the above

Ans. B

Sol. After a transaction completes successfully, the changes it has made to the database persist, even if there are system failures.

54. The primary key must be

A. Unique B. Not null

C. Both Unique and Not null D. Either Unique or Not null

Ans. C

Sol. Primary key must satisfy unique and not null condition for sure.

55. The result which operation contains all pairs of tuples from the two relations, regardless of whether their attribute values match.

A. Join

B. Cartesian product

C. Intersection

D. Set difference

Ans. B

Sol. Cartesian product is the multiplication of all the values in the attributes.

56. Which one of the following is a procedural language?

A. Domain relational calculus

B. Tuple relational calculus

C. Relational algebra

D. Query language

Ans. C

Sol. Domain and Tuple relational calculus are non-procedural language. Query language is a method through which database entries can be accessed.

57. An index is clustered, if:

A. It is on a set of fields that form a candidate key

B. It is on a set of fields that include the primary key

C. The data records of the file are organized in the same order as the data entries of the index

D. the data records of the file are organized not in the same order as the data entries of the disk

Ans. C

Sol. An index is clustered, if the data records of the file are organized in the same order as the data entries of the index

58. Choose the correct statement:

A. For fixed length records unspanned organization is preferred

B. For variable length records unspanned organization is preferred

C. For fixed length records spanned organization is preferred

D. None of the above

Ans. A

Sol. For fixed length record unspanned organization is preferred. A block is the unit of data transfer between disk and memory. Unspanned records are records that is found in one and only block. Records do not span across block boundaries and are used with fixed length.

59. If a relation is in BCNF, it is also in:

A. 1NF

B. 2NF

C. 3NF

D. All of the above

Ans. D

Sol. BCNF satisfied all the conditions that are followed from 1NF and 2NF and is more strict than 3NF in the sense for the FD: X→Y, BCNF states X must be a candidate key, unlike 3NF.

60. For a database relation R (a, b, c, d) where the domains of a, b, c and d only include atomic values, only the following functional dependency and those that can be inferred from them hold:

a → c , b→ d

The relation is in:

A. 1NF but not in 2NF

B. 2NF but not in3NF

C. 3NF

D. None of the above

Ans. A

Sol. Since, it is already mentioned that a, b, c, d are atomic values, so by definition of 1NF, it can be inferred that it is already in 1NF.

But a→ c

b→d, the candidate key is → {ab}

So, all the non-key attributes only partially depend on the prime attributes of the candidate key(a and b). So, it is not in 2NF.

****