GATE

Subject : CS 2003 - SOLUTIONS

(Q. NO. 1 – 30) 1 MARKS

1. The function returns the value

$$s = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots\ldots$$

It is nothing but the implementation of Taylor's series to calculate e^x.

(B) is the answer.

2. Here A is an array of 10 pointers whereas B is a two - dimensional array.

I is a valid left hand side of an assignment as A[2] is a pointer.

II is also valid as A[2][3]will refer to the data to which the pointer points.

III is invaild. In C, while using multi dimensional arrays the parameter declaration must include all other dimensions except the first dimension i.e. the number of rows is optional. But here we are not specifying the relevant second dimension and so the compiler will give an error.

IV is valid as B is a true 2-dimensional array.

So, all except III are valid left hand sides of assignment in C.

(A) is the answer.

3. Here the probabilities P(A) and P(B) are independent probabilities.

Given that P(A) = 1, P(B) = 1/2

$$P(A/B) = \frac{P(A \cap B)}{P(B)}$$

$$= \frac{P(A) \,.\, P(B)}{P(B)} \quad \left\{P\left(A \cap B\right) = P(A).P(B) \text{ as A and B are independent}\right\}$$

$$= P(A)$$

$$\therefore P(A/B) = 1$$

$$P(B/A) = \frac{P\left(B \cap A\right)}{P(A)}$$

$$= \frac{P(B).P(A)}{P(A)}$$

$$= P(B)$$

$$\therefore P(B/A) = 1/2$$

(D) is the answer.

4. A is a sequence of 8 distinct integers sorted in ascending order. We can select a sub-array of 5 elements (in sorted order) in ${}^8C_5$ ways. So the remaining 3 elements will also form a sorted sub-array.

$\therefore$ Number of distinct pairs $= {}^8C_5$

$= {}^8C_3$

$= 56$

(C) is the answer.

5. Each couple can go to the party in 3 different ways:

(1) Both husband and wife go.

(2) Only wife goes.

(3) Nobody goes.

And since we have n couples, number of different gatherings possible at the party = 3^n

(B) is the answer.

6. To form a binary search tree with n elements, we have one key as the root, the left subtree will have $(k-1)$ keys less than the root and the right subtree will have all the remaining keys greater than the root i.e. $(n-(k-1))$ keys.

So, right subtree will have keys = $n - k + 1 - 1 = n - k$

So, taking k as the root, the number of different binary search trees is given by :

$$T(n) = \sum_{k=1}^{n} T(k-1)T(n-k)$$

(B) is the answer.

7. For the concatenation of two strings, the identity element will be ε.

For any string w in the set $\sum *$ over $\{0,1\}$, we have

$$\varepsilon.w = w.\varepsilon = w$$

$\therefore$ Identity element exists

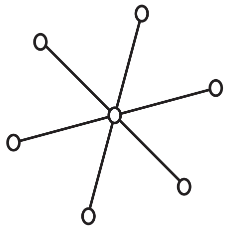
So, it is a monoid.

But we do not have inverse for any of the string.

So, the inverse does not exist for any string and hence it is not a group.

(A) is the answer.

8. • In the worst case, the graph will be a star



Here n = 7 and k = 1

Now if we remove the central vertex, we get 6 connected components i.e. $7-1$

So, maximum connected components = $n-1$

• In the best case, the removed vertex itself is a separate connected component. So removing that vertex will get us one less connected component than the initial ones, i.e. $k-1$.

(C) is the answer.

9. The given number is in 2's complement form

$(11111011)_2 = (-5)_{10}$

(A) $(11100111)_2 = (-25)_{10}$

(B) $(11100100)_2 = (-28)_{10}$

(C) $(11010111)_2 = (-41)_{10}$

(D) $(11011011)_2 = (-37)_{10}$

So we can see that –25 is divisible by –5.

(A) is the answer.

10. I is about data hazard.

II is about control hazards due to conditional jumps.

III is about structural hazard as the CPU has only one ALU.

(D) is the answer.

11. To multiply two n – bit numbers in an array multiplier, number of gates needed = 2n–1 and each gate has 1 unit delay.

$\therefore$ Total delay of the multiplier $= (2n-1)*1$

$= 2n-1$

$= \theta(n)$

(C) is the answer.

12. • A problem H is NP – hard when every problem L in NP can be reduced in polynomial time to H. The given prolem π can be reduced in polynomial time to 3–SAT (NP-Complete problem). So, π is NP hard. Since a polynamial time reduction is possible from π to 3-SAT, so π is in NP.

$\pi \in$ NP and $\pi \in$ NP-hard.

$\therefore$ π is NP- Complete.

(C) is the answer

13. When P = NP, we have L = (0+1)* i.e. L is regular and hence recursive.

When P $\neq$ NP, we have L = ϕ i.e. L is regular and recursive as well.

(A) is the answer.

14. The given regular expression is R = 0*(10*)*

$\therefore$ L(R) = { ε, 0, 00, 01, 010, 11, 111, 01010, 011,......}

(B) does not accept 11. So, it can be eliminated.

(C) does not accept ε. So, it can be eliminated as well.

(A) generates all the strings generated by R.

So, (A) is the answer.

15. Since it is given that L can be effectively enumerated in lexicographic order, then we can have a Turing machine that accepts L.

    The Turing machine will halt correctly at final state if the string is in lexicographic order else it will reject the string and halt in non-final state.

    So, L is recursive.

    However there is no way in which we can construct a PDA for the given language L.

    So, L is recursive but not context-free.

    (D) is the answer.

16. For a given grammar to be LL (1), it should satisfy the following requirments :

    i) The grammar should be free from left recursion.

    ii) The grammar should not be left factored.

    iii) The grammar should be unambiguous.

    (D) is the answer.

17. The parsing tables of SLR and LALR parser for any grammar G have equal number of states.

    $\therefore$ n1 = n2

    (B) is the answer.

18. A Syntax Directed Definition (SDD) is called S-attributed if it has only synthesized attributes, whereas L-attributed definitions contain both synthesized and inherited attributes.

    So inherited attributes in a SDD can be evaluated only if the definition is L-attributed .

    (B) is the answer.

19. The in-order traversal of a binary search tree always produces the sequence of keys in an increasig order.

    So, here we do not even need to construct the tree but rather check the options and find the increasing order of the sequence of elements in a BST.

    (C) is the answer.

20. $(n+k)^m = \theta(n^m)$ as k and m are constants.

    So, I is true.

    II) is also true because $2^{n+1} = \frac{2^n}{2} = O(2^n)$

    III) is false $2^{2n+1} = \frac{2^{2n}}{2} = O(2^{2n})$

    and $2^{2n} > c.2^n$ for all $c > 0$

    $\therefore 2^{2n+1} \neq O(2^n)$

    (A) is the answer.

21. In DFS (Depth First Search), we start at the root (selecting some arbitrary node as the root in the case of a graph) and explore as far as possible along the branch before backtraking.

For the given graph, I, III and IV are the possible depth first traversals. II is not a DFS traversal as e comes after f and there is no direct link between e and f.

(D) is the answer.

22. Even if we use binary search to identify the correct position of an element in insertion sort, the worst case running time would still remain $\theta(n^2)$ because we would still require O(n) swaps to insert the element at its correct position.

So for n elements, we have

- Time to identify their correct position = O(nlogn)
- Time to insert elements at their correct position = $O(n^2)$

$\therefore$ Running time of the algorithm $= O(n \log n + n^2)$

$= O(n^2)$

(A) is the answer.

23. In a min-heap, the smallest element can be found in O(1) time.

For a heap with n elements, the 7th smallest element will be present in the first 7 levels from the root.

So, we need to check only 2^7-1 = 127 nodes which is independent of n.

$\therefore$ Time to find 7th smallest element = O(1)

However, if we are not allowed to traverse the heap and allowed to perform only default heap operations, then the time required will be θ (logn) as we will have to perform the following steps.

i) Delete min + Reverse heapify 6 times = O (6 logn)

ii) Extract 7th minimum = O(1)

iii) Insert 6 elements + Reverse heapify to restore the original heap = O (6 logn)

But here we are just traversing the heap. So time to find 7th minimum is constant.

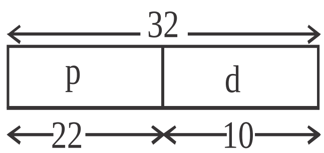
(D) is the answer.

25. Using a larger block size in a fixed block size file system increases the disk throughput as the number of blocks would decrease thus reducing the time spent moving the heads across the disk, which would result in more continuous reads / writes per second.

But larger block size can also lead to wastage of space when only small sized file is required and hence poor utilization.

(A) is the answer.

26. Virtual address = 32 bit



Page size = 1 KB

$= 2^{10}$B

So, p = 10 bits

$\therefore$ d = 32 – 10 = 22 bits

i.e. we require a page table that can occupy 2^{22} pages which is very large. So using single level page table, we require large memory overhead in maintaining page tables.

(C) is the answer.

27. In source routing, the sender of a packet can partially or completely specify the route of the packet.

Whereas in non-source routing, routers in the network determine the path of the packet on its destination.

(D) is the answer.

28. End to end connectivity must be implemented by a transport layer protocol. Both TCP and UDP are transport layer protocols.

In-order delivery of packets, detection of duplicate packets and recovery from packet losses is done by TCP but not by UDP.

(D) is the answer.

29. If the transaction reads data item after it is written by an uncommitted transaction, then it may lead to irrecoverability in the database system.

If the uncommitted transaction fails or rolls back, then the transaction that had read its data item will have acted on data that was never committed i.e. never existed.

So irrecoverable error occurs when an uncommitted transaction fails.

(D) is the answer.

30. The given query is equivalent to cross product of the relations r1, r2, ...,rm and selects those tuples where the condition P is true and projects the attributes a1, a2, ..., an. So, the equivalent relational algebra expression is same as option (A).

(A) is the answer.

(Q. NO. 31 – 90) 2 MARKS

31. a and b are given as minimal elements. So, no other element in S is of lower order than either a or b.

Also c is given as the maximal element. So, there is no other element in S having higher order than c.

P(a) = True means all elements 'x' which have an edge from element 'a' have to be true.

So in the statement $P(a) \rightarrow P(x)$ can be done only by setting P(x) = True.

Elements which have an edge from b always satisfy $P(b) \rightarrow P(x)$ as P(b) = False.

So options (A) and (B) are true.

(C) can also be true as $b \leq x$ ensures that here $x \neq a$ and hence for all the elements P(x) = False would still follow the given implication.

(D) cannot be true as P(a) = true, $\forall$ x such that a<=x. And one such x is c which is the maximal element.

(D) is the answer.

32. Only the formula $(\forall x)[\alpha \Rightarrow \beta] \Rightarrow \left((\forall x)[\alpha] \Rightarrow (\forall x)(\beta)\right)$ is a vaild first order formula.

(D) is the answer.

[Hint : Take an example for α and β and find the truth values of the statements given in the options.]

33. Q_{yy} is always true as every number divides itself.

$\therefore (\forall y)[Q_{xy} \leftrightarrow \neg Q_{yy}]$ is same as $(\forall y)$ $[Q_{xy} \leftrightarrow \text{False}]$

Also, $(\forall y)$ $[Q_{xy} \leftrightarrow \text{False}]$ can be simplified to $(\forall y)[\neg Q_{xy}]$

So, α becomes

$$\alpha = (\forall x)\left[P_x \leftrightarrow (\forall y)[\neg Q_{xy}]\right] \rightarrow (\forall x)[\neg P_x]$$

Now consider LHS of α

$$\text{LHS} = (\forall x)\left[P_x \leftrightarrow (\forall y)[\neg Q_{xy}]\right]$$

$(\forall y)$ $[\neg Q_{xy}]$ means not all values of y divide x which is true.

$$\therefore \text{LHS} = (\forall x)[P_x \leftrightarrow \text{True}]$$
$$= (\forall x)(P_x)$$
$$\text{So}, \alpha = (\forall x)(P_x) \rightarrow (\forall x)[\neg P_x]$$

Now consider the statements I_1 and I_2:

For I_1, $(\forall x)$ $[Px]$ will be false as not every number x will be a prime number. So, α becomes true. [$\because$ False $\rightarrow$ P is always true for any P]

For I_2, as well $(\forall x)(Px)$ will be false as not every number x will be a composite number. So α will be true for I_2.

Thus, both I_1 and I_2 satisfy α

(D) is the answer.

34. The given question is same as number of ways in which r identical objects can be distributed in n distinct boxes i.e. ${}^{r+n-1}C_{n-1}$ ways

Here we have m identical balls and it is give n that $m \geq kn$

$\therefore m - kn \geq 0$

and n distinct bags

$\therefore r = m - kn$ and $n = n$

$\therefore$ Required number of ways = ${}^{m-kn+n-1}C_{n-1}$

(B) is the answer.

35. For $m = 1$, $T(m^2) = T(1)$

$T(1) = 1$

For $m = 2$, $T(m^2) = T(4)$

$T(2) = T(1) + \lfloor \sqrt{2} \rfloor$
$= 1+1$
$= 2$

$T(3) = T(2) + \lfloor \sqrt{3} \rfloor$
$= 2+1$
$= 3$

$\therefore T(4) = T(3) + \lfloor \sqrt{4} \rfloor$
$= 3 + 2$
$= 5$

For m = 3, $T(m^2) = T(9)$
$T(9) = 16$
Similarly for m = 4, $T(m^2) = T(16)$
$T(16) = 38$
Now, we can check with options.
(A) gives correct values up to T(9) but it produces 34 for T(16).
(B) gives correct result for all values of m.
(C) and (D) do not give correct result for T(4).
So, (B) is the answer.

36. For a complete graph with n vertices,
i) if n is odd, then there is no perfect matching.
ii) if n is even, then perfect matchings can be counted as (n-1) (n-3)(n-5)......3.1
So in the given question, n = 6
$\therefore$ # Perfect matchings $= (6-1)(6-3)(6-5)$
$= 5 \times 3 \times 1$
$= 15$
(A) is the answer.

37. Given that $f : A \rightarrow B$ is a one - one function $\therefore |B| \geq |A|$ so $g : 2^A \rightarrow 2^B$ will also be one - one.
where $g(c) = \{ f(x) \mid x \in c\}$ $\therefore$ Range size = $|2^A|$
However $h : 2^B \rightarrow 2^A$ will not be one-one
where $h(D) = \{x \mid x \in A, f(x) \in D\}$ as every subset of B will be mapped to that subset of A for which it has all the images of subset of A. Here range size will be $|2^A|$.
So, g (h (D)) will also have range size as $|2^A|$.
We know that $|A| \leq |B|$. So $|2^A| \leq |2^B|$
$\therefore g(h(D)) \subseteq D$
(A) is the answer.

38. Consider each case separately :

i) $x = a$ and $y = a$

$(a \times a) + (a \times a) = a + a = b \neq c$

ii) $x = a$, and $y = b$

$(a \times a) + (a \times b) = a + b = a \neq c$

iii) $x = a$ and $y = c$

$(a \times a) + (a \times c) = a + c = c$

$(b \times a) + (c \times c) = b + b = b \neq c$

iv) $x = b$ and $y = a$

$(a \times b) + (a \times a) = b + a = a \neq c$

v) $x = b$ and $y = b$

$(a \times b) + (a \times b) = b + b = a \neq c$

vi) $x = b$ and $y = c$

$(a \times b) + (a \times c) = b + c = c$

$(b \times b) + (c \times c) = c + b = c$

$\therefore$ (b, c) is a solution

vii) $x = c$ and $y = a$

$(a \times c) + (a \times a) = c + a = a \neq c$

viii) $x = c$ and $y = b$

$(a \times c) + (a \times b) = c + b = c$

$(b \times c) + (c \times b) = a + c = c$

$\therefore$ (c, b) is a solution.

ix) $x = c$ and $y = c$

$(a \times c) + (a \times c) = c + c = b \neq c$

So, we have 2 solutions (b,c) and (c, b)

(C) is the answer.

39. p_i denotes the ith prime number.

$\therefore\ p_1 = 2, p_2 = 3, p_3 = 5$

Also $f(s) = \pi_{i=1}^{n}\ p_i^{g(a_i)}$

So, f (s) is a product of 3 prime numbers. By looking at the options we can say the length of the string to be encoded is 3.

So for first lettar of the string we have $f(s) = 2^x$ for some x.

And $h(s_i......s_n) = \pi_{i=1}^{n}\ p_i^{f(s_i)}$

$\therefore\ h(s_1) = 2^{2^x}$

So by looking at the options we can say that

$s_1 = a$
$f(a) = 2^3 = 8$
Also $s_2 = s_3 = a$
$\therefore$ We get h as

$h(s_1) = 2^{f(s_1)} = 2^{f(a)} = 2^8$

$h(s_2) = 3^{f(s_2)} = 3^{f(a)} = 3^8$

$H(s_3) = 5^{f(s_3)} = 5^{f(a)} = 5^8$

$\therefore$ aaa is the string to be encoded and its encoding h is $2^8 3^8 5^8$
(B) is the answer.

40. By handshaking lemma,
The sum of degree of vertices in a graph is twice the edges.

$$\sum_{i=1}^{n} \deg(v_i) = 2e$$

Let the minimum degree of each of the vertex in the graph be m

So, we have at least $\frac{m \times |v|}{2}$ edges.

$$\therefore e \geq \frac{m \times |v|}{2}$$

So, $m \leq \frac{2e}{|v|}$ ——— (1)

It is given that

$e \leq 3|v| - 6$ ——— (2)

Put (2) in (1)

$$\therefore m \leq \frac{2(3|v| - 6)}{|v|}$$

$$\therefore \boxed{m \leq 6 - \frac{12}{|v|}}$$

So we can see that for m = 6, the inequality becomes

$0 \leq \frac{-12}{|v|}$ which is not possible

So, minimum degree cannot be 6.
(D) is the answer.

41. Let the augmented matrix be

$$[A:B]=\left[\begin{array}{ccc:c}2 & 1 & -4 & \alpha\\ 4 & 3 & -12 & 5\\ 1 & 2 & -8 & 7\end{array}\right]$$

$R_1 \leftrightarrow R_3$

$$[A:B]\sim\left[\begin{array}{ccc:c}1 & 2 & -8 & 7\\ 4 & 3 & -12 & 5\\ 2 & 1 & -4 & \alpha\end{array}\right]$$

$R_2 \to R_2 - 4R_1$

$R_3 \to R_3 - 2R_1$

$$\therefore [A:B]\sim\left[\begin{array}{ccc:c}1 & 2 & -8 & 7\\ 0 & -5 & 20 & -23\\ 0 & -3 & 12 & \alpha-14\end{array}\right]$$

$R_3 \to R_3 - R_2$

$$\therefore [A:B]\sim\left[\begin{array}{ccc:c}1 & 2 & -8 & 7\\ 0 & 1 & 4 & -23/5\\ 0 & 2 & -8 & \alpha-21\end{array}\right]$$

$R_2 \to R_2/(-5)$

$$\therefore [A:B]\sim\left[\begin{array}{ccc:c}1 & 2 & -8 & 7\\ 0 & 1 & 4 & -23/5\\ 0 & 2 & -8 & \alpha-21\end{array}\right]$$

$R_3 \to R_3 - 2R_2$

$$\therefore [A:B]\sim\left[\begin{array}{ccc:c}1 & 2 & -8 & 7\\ 0 & 1 & -4 & \frac{23}{5}\\ 0 & 0 & 0 & \alpha-21-\frac{46}{5}\end{array}\right]$$

So, for the given system to have infinitely many solutions, we must have.

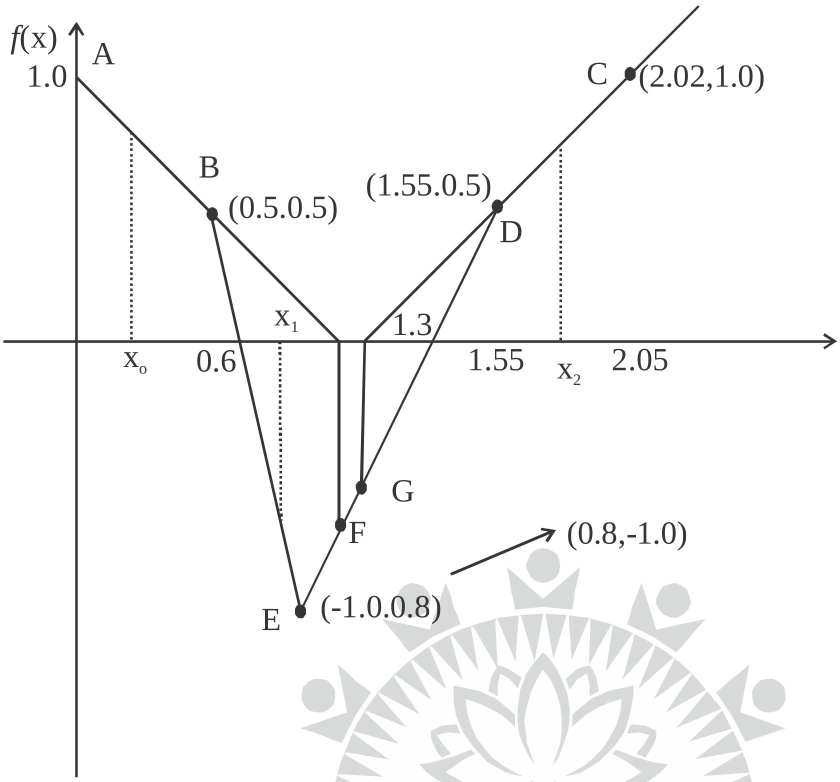
$$\alpha-21-\frac{46}{5}=0$$

$\therefore \alpha = 30.2$

So for only value of α, the given system has infinitely many solutions.

(B) is the answer.

42



Here f (x) = 0 i.e. y =0

So we have to find roots of f (x) = 0 for initial guesses i.e. x_0, x_1 and x_2

- For point x_0, the line AB itself is the tangent. So we have to find where the line AB cuts X – axis Equation of line AB.

$$y - 0 = \frac{0.5 - 0}{0.5 - 1}(x - 1)$$

$$y = -(x - 1)$$

$$y = -x + 1$$

On X-axis, y = 0, $\therefore$ x = 1

$\therefore$ The tangent cuts X – axis at the point (1,0). So, the point on the curve corresponding to (1,0) is shown by F and tangent at point F is the line DE itself which cuts X – axis at 1.3 and f (1.3) = 0. So, 1.3 is a root.

- For point x_1, line BE itself is the tangent which cuts X – axis ay 0.6 and f (0.6) = 0. So, 0.6 is a root.

- For point x_2, tangent at this point is line CD.

Equation of line CD : x – y = 1.05 and this line cuts X – axis at x = 1.05 and tangent at this point is line DE, which cuts X – axis at 1.3 and f (1.3) = 0. So 1.3 is a root .

(D) is the answer.

43. Here biasing is done by adding 31 to the exponent so the actual value of the exponent will be 31 less than the given value.

Given that e is of 6 bits, the maximum value it can have is 111110 which is 62.

We cannot store all 1's as it will be used to represent special numbers like infinity or NAN.

The set R of representable real numbers is given by

$$R = \left\{ \pm \left(1 + [0, +511].2^{-9} \right) 2^{[0,62]-31} \right\} \cup \{0\}$$

$\therefore$ Largest number $= 1.111111111 \times 2^{62-31}$

$= (2 - 2^{-9}) \times 2^{31}$

$\therefore$ Second Largest number $= 1.111111110 \times 2^{62-31}$

$= (2 - 2^{-8}) \times 2^{31}$

$\therefore$ Difference between these two numbers

$= \left(2 - 2^{-9}\right) \times 2^{31} - \left(2 - 2^{-8}\right) \times 2^{31}$

$= \quad 2^{31}\left[2 - 2^{-9} - 2 + 2^{-8}\right]$

$= \quad 2^{31}\left[2^{-8} - 2^{-9}\right]$

$= \quad 2^{31} \times 2^{-9}$

$= \quad 2^{22}$

(C) is the answer.

44. Here we need to count the difference of the number of 0's and 1's in the first k bits of a number until the difference reaches 2 or -2 [as $n_k - z_k = 2$ is same as $z_k - n_k = -2$]

So we have the following possiblities of difference :

$-2, -1, 0, 1, 2$ which needs to be represented by 5 different states in the state transition diagram.

(A) is the answer.

45. Using POS representation,

xy \ zw	00	01	11	10
00	×		0	
01	0		×	0
11		×	×	0
10	×	0	0	×

$f = (x + z + w)\ (\bar{y} + \bar{z})\ (\bar{z} + \bar{w})(\bar{x} + y)$

So number of literals in POS form of f = 9

Using SOP representation,

xy \ zw	00	01	11	10
00	×	1		1
01		1	×	
11	1	×	×	
10	×			×

$f = yw + \bar{y}\,\bar{w} + \bar{x}\,\bar{z}\,w + xy\bar{z}$

So number of literals in SOP form of f = 10

(C) is the answer.

46. When k = 1 and $C_o = 1$, we can do A – B when k = 0 and $C_o = 0$, we can do A + B. We cannot perform A + 1 here as we need to set B = 1 which cannot be done by using just two control lines k and C_o.

(A) is the answer.

47. (D) is the answer.

48. The given code counts the number of 1 bits in A0.

Here the value of B is decremented by 1 only if carry flag is 1 and carry bits are filled using right rotation.

(B) is the answer.

49. Here A is a 8-bit register and it is rotated via carry 8 times so now after one more rotation, the original value of register A will be restored.

So the instruction to be inserted at location X will be :

RRC A, #1

(A) is the answer.

50. We have to find the set of 7-bit binary strings in which the 1st, 4th and the last bits are 1.

So, s will be of the form.

$s = \underline{1}\ \_\ \_\ \underline{1}\ \_\ \_\ \underline{1}$

<u>Case I:</u> When 5th and 6th bits are 00, we will have 4 choices for the 2nd and 3rd bit positions i.e. 00, 01, 10 and 11. So, the strings accepted will be:

i) 1001001

ii) 1011001

iii) 1101001

iv) 1111001

<u>Case II :</u> When the 2nd and 3rd bits are 00

So strings accepted will have 4 choices for 5th and 6th bit positions i.e. 00,01,10 and 11.

The strings accepted by the automata will be:

i) 1001001

ii) 1001011

iii)1001101

iv)1001111

Taking union of cases I and II, we get 7 distinct strings.

(C) is the answer.

51. G : $S \to aSb$

$S \to SS$

$S \to \varepsilon$

(A) is false as the given grammar is ambiguous. It produces two distinct parse trees for the string 'ε' itself.

(B) is false as the given grammar accepts the language $(a^n b^n)^*$ which is a DCFL and DCFLs are closed under Kleen's closure.

(C) is true. The given grammar is a DCFL and hence we can have a deterministic PDA for the same.

(D) is false. We cannot have a DFA for L(G).

(C) is the answer.

52. Here L_1 and L_2 both should have same complexity i.e. belong to the same class. f is a polynomial time computable bijection. So, f^{-1} is also polynomial time computable bijection.

(A) is true as L1 can also be infinite.

(B) is true P problems are by default in NP.

(C) cannot be true, as if L1 is decidable the L2 must also be decidable because L1 is is polynomial time reducible to L2.

(D) is true as L1 can be recursive too.

(C) is the answer.

53. Consider option (A),

w = 0 and $w \in (0+1)^+$

$$\begin{aligned} \therefore (q_{0,} 0) &= 1q_1B \\ &= q_0 1B \\ &= 1q_1, B \\ &= q_0 1B \end{aligned}$$

So on the input 0, the Turing Machine M doesn't halt at all.

Hence, (A) is true. So we don't need to check other options.

(A) is the answer.

54. L_0 halts on the input w where L_1 doesn't.

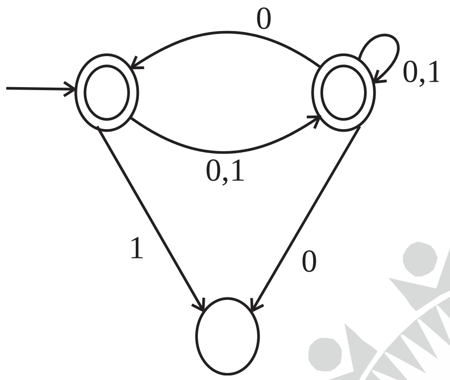
So, $L = L_0 \cup L_1$ is recursively enumerable but not recursive (as L_1 doesn't halt on input w).

If L is recursively enumerable, then because not as recursively enumerable languages are not closed under complementation.

(A) is the answer.

55. L(M) = {1,00,10,000,001,010,0100,......}

Now M_1 will be



So, $L(M_1) = L_1 = \{\varepsilon, 0,1,00,01,10,11,000,001,...\}$

We can see that L_1 accepts all strings over {0,1}

$\therefore L_1 = (0+1)^*$

(B) is the answer.

56. First (S) = {i, a}

First (S') ={e, ε }

First (E) = {b}

We need to consider follow (S') as first (S') contains ε.

follow (S') = {e, $}

Now, consider the predictive parsing table M for this grammar.

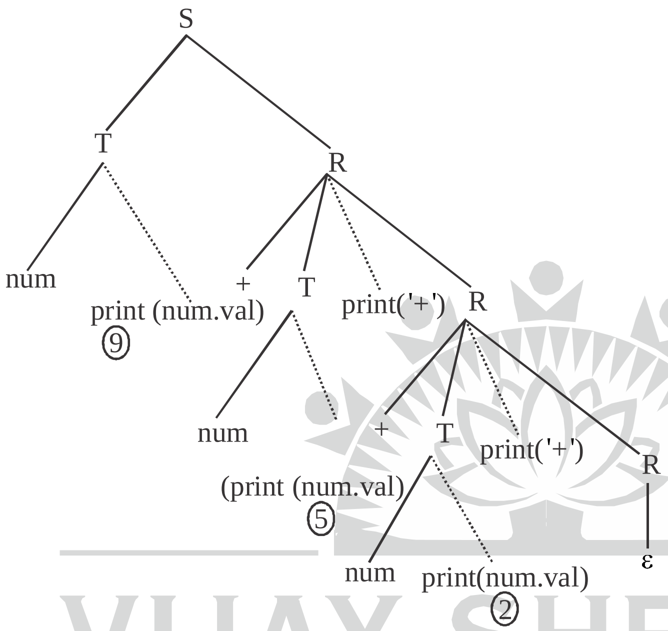
	a	i	b	e	t	$
S	$S \to a$	$S \to ietSS'$				
S^1				$S' \to eS$ $S' \to \varepsilon$		$S' \to \varepsilon$
E			$E \to b$			

So the entries M [S', e] and M [S' $] are {$S' \to \varepsilon$} respectively.

(D) is the answer.

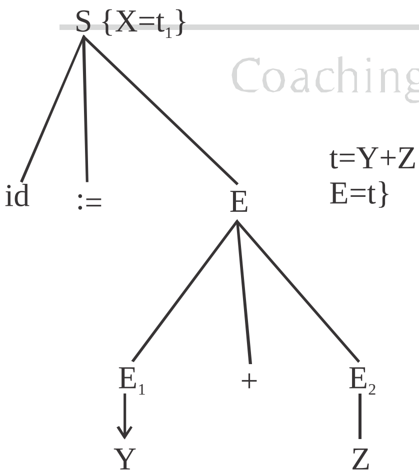
57. first (cC) = c
first (d) = d
$\therefore$ first (cC) $\cap$ first (d) $= \phi$
So there are no multiple entries in the predictive parsing table and hence the grammar is LL(1).
(A) is the answer.

58. For the input string '9+5+2', the tree generated is as follows :



So, to obtain the output, we traverse the generated tree in DFS order.
The output will be printed as : 95+2+
(B) is the answer.

59.



So, the 3- address code generated by the definition ‘X : = Y + Z’ is :
t = Y + Z
X = t
(B) is the answer.

60. f1 and f2 are executed sequentially.

So, if the total time to execute the two modules is t and f1 takes 'x' units of time to execute then f2 should take (t-x) time to execute.

And the probability density function of time taken to execute the two modules is given by their convolution.

$\therefore$ Overall time = $\int_{o}^{t} f_1(x)f_2(t-x)dx$

(C) is the answer.

61. In the best case, (i.e. when the input is sorted) the number of inversions is 0.

In the worst case, (i.e. when the input is reverse-sorted) the number of inversions is

$\frac{n(n-1)}{2}$

So, average number of inversions $= \frac{0+\frac{n(n-1)}{2}}{2}$

$= \frac{n(n-1)}{4}$

(B) is the answer.

62. In the worst case, the number of inversions is $\frac{n(n-1)}{2}$ i.e. $O(n^2)$.

But it is given that the number of inversions is at most n.

So the complexity of the insertion sort would reduce from $O(n^2)$ to $O(n)$.

(D) is the answer.

63. • For a heap with n elements, height =O(logn)

i) Deletion of smallest element takes O(logn) time

ii) Insertion of an element if it is not already present will take time O(n) + O(logn)

O(n) $\rightarrow$ for finding if the element is already present.

O(logn) $\rightarrow$ for inserting the element.

• For a balanced binary search tree, height = O(logn)

i) Deletion of smallest element will take O(logn) time. {Smallest element will be present at the left most node. So we just need to traverse the height}

ii) Insertion of element if it is not present in the tree will take O(logn) time.

So, a balanced binary search tree can be used but not a heap.

(B) is the answer.

64. Eash Push and Pop operation takes X seconds each. The time elapse between the end of one such stack operation and the start of the next operation = Y sec.

stack life of last element (T_n) = Y (as it will be popped off as soon as pushed on the stack)

Stack life of second last element (T_{n-1}) $= T_n + 2X + 2Y$

$= 2X + 3Y$

Similarly $T_{n-2} = T_n + 2X + 2Y$

$= 4X + 5Y$

$\therefore$ Stack life of 1st element $= 2(n-1)x + (2n-1)Y$

Sum of all such stack lives $= \sum_{i-1}^{n} 2(n-1)\,x + \sum(2n-1)Y$

$$= \cancel{2} \times \frac{n(n-1)}{\cancel{2}} + \cancel{2}Y\frac{n(n+1)}{\cancel{2}} - nY$$

$$= n(n-1)x + n(n+1)Y - nY$$

$\therefore$ Average of n stack lives $= \dfrac{\text{Sum}}{n}$

$= (n-1)\,X + (n+1)Y - Y$

$= nX - X + nY + \cancel{Y} - \cancel{Y}$

$= n(X+Y) - X$

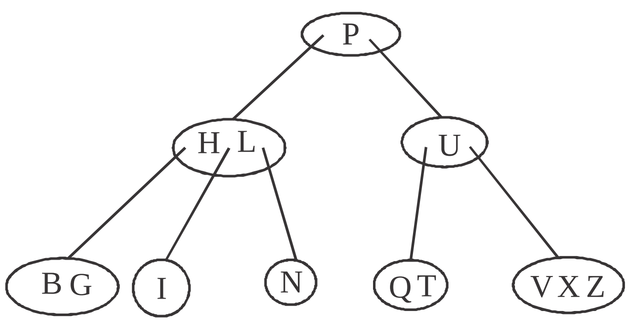
(B) is the answer.

65. Here the maximum degree of a B-tree is 4 i.e a node can have a maximum of 3 keys.

When we insert G in the given tree, the node in which it is inserted becomes full. So we need to split it and send one of its element to the root. But the root is full too. So we now split the root node as well.

So inserting G in the node incurs 1 leaf node split + 1 root node split.

The resultant tree is :



(B) is the answer.

66. To answer the given question, we need to find the upper and lower bounds on the complexity of finding an integer cube root m of n.

Cube root can be found in not more than O(logn) using binary search. So we can eliminate the options (A) and (B).

Also note that the input n is encoded in binary and hence input size is O(logn) because the minimum number of bits needed to represent an arbitrary n in binary notation is proportional to logn. All the bits in the number must be processed at least one. So θ (logn) is the lower bound on the time to solve the problem and therefore the problem cannot be solved in $O((\log \log n)^w)$ time (where w > 0).

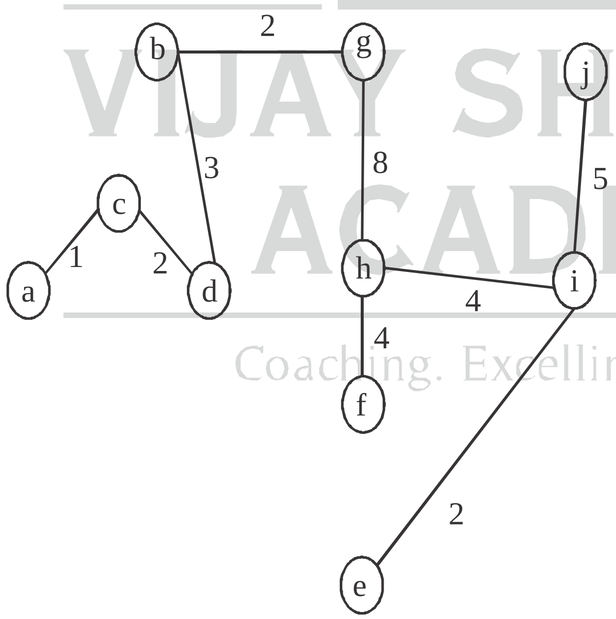
So (C) is the answer.

67. When the shortest path is computed from v_1(one of the vertices in V_1) to every other vertex in G_1, G_1 is connected if all these costs are 0 as edge weights of subgraph G_1 is 0 and those should be included in the shortest path.

If the cost is not 0 to at least one vertex in G_1 (and not G), then G_1 is disconnected.

(B) is the answer.

68. The minimum spanning tree of the give graph is :

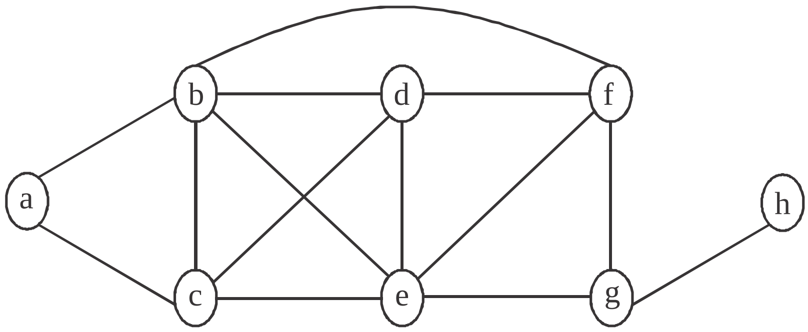


Weight of the MST = 1 + 2 + 3 + 2 + 8 + 4 + 4 + 5 + 2
= 31

(B) is the answer.

69. The given problem is same as activity selection problem. It can also be thought of as a graph theory problem. Here we construct a graph with one node to corresponding to each activity from A to H. So we need 8 nodes.

Now we connect those nodes that occur between the start time and end time of an activity. And the chromatic number of the graph is the number of rooms required.



The chromatic number of the above graph turns out to be 4.

(B) is the answer.

70. Here A [j, k] gives the count of maximum edges on a path from j to k.

The expression

A[j, k] = max (A[j, k], (A[j, i] + A[i, k]))

calculates the count of maximum edges by considering every possible vertex k which can become an intermediate vertex and can give longer path.

(D) is the answer.

71. (C) is the answer

72. Consider option (A)

$\left((P \vee R) \wedge (Q \vee \neg R)\right) \rightarrow (P \vee Q)$

$= (P+R)(Q+\overline{R}) \rightarrow (P+Q)$

$= \overline{(P+R)(Q+\overline{R})} + (P+Q)$

$= \overline{(P+R)} + \overline{Q+R} + (P+Q)$

$= \overline{P}\overline{R} + \overline{Q}R + P + Q$ $\quad \left[\because P + \overline{P}\overline{R} = P + \overline{R}\right]$

$= P + \overline{R} + Q + R$ $\quad \left[\because R + \overline{R} = \text{True}\right]$

$=$ True

Hence it is logically valid

So, (A) is true.

Consider option (B)

$(P \vee Q) \rightarrow \left((P \vee R)(Q \wedge \neg R)\right)$

$= (P+Q) \rightarrow \left((P+R)\left(Q+\bar{R}\right)\right)$

$= \overline{(P+Q)} + (P+R)\left(Q+\bar{R}\right)$

$= \bar{P}\bar{Q} + PQ + P\bar{R} + QR + R.\bar{R}$

$= \bar{P}\bar{Q} + PQ + P\bar{R} + QR$

The above formula does not always evaluate to true.
Hence it is logically invalid.
So, (B) is the answer.

73. global int i=100, j=5;

```
void P(x)
{     int i =10;               // Here a new variable i is created and the value 10 is stored in it.
      print (x + 10);          // print (i + j + 10) = 10 + 5 + 10 = 25. So 25 will be printed.

      // call by need for i + j. So i + j is replaced by 15 everywhere
      i = 200;                 // Local value of i is changed to 200
      j = 20;                  //global value of j is changed to 20 as there is no local j.
      print (x);               // print (i + j) = 10 + 5 = 15. So 15 will be printed
  }
  main ()
  {
      P(i + j);                // Here i and j refer to global values i.e. 100 and 5.
  }
```

In call by name, the argument is evaluated every time it is used, whereas in call by need, it is evaluated the first time it is used and the value is recorded so that it need not be re-evaluated.
(C) is the answer.

74. Using call by name,

```
global int i = 100, j = 5;
void P(x)
{
int  i = 10;             // Here a new variable i is created and the value 10 is stored in it.
print (x + 10);          // print (i + j + 10) = 10 + 5 + 10 = 25. So, 25 is printed.
i = 200;                 // Local value of i and global value of j is
j = 20;                     changed to 200 and 20 respectively.
print (x);               // print (i + j) = 200 + 20 = 220. So, 220 is printed.
}
main ( )
{
P(i + j);                // Since it is call by name, we don't calculate its value here. i and j refer to
                         global values.
}
```

(B) is the answer.

75. The concept here is polymorphism. When a name denotes instances of many different classes related by some common supreclass, in object-oriented programming this is referred to as polymorphsim.

Here, the function calls are resolved at runtime. It does not depend upon the type of pointer used for calling purpose. It depends only on the fact that which object this pointer is pointing to.

In the given code, since all pointers point to derived class objects so only derived class function will be called and 2 will be printed thrice.

(D) in the answer.

76. (A) is true because DLLs (dynamically linked libraries) are separated from executable and hence, the size of executable becomes smaller.

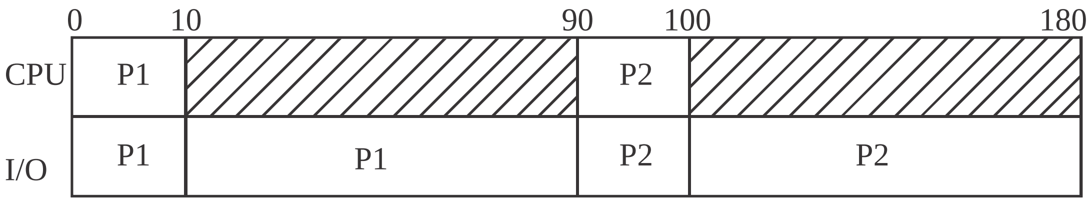
(B) is true. DLLs are shared among multiple executable programs. So, the total memory of the system is wasted which eventually decreases the page fault rate.

(C) is false. In DLL linking is postponed until execution time. It takes place during program runtime. Also DLL takes more time to load and link the required libraries.

(D) is true. There is no need to explicitly relink the DLLs to existing programs i.e. if a DLL is replaced to a new version, it will automatically get linked during the runtime.

(C) is the answer.

77. For FCFS and SRTF,



So in 100 ms, CPU utilization $= \frac{20}{100} \times 100$

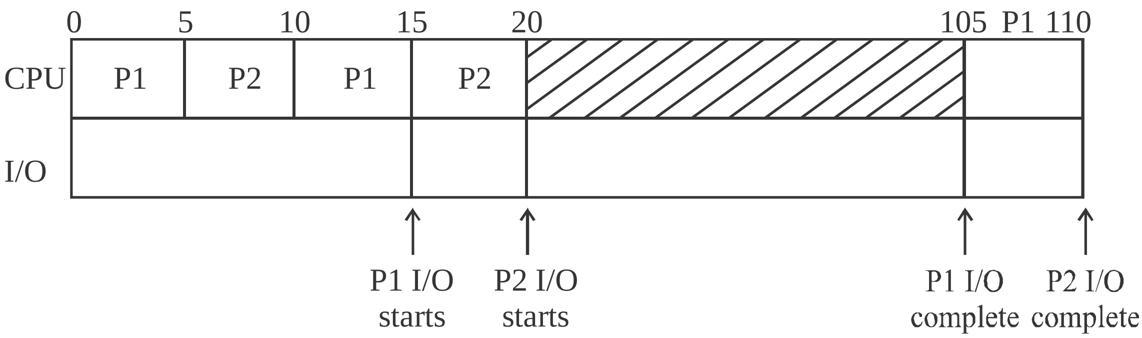
$= 20\%$

For static priority scheduling,

Suppose process 1 is having higher priority, then the scheduling sequence will be same as FCFS.

Now suppose process 2 is having higher priority, then the scheduling sequence will be same as FCFS but with processes 1 and 2 interchanged. Hence, the CPU utilization remains 20%.

For round robin scheduling with time quantum q = 5



So CPU utilization $= \dfrac{20}{105} \times 100$

$= 19.047\%$

So, round robin with time quantum q = 5 ms results in least CPU utilization.

(D) is the answer.

78. Average memory access time

= TLB Hit * Cache Hit + TLB Hit * Cache Miss + [TLB Miss * Cache Hit + TLB Miss * Cache Miss

= 0.96*0.9(1+1) + 0.96*0.1(1+1+10) + 0.04*0.9(1+1+10+10) + 0.04*0.1 (1+1+10+10+10)

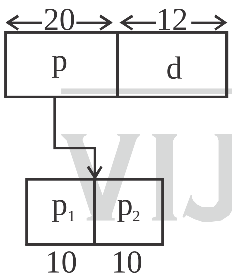
= 3.8

$\approx$ 4 ns

Note that here page table access time is same as main memory access time as both the tables (of 2-level paging) are stored in main memory.

(D) is the answer.

79. Virtual address = 32 bits



Both first and second level page tables require 10 bits for addressing the page tables. And each page table entry is of 32 bits i.e. 4 bytes.

$\therefore$ Size of first level page table = 4B * 2^{10}

= 4KB

The given process has two contiguous code pages starting at virtual address 0×00000000

$\therefore$ Address of first code page :

0×00000000 = 0000 0000 00 | 0000 0000 00 | 0000 0000 0000

Address of second code page :

0×00001000 = 0000 0000 00 | 00 0000 0001 | 0000 0000 0000

Also the process has two contiguous data pages starting at virtual address 0×00400000

$\therefore$ Address of first data page :

0×00400000 = 0000 0000 01 | 00 0000 0000 | 0000 0000 0000

Address of second data page :

0×00401000 = 0000 0000 01 | 00 0000 0001 | 0000 0000 0000

The process also has one stack page starting at virtual address $0 \times$ FFFFF000

$\therefore$ Address of stack page :

$0 \times$ FFFFF000 = 1111 1111 11 | 11 1111 1111 | 0000 0000 0000

So for second level page tables, we require 3 distinct addressess which are 0000 0000 00, 0000 0000 01, 1111 1111 11. Each of these second level page table entries will be addressed to first level page table using 10 bits.

So size of second level page table.

$= 3 \times 4$ KB

$= 12$ KB

$\therefore$ Total memory required to store the page table of this process = 4 KB + 12 KB

(C) is the answer.

80. To get the given pattern, Process P should be executed before Process Q. To ensure that Process Q does not start its execution prior to Process P, the initial value of T should be 0.

So,we can easily eliminate options (A) and (C).

Now consider option (B),

S = 1 and T = 0

W : P(S) S = 0 Y : P(T) T = 0 ($\therefore$ blocked until T = 1)

X : V(T) T = 1 Z : V(S) S = 1

This sequence of operations will give us the desired output.

(B) is the answer.

81. We have to ensure that the output string never contains a substring of the form 01^n0 and 10^n1 where n is odd.

Consider option (A),

S = T = 1 initially

W : P(S) S = 0 Y : P(T) T = 0

X : V(S) Z : V (T)

So, the above sequence will print 0s and 1s in interleaved manner like 010001001011101...... and hence the output will definitely contain substrings of the form 01^n0 and 10^n1 where n is odd.

Consider option (B)

S = T = 1 initially

W : P(S) S = 0 Y : P(T) T = 0

X : V(T) T = 1 Z : V(S) S = 1

Even the above sequence will print 0s and 1s in any order.

Consider option (c),

S = 1 initally

W : P(S) S=0 Y : P (S) ($\therefore$ blocked until S = 1)

X : V (S) Z : V (S)

So, the above sequence will print either 00110011... or 1100110011...

It won't print any string wherein the substrings are of the form 01^n0 or 10^n1 where n is odd.

(C) is the answer.

82. Subnet mask of a network = 255.255.31.0

(A)

$$\begin{array}{r} 172.57.88.62 \\ \wedge\, 255.255.31.0 \\ \hline 172.57.24.0 \\ \hline \end{array} \qquad \begin{array}{r} 172.56.87.233 \\ \wedge\, 255.255.31.0 \\ \hline 172.56.23.0 \\ \hline \end{array}$$

Clearly both belong to different networks.

(B)

$$\begin{array}{r} 10.35.28.2 \\ \wedge\, 255.255.31.0 \\ \hline 10.35.28.0 \\ \hline \end{array} \qquad \begin{array}{r} 10.35.29.4 \\ \wedge\, 255.255.31.0 \\ \hline 10.35.29.0 \\ \hline \end{array}$$

Both IP addresses belong to different networks.

(C)

$$\begin{array}{r} 191.203.31.87 \\ \wedge\;\; 255.255.31.0 \\ \hline 191.203.31.0 \\ \hline \end{array} \qquad \begin{array}{r} 191.234.31.88 \\ \wedge\;\; 255.255.31.0 \\ \hline 191.234.31.0 \\ \hline \end{array}$$

Even here, both the IP addresses belong to different networks.

(D)

$$\begin{array}{r} 128.8.129.43 \\ \wedge\; 255.255.31.0 \\ \hline 128.8.1.0 \\ \hline \end{array} \qquad \begin{array}{r} 128.8.161.55 \\ \wedge\; 255.255.31.0 \\ \hline 128.8.1.0 \\ \hline \end{array}$$

Here both the addresses match, so both belong to the same network.

(D) is the answer.

83. $d = 2\text{km} = 2 \times 10^3\,\text{m}$

$BW = 10^7$ bps

$v = 2 \times 10^8$ m/s

For CSMA / CD,

$$T_t \geq 2P_t$$

$$\therefore \quad \frac{L}{BW} \geq 2 \times \frac{d}{v}$$

$$\therefore \quad L \geq 2 \times \frac{d}{v} \times BW$$

$$\therefore \quad L \geq 2 \times \frac{2 \times 10^3\,\text{m}}{2 \times 10^8\,\text{m/s}} \times 10^7\,\text{bps}$$

$$\therefore \quad L \geq 200 \text{ bits}$$

$$\therefore \quad L \geq 25 \text{ bytes}$$

(D) is the answer.

84. L = 1000 B

$T_t = 50\ \mu s$

$P_t = 200\ \mu s$

$$\text{Throughput} = \text{Efficiency} \times \text{Bandwidth}$$

$$= \frac{WS \times T_t}{T_t + 2P_t} \times BW$$

$$= \frac{WS \times L / BW}{T_t + 2P_t} \times BW$$

$$= \frac{WS \times L}{T_t + 2P_t}$$

$$= \frac{5 \times 1000 \times 8\ \ b}{(50 + 400) \times 10^{-6} s}$$

$$= 88.88 \times 10^6 \text{ bits per second}$$

$$= 11.11 \times 10^6 \text{ bytes per second}$$

(B) is the answer.

85. For our simplicity, let

D – Date_of_Birth

A – Age

E – Eligibility

N – name

R – Roll_number

C – Course_name

Cn – Course_name

G – Grade

For the relation (D, A, R, N)

$D \to A$

$N \to R$

$R \to N$

$(DN)^+ = \{D, A, R, N\}$

So, DN is a candidate key

Also $R \to N$

$\therefore DR \to DN$

So, DR is also a candidate key

$\therefore$ Prime attributes = {D, N, R}

We have

Proper subset of candidate key $\longrightarrow$ Non - key attribute

as $D \rightarrow A$

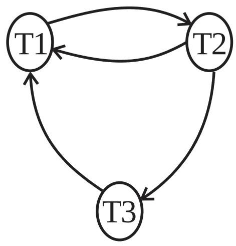
So, the relation is not in 2NF.

(D) is the answer.

86. The query gives the names of students who have scored an A grade in at least one of the courses taught by Korth.

(C) is the answer.

87. The precedence graph of the given schedule is



So, there is a cycle in the precedence graph and hence the schedule is not serilizable.

(D) is the answer.

88. Nothing will be printed by this program fragment.

The given code makes all the entries from A[3] to A[n] as 1 and hence the if condition (!A[j]) is always 0 and the printf statement inside it is never executed.

So, it prints an empty set.

(D) is the answer.

89. In main :

x = 5 //global variable x takes value as 5

P(&x); // address of x is passed to p.

In function P (int*y) :

{

int x = *y + 2 // x = 5 + 2 = 7

Q(x); //Q(7). Here x is passed by value. Now control switches to funtion Q.

* y = x − 1; // * y = 7 – 1 = 6 i.e. global value of x becomes 6

print (x) ; //Local value of x is printed i.e. 7 and control switches to main function

}

In function Q (int z) :

z + = x; // Local value of z becomes z = 7 + 5 = 12

print (z) ; // 12 is printed. And control switches back to function P.

Now in main :

print (x); // global value of x is printed i.e. 6

So the output is 12 7 6

(A) is the answer.

90. The given code returns 1 if :

i) the list is empty

ii) the list has exactly one element

iii) the elements in the list are sorted in non-decreasing order of data value.

option (A) is true but "if and only if " makes it false.

(B) covers all the 3 cases mentioned above.

So, (B) is the answer.