IAE ACADEMY INSTITUTE FOR ENGINEERS

GATE CS · 2024 · SET 1

IAE Academy free study material · Updated 27 October 2026

GATE CS 2024 Set 1 Solved Paper

INSIDE

- All 65 questions, in the order of the paper
- The official answer to every question
- Step-by-step worked solutions
- The answer key on one page

Question paper and official answer key: IISc Bengaluru

Online, with the questions linked to their topic notes: **iaeacademy.in/study/papers/cs-2024-1**

FREE COUNSELLING

Preparing for GATE CS? Talk to IAE Academy.

Or call **+91 84204 36524**

Every day, 10:30 am to 7:30 pm

Chat on WhatsApp

Free. Not for sale. Pass it on.

About this paper

65 questions, 100 marks. Question types: 29 MCQ · 20 MSQ · 16 NAT.

Official answer key: IISc Bengaluru · https://gate2024.iisc.ac.in/wp-content/uploads/2024/CS1FinalAnswerKey.pdf

We placed each question under a subject and topic ourselves; the exam paper does not label them.

This paper online, with the questions linked to their topic notes:
iaeacademy.in/study/papers/cs-2024-1

Answer key

Q	TYPE	MARKS	ANSWER
Q.1	MCQ	1	**A**
Q.2	MCQ	1	**D**
Q.3	MCQ	1	**A**
Q.4	MCQ	1	**C**
Q.5	MCQ	1	**A**
Q.6	MCQ	2	**C**
Q.7	MCQ	2	**A**
Q.8	MCQ	2	**C**
Q.9	MCQ	2	**A**
Q.10	MCQ	2	**A**
Q.11	MCQ	1	**D**
Q.12	MCQ	1	**B**
Q.13	MCQ	1	**B**
Q.14	MCQ	1	**B**
Q.15	MCQ	1	**C**
Q.16	MCQ	1	**C**
Q.17	MCQ	1	**A**
Q.18	MCQ	1	**C**
Q.19	MCQ	1	**C**
Q.20	MCQ	1	**D**
Q.21	MCQ	1	**A**
Q.22	MSQ	1	**A, C**
Q.23	MSQ	1	**C, D**
Q.24	MSQ	1	**D**
Q.25	MSQ	1	**B, C**
Q.26	MSQ	1	**A, D**
Q.27	MSQ	1	**B, C**
Q.28	MSQ	1	**A, C**
Q.29	MSQ	1	**A, D**
Q.30	MSQ	1	**A, B, C**
Q.31	MSQ	1	**A, C**
Q.32	NAT	1	**2**
Q.33	NAT	1	**6**
Q.34	NAT	1	**16**
Q.35	NAT	1	**2**
Q.36	MCQ	2	**B**
Q.37	MCQ	2	**A**
Q.38	MCQ	2	**A**
Q.39	MCQ	2	**D**
Q.40	MCQ	2	**A**
Q.41	MCQ	2	**D**
Q.42	MCQ	2	**A**
Q.43	MCQ	2	**A**
Q.44	MSQ	2	**B, C, D**
Q.45	MSQ	2	**C**
Q.46	MSQ	2	**B, C, D**
Q.47	MSQ	2	**A, B**
Q.48	MSQ	2	**B, D**
Q.49	MSQ	2	**A**
Q.50	MSQ	2	**B, C**
Q.51	MSQ	2	**B, C**
Q.52	MSQ	2	**B, D**
Q.53	MSQ	2	**B, C**
Q.54	NAT	2	**4096**
Q.55	NAT	2	**3**
Q.56	NAT	2	**3.00**
Q.57	NAT	2	**14**
Q.58	NAT	2	**40**
Q.59	NAT	2	**179**
Q.60	NAT	2	**60**
Q.61	NAT	2	**44**
Q.62	NAT	2	**6596**
Q.63	NAT	2	**0.370 to 0.380**
Q.64	NAT	2	**4**
Q.65	NAT	2	**6**

General Aptitude Q.1 to Q.10

Q.1 MCQ · 1 MARK

General Aptitude › Reasoning

If '→' denotes increasing order of intensity, then the meaning of the words [dry → arid → parched] is analogous to [diet → fast → _____].
Which one of the given options is appropriate to fill the blank?

(A) starve

(B) reject

(C) feast

(D) deny

Official answer: **A**

SOLUTION

1. The arrow means increasing intensity: each word in the chain is a stronger form of the same idea as the word before it.
2. Check the given chain: dry, then arid (very dry), then parched (extremely dry). The idea stays the same (lack of moisture) and its degree rises.
3. Now read [diet → fast]. Both are about limiting food intake. A diet is mild restriction, and a fast is a stronger, deliberate abstention from food.
4. The blank must be the next stronger degree of the same idea, which is going without food to the point of suffering.
5. Test the options. 'starve' means to suffer or die from lack of food, the extreme end of food deprivation, so it fits.
6. 'reject' and 'deny' are not about food intake. 'feast' is the opposite idea, since it means eating lavishly, so it breaks the pattern.
7. Therefore the blank is 'starve', which is option (A).

KEY IDEA

In an intensity chain, find the shared idea (here, restricting food) and pick the option that is the next, stronger degree of that same idea. Diet, then fast, then starve.

WATCH OUT FOR

Picking 'deny' or 'reject' because they sound like refusing something. They are not about food and are not a stronger degree of fasting. Picking 'feast' ignores that the direction is increasing restriction.

Q.2 MCQ · 1 MARK

General Aptitude › Numerical

If two distinct non-zero real variables x and y are such that $(x+y)$ is proportional to $(x-y)$ then the value of $\frac{x}{y}$

(A) depends on xy

(B) depends only on x and not on y

(C) depends only on y and not on x

(D) is a constant

Official answer: **D**

SOLUTION

1. Since $x \neq y$, $(x-y) \neq 0$. "Proportional" means $x+y = k(x-y)$ for some constant k.
2. Expand: $x+y = kx-ky$.

3. Collect the x terms on one side and the y terms on the other: $x - kx = -ky - y$, so $x(1-k) = -y(1+k)$.
4. Here $k \neq 1$. If $k = 1$, the equation $x + y = x - y$ gives $y = 0$, which is not allowed.
5. Divide both sides by $y(1-k)$, which is non-zero:
$$\frac{x}{y} = -\frac{1+k}{1-k} = \frac{k+1}{k-1}$$
6. The right side contains only the constant k. So $\frac{x}{y}$ is a constant and does not depend on x, y or xy. The answer is (D).

KEY IDEA
Turn "proportional" into an equation with a constant k, then solve for x/y. Only k remains on the right, so the ratio is fixed.

WATCH OUT FOR
Treating x and y as independent and picking an option that says the ratio depends on one of them. Forgetting that k is a fixed constant, so the ratio cannot change with x or y.

Q.3 MCQ · 1 MARK

General Aptitude › Statistics & probability

Consider the following sample of numbers:

9, 18, 11, 14, 15, 17, 10, 69, 11, 13

The median of the sample is

(A) 13.5
(B) 14
(C) 11
(D) 18.7

Official answer: **A**

SOLUTION

1. The median is the middle value of the data once it is arranged in order. First sort the 10 numbers in ascending order.
2. Sorted: $9, 10, 11, 11, 13, 14, 15, 17, 18, 69$.
3. There are $n = 10$ values, an even count, so no single middle value exists. The median is the mean of the $\frac{n}{2} = 5$th and $\left(\frac{n}{2}+1\right) = 6$th values.
4. The 5th value is 13 and the 6th value is 14.
5. $$\text{Median} = \frac{13+14}{2} = 13.5$$
6. The outlier 69 does not change the median. It does pull the mean up: the sum is $9+10+11+11+13+14+15+17+18+69 = 187$, so the mean is $\frac{187}{10} = 18.7$. Option (D) is this mean, not the median.
7. Hence the median is 13.5, option (A).

KEY IDEA
Sort the data, then for an even count take the mean of the two middle values. The median ignores how extreme the largest value is.

WATCH OUT FOR
Picking the mean (18.7, option D) because the outlier 69 inflates it. Another slip is taking the 5th or 6th value alone (13 or 14) instead of averaging the two middle values for an even count. Forgetting to sort first also gives wrong answers.

Q.4 MCQ · 1 MARK

General Aptitude › Numerical

The number of coins of Rs. 1, Rs. 5, and Rs. 10 denominations that a person has are in the ratio 5:3:13. Of the total amount, the percentage of money in Rs. 5 coins is

(A) 21%

(B) 14 2/7 %

(C) 10%

(D) 30%

Official answer: **C**

SOLUTION

1. Let the numbers of coins of Rs. 1, Rs. 5 and Rs. 10 be $5k$, $3k$ and $13k$, since they are in the ratio $5 : 3 : 13$.
2. Money in each denomination = number of coins $\times$ face value, so the amounts are $5k \times 1 = 5k$, $3k \times 5 = 15k$ and $13k \times 10 = 130k$ (in Rs.).
3. Total amount $= 5k + 15k + 130k = 150k$.
4. Share of the Rs. 5 coins $= \dfrac{15k}{150k} \times 100\% = 10\%$.
5. Hence the answer is option (C).

KEY IDEA

A ratio of coin counts is not a ratio of money. Multiply each count by its face value first, then take the share of the total.

WATCH OUT FOR

Using the count ratio directly, 3/(5+3+13) = 3/21, which is about 14.3% (option B). This ignores that the coins have different face values.

Q.5 MCQ · 1 MARK

General Aptitude › Numerical

For positive non-zero real variables p and q, if

$$\log(p^2 + q^2) = \log p + \log q + 2\log 3,$$

then, the value of $\dfrac{p^4 + q^4}{p^2 q^2}$ is

(A) 79

(B) 81

(C) 9

(D) 83

Official answer: **A**

SOLUTION

1. Combine the logarithms on the right: $\log p + \log q + 2\log 3 = \log(pq) + \log 9 = \log(9pq)$.
2. The logarithm is one-to-one, so $\log(p^2 + q^2) = \log(9pq)$ gives $p^2 + q^2 = 9pq$.
3. Rewrite the target using $a^2 + b^2 = (a+b)^2 - 2ab$ with $a = p^2$, $b = q^2$:
$$\frac{p^4 + q^4}{p^2 q^2} = \frac{(p^2 + q^2)^2 - 2p^2 q^2}{p^2 q^2}.$$
4. Substitute $p^2 + q^2 = 9pq$:
$$\frac{81p^2 q^2 - 2p^2 q^2}{p^2 q^2}.$$
5. Cancel $p^2 q^2$, which is non-zero because p and q are positive: $81 - 2 = 79$. The answer is option (A).

KEY IDEA
Turn the log equation into $p^2 + q^2 = 9pq$, then write $p^4 + q^4 = (p^2 + q^2)^2 - 2p^2q^2$ so everything is in terms of pq.

WATCH OUT FOR
Writing $p^4 + q^4 = (p^2 + q^2)^2$ and forgetting the $-2p^2q^2$ term, which gives 81 (option B). Another slip is reading $2\log 3$ as $\log 6$ instead of $\log 9$.

Q.6 MCQ · 2 MARKS

General Aptitude › Grammar & vocabulary

In the given text, the blanks are numbered (i)–(iv). Select the best match for all the blanks.

Steve was advised to keep his head (i) _____ before heading (ii) _____ to bat; for, while he had a head (iii) _____ batting, he could only do so with a cool head (iv) _____ his shoulders.

(A) (i) down (ii) down (iii) on (iv) for

(B) (i) on (ii) down (iii) for (iv) on

(C) (i) down (ii) out (iii) for (iv) on

(D) (i) on (ii) out (iii) on (iv) for

Official answer: **C**

SOLUTION

1. Blank (i): the idiom is "keep one's head down", meaning to stay focused and avoid attention. "Keep his head on" is not an idiom. So (i) = down, which eliminates B and D.
2. Blank (ii): "heading out to bat" is the standard phrase for going out onto the field to bat. "Heading down to bat" is not idiomatic. So (ii) = out, which eliminates A.
3. Blank (iii): "have a head for something" means to have a natural aptitude for it, so "a head for batting". So (iii) = for.
4. Blank (iv): "a cool head on his shoulders" is the fixed phrase. So (iv) = on.
5. Check option C: down, out, for, on. The text reads: "Steve was advised to keep his head down before heading out to bat; for, while he had a head for batting, he could only do so with a cool head on his shoulders." Every blank fits.

KEY IDEA
Fit each blank to a fixed idiom: keep one's head down, heading out to bat, a head for something, a cool head on one's shoulders. Eliminate options blank by blank.

WATCH OUT FOR
Choosing "on" for blank (i) or "down" for blank (ii) because the words sound natural next to "head" or "bat". Only "keep his head down" and "heading out to bat" are standard.

Q.7 MCQ · 2 MARKS

General Aptitude › Geometry

A rectangular paper sheet of dimensions 54 cm × 4 cm is taken. The two longer edges of the sheet are joined together to create a cylindrical tube. A cube whose surface area is equal to the area of the sheet is also taken.

Then, the ratio of the volume of the cylindrical tube to the volume of the cube is

(A) 1/π

(B) 2/π

(C) 3/π

(D) 4/π

Official answer: **A**

SOLUTION

1. Joining the two longer (54 cm) edges makes the 54 cm side the height of the tube, so $h = 54\ \text{cm}$. The circumference is the shorter side, $2\pi r = 4\ \text{cm}$.
2. Radius of the tube: $r = \dfrac{4}{2\pi} = \dfrac{2}{\pi}\ \text{cm}$.
3. Volume of the cylinder: $V_c = \pi r^2 h = \pi \cdot \dfrac{4}{\pi^2} \cdot 54 = \dfrac{216}{\pi}\ \text{cm}^3$.
4. Area of the sheet: $54 \times 4 = 216\ \text{cm}^2$. The cube's surface area is $6a^2 = 216$, so $a^2 = 36$ and $a = 6\ \text{cm}$.
5. Volume of the cube: $V_k = a^3 = 6^3 = 216\ \text{cm}^3$.
6. Ratio: $\dfrac{V_c}{V_k} = \dfrac{216/\pi}{216} = \dfrac{1}{\pi}$, which is option (A).

KEY IDEA

Rolling a rectangle into a tube turns one side into the circumference and the other into the height. Joining the two longer edges makes the long side the height and the short side the circumference. The sheet's area is the same as the cube's total surface area.

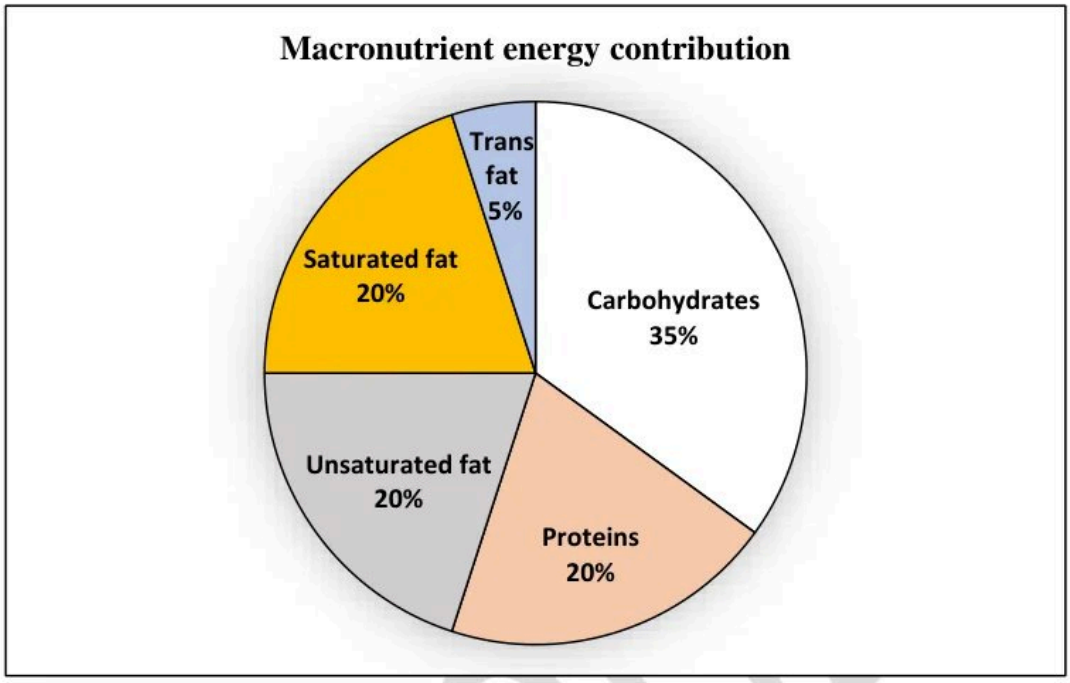
WATCH OUT FOR

Taking the 54 cm side as the circumference, which gives $r = 27/\pi$ and a wrong ratio. Another slip is equating the cube's volume, not its surface area, to the sheet's area, or using $4a^2$ in place of $6a^2$.

Q.8 MCQ · 2 MARKS

General Aptitude › Data interpretation

Q.8 The pie chart presents the percentage contribution of different macronutrients to a typical 2,000 kcal diet of a person.



The typical energy density (kcal/g) of these macronutrients is given in the table.

Macronutrient	Energy density (kcal/g)
Carbohydrates	4
Proteins	4
Unsaturated fat	9
Saturated fat	9
Trans fat	9

The total fat (all three types), in grams, this person consumes is

(A) 44.4

(B) 77.8

(C) 100

(D) 3,600

Official answer: **C**

SOLUTION

1. Read the chart: the three fat types are unsaturated fat 20%, saturated fat 20% and trans fat 5%, so fat supplies $20 + 20 + 5 = 45\%$ of the energy.
2. Energy from fat $= 0.45 \times 2000 = 900$ kcal.
3. All three fat types have the same energy density, 9 kcal/g, so they can be treated together.

4. Mass of fat $= \dfrac{\text{energy}}{\text{energy density}} = \dfrac{900\ \text{kcal}}{9\ \text{kcal/g}} = 100\ \text{g}$.
5. The answer is 100 g, option (C).

KEY IDEA
Add the percentage shares of all three fat types, convert to kcal using the 2,000 kcal total, then divide by the energy density (kcal/g) to get grams.

WATCH OUT FOR
Using only the saturated and unsaturated fat shares (40%) and leaving out the 5% trans fat, which gives 800/9 = 88.9 g (not an option). Another slip is multiplying 2,000 by 1.8 or dividing by 4 instead of 9, giving a wrong option such as 44.4 or 77.8.

Q.9 MCQ · 2 MARKS

General Aptitude › Spatial

A rectangular paper of 20 cm × 8 cm is folded 3 times. Each fold is made along the line of symmetry, which is perpendicular to its long edge. The perimeter of the final folded sheet (in cm) is

(A) 18
(B) 24
(C) 20
(D) 21

Official answer: **A**

SOLUTION

1. Each fold is made along the line of symmetry perpendicular to the long edge. It halves the long edge and leaves the short edge unchanged.
2. Fold 1: the long edge is $20\ \text{cm}$, so it becomes $10\ \text{cm}$. The sheet is $10\ \text{cm} \times 8\ \text{cm}$.
3. Fold 2: the long edge is now $10\ \text{cm}$, so it becomes $5\ \text{cm}$. The sheet is $5\ \text{cm} \times 8\ \text{cm}$.
4. After fold 2 the $8\ \text{cm}$ side is the longer one. Fold 3 is perpendicular to it, so $8\ \text{cm}$ becomes $4\ \text{cm}$. The sheet is $5\ \text{cm} \times 4\ \text{cm}$.
5. Perimeter $= 2(l + b) = 2(5 + 4) = 18\ \text{cm}$, which is option (A).

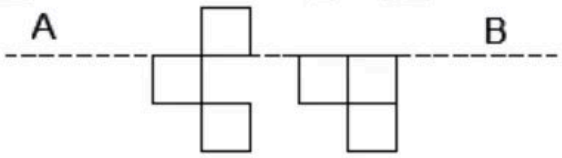
KEY IDEA
Each fold halves whichever edge is currently the longer one. Re-identify the long edge after every fold, because the sheet's orientation can change.

WATCH OUT FOR
Halving the original 20 cm edge three times (20 to 2.5 cm, giving 2(2.5+8) = 21), or forgetting that the 8 cm side becomes the long edge after fold 2.

Q.10 MCQ · 2 MARKS

General Aptitude › Spatial

Q.10 The least number of squares to be added in the figure to make AB a line of symmetry is



(A) 6

(B) 4

(C) 5

(D) 7

Official answer: **A**

Computer Science and Information Technology Q.11 to Q.65

Q.11 MCQ · 1 MARK

Engineering Mathematics › Calculus

Let $f : \mathbb{R} \to \mathbb{R}$ be a function such that $f(x) = \max\{x, x^3\}$, $x \in \mathbb{R}$, where $\mathbb{R}$ is the set of all real numbers. The set of all points where $f(x)$ is NOT differentiable is

(A) $\{-1, 1, 2\}$

(B) $\{-2, -1, 1\}$

(C) $\{0, 1\}$

(D) $\{-1, 0, 1\}$

Official answer: **D**

SOLUTION

1. $f(x) = \max\{x, x^3\}$ is the larger of two polynomials. Where one is strictly larger on an open interval, f equals that polynomial and is differentiable. So only points where the two are equal can fail.
2. Equal values: $x = x^3 \Rightarrow x(x-1)(x+1) = 0$, so $x \in \{-1, 0, 1\}$.
3. Sign of $x^3 - x = x(x-1)(x+1)$ on each interval: negative for $x < -1$, so $f = x$; positive for $-1 < x < 0$, so $f = x^3$; negative for $0 < x < 1$, so $f = x$; positive for $x > 1$, so $f = x^3$. The branch changes at each of the three points.
4. At $x = -1$: left derivative $= \frac{d}{dx}x = 1$, right derivative $= 3x^2 = 3$. They differ, so f is not differentiable there.
5. At $x = 0$: left derivative $= 3x^2 = 0$ (branch x^3 on $(-1, 0)$), right derivative $= 1$ (branch x on $(0, 1)$). They differ, so f is not differentiable there.
6. At $x = 1$: left derivative $= 1$ (branch x), right derivative $= 3x^2 = 3$ (branch x^3). They differ, so f is not differentiable there.
7. So f fails to be differentiable exactly on $\{-1, 0, 1\}$, which is option (D).

KEY IDEA

A maximum of two differentiable functions can fail to be differentiable only where they are equal. There, check whether the one-sided slopes agree; if they differ, the point is a corner.

WATCH OUT FOR

Dropping $x = 0$ because both x and x^3 are zero there and x^3 has zero slope. The slopes are 1 and 0, which differ, so 0 is a corner. Equal values at a meeting point do not make the slopes equal.

Q.12 MCQ · 1 MARK

Engineering Mathematics › Linear algebra

The product of all eigenvalues of the matrix $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ is

(A) -1

(B) 0

(C) 1

(D) 2

Official answer: **B**

SOLUTION

1. The product of all eigenvalues (counted with multiplicity) of a square matrix equals its determinant: $\prod \lambda_i = \det A$.
2. Compute $\det A = 1(5 \cdot 9 - 6 \cdot 8) - 2(4 \cdot 9 - 6 \cdot 7) + 3(4 \cdot 8 - 5 \cdot 7)$.

3. $= 1(45-48) - 2(36-42) + 3(32-35) = -3 + 12 - 9 = 0$.
4. Check: $R_3 - R_2 = R_2 - R_1 = (3,3,3)$, so the rows are linearly dependent and A is singular, which gives $\det A = 0$ and at least one zero eigenvalue.
5. Hence the product of the eigenvalues is 0, which is option (B).

KEY IDEA
The product of the eigenvalues of a square matrix equals its determinant, so you only need $\det A$. Here the rows are in arithmetic progression, so the matrix is singular.

WATCH OUT FOR
Trying to find the eigenvalues from the characteristic polynomial and making an arithmetic slip. The determinant gives the product directly. Another slip is a sign error in the cofactor expansion.

Q.13 MCQ · 1 MARK

Digital Electronics › Number systems

Consider a system that uses 5 bits for representing signed integers in 2's complement format. In this system, two integers A and B are represented as A=01010 and B=11010. Which one of the following operations will result in either an arithmetic overflow or an arithmetic underflow?

(A) A + B
(B) A − B
(C) B − A
(D) 2 * B

Official answer: **B**

SOLUTION

1. A 5-bit 2's complement number lies in $[-2^4, 2^4 - 1] = [-16, 15]$. A result outside this range is an overflow (above 15) or an underflow (below -16).
2. Decode $A = 01010$: the sign bit is 0, so $A = 8 + 2 = +10$.
3. Decode $B = 11010$: the sign bit has weight -16, so $B = -16 + 8 + 2 = -6$.
4. (A) $A + B = 10 + (-6) = 4$, which is inside $[-16, 15]$. No overflow.
5. (B) $A - B = 10 - (-6) = 16$, which is greater than 15 and cannot be held in 5 bits (it would wrap to $10000 = -16$). This is an arithmetic overflow.
6. (C) $B - A = -6 - 10 = -16$, which is exactly the minimum, 10000. It is representable, so there is no underflow.
7. (D) $2 \times B = -12$, which is inside the range. It is also representable, as 10100.
8. Only $A - B$ leaves the representable range, so the answer is (B).

KEY IDEA
Decode both operands to decimal, compute the exact result, and compare it with the range $[-2^{n-1}, 2^{n-1} - 1]$. For $n = 5$ that range is $[-16, 15]$.

WATCH OUT FOR
Reading $B = 11010$ as -10 by flipping the bits without adding 1, or as -26. Another slip is calling $B - A = -16$ an underflow, when -16 is the minimum value and is representable.

Q.14 MCQ · 1 MARK

Engineering Mathematics › Probability & statistics

Consider a permutation sampled uniformly at random from the set of all permutations of $\{1, 2, 3, \cdots, n\}$ for some $n \geq 4$. Let X be the event that 1 occurs before 2 in the permutation, and Y the event that 3 occurs before 4. Which one of the following statements is TRUE?

(A) The events X and Y are mutually exclusive
(B) The events X and Y are independent

(C) Either event X or Y must occur

(D) Event X is more likely than event Y

Official answer: **B**

SOLUTION

1. Swapping the positions of 1 and 2 in a permutation is a one-to-one map from permutations with 1 before 2 onto those with 2 before 1. So $P(X) = \frac{1}{2}$. The same argument with 3 and 4 gives $P(Y) = \frac{1}{2}$.
2. Option (D): $P(X) = P(Y) = \frac{1}{2}$, so X is not more likely than Y. (D) is false.
3. Option (A): since $n \geq 4$, the permutation $1, 2, 3, 4, \ldots$ has 1 before 2 and 3 before 4. Both events occur together, so $P(X \cap Y) > 0$ and the events are not mutually exclusive. (A) is false.
4. Option (C): the permutation $2, 1, 4, 3, \ldots$ has 2 before 1 and 4 before 3, so neither event occurs. It is therefore false that one of them must occur. (C) is false.
5. Option (B): only the relative order of the elements $1, 2, 3, 4$ matters for X and Y. In a uniformly random permutation, all $4! = 24$ relative orders of these four elements are equally likely. To get both X and Y, choose which 2 of the 4 positions hold $\{1, 2\}$ in $\binom{4}{2} = 6$ ways. Then 1 must come before 2 in those two positions and 3 before 4 in the other two, which is 1 way each.
6. So $P(X \cap Y) = \dfrac{6 \cdot 1 \cdot 1}{24} = \dfrac{1}{4}$. This equals $P(X)\,P(Y) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$. Hence X and Y are independent, and (B) is true.

KEY IDEA

Events that depend on the relative order of disjoint sets of elements, such as {1,2} and {3,4}, are independent in a uniformly random permutation. Each pair has a probability of 1/2 and the joint probability is 1/4.

WATCH OUT FOR

Assuming that "1 before 2" and "3 before 4" must interact because they live in the same permutation. Independence holds here, and it can be checked by counting the 24 orders of the four elements.

Q.15 MCQ · 1 MARK

Computer Organization and Architecture › I/O, interrupts & DMA

Which one of the following statements is FALSE?

(A) In the cycle stealing mode of DMA, one word of data is transferred between an I/O device and main memory in a stolen cycle

(B) For bulk data transfer, the burst mode of DMA has a higher throughput than the cycle stealing mode

(C) Programmed I/O mechanism has a better CPU utilization than the interrupt driven I/O mechanism

(D) The CPU can start executing an interrupt service routine faster with vectored interrupts than with non-vectored interrupts

Official answer: **C**

SOLUTION

1. Statement (A): In cycle stealing DMA, the DMA controller steals one bus cycle at a time from the CPU, and in that single stolen cycle exactly one word is transferred between the I/O device and main memory. This is the standard definition of cycle stealing, so (A) is TRUE.
2. Statement (B): Burst mode DMA acquires the system bus and holds it for the entire block transfer, moving all words consecutively without releasing the bus between words; cycle stealing mode gives the bus back to the CPU after every single word, incurring repeated arbitration overhead. For bulk (block) data transfer this makes burst mode's throughput higher than cycle stealing mode's, so (B) is TRUE.
3. Statement (C): Programmed I/O forces the CPU into a busy-wait polling loop, continuously testing the device's status flag and doing no other useful work while waiting. This is 100% CPU time spent waiting. Interrupt-driven I/O instead lets the CPU execute other instructions and is only interrupted when the

device becomes ready, so it uses CPU cycles far more productively. Therefore interrupt-driven I/O has BETTER CPU utilization than programmed I/O, the reverse of what (C) claims. So (C) is FALSE.

4. Statement (D): With vectored interrupts, the interrupting device (or dedicated hardware) supplies the address/vector of its ISR directly, so the CPU jumps straight to the correct ISR. With non-vectored interrupts, the CPU must first execute a polling routine or read a fixed common address to work out which device interrupted, before it can locate and start the correct ISR. This extra identification step delays ISR execution under non-vectored interrupts, so vectored interrupts let the CPU start executing the ISR faster, making (D) TRUE.
5. Exactly one statement must be false; (A), (B), and (D) all hold under standard DMA/interrupt theory, while (C) inverts the well-known CPU-utilization ordering (interrupt-driven I/O > programmed I/O). Hence the false statement is (C).

KEY IDEA
Compare each option against standard I/O/DMA facts: cycle stealing (one word per stolen cycle), burst mode (higher throughput than cycle stealing for bulk transfer), programmed I/O (CPU busy-waits, so WORSE utilization than interrupt-driven I/O; this inverts the fact, making it the false statement), and vectored interrupts (faster ISR dispatch than non-vectored, which needs a polling/query sequence).

WATCH OUT FOR
Confusing "faster/simpler to implement" with "better CPU utilization": programmed I/O is simpler but wastes CPU cycles polling, so it has WORSE utilization than interrupt-driven I/O, not better.

Q.16 MCQ · 1 MARK

Computer Networks › DNS & HTTP

A user starts browsing a webpage hosted at a remote server. The browser opens a single TCP connection to fetch the entire webpage from the server. The webpage consists of a top-level index page with multiple embedded image objects. Assume that all caches (e.g., DNS cache, browser cache) are all initially empty. The following packets leave the user's computer in some order.

(i) HTTP GET request for the index page
(ii) DNS request to resolve the web server's name to its IP address
(iii) HTTP GET request for an image object
(iv) TCP SYN to open a connection to the web server

Which one of the following is the CORRECT chronological order (earliest in time to latest) of the packets leaving the computer ?

(A) (iv), (ii), (iii), (i)
(B) (ii), (iv), (iii), (i)
(C) (ii), (iv), (i), (iii)
(D) (iv), (ii), (i), (iii)

Official answer: **C**

SOLUTION

1. All caches are empty, so the browser does not know the web server's IP address. The first packet to leave the computer is the DNS request (ii).
2. A TCP SYN needs the server's IP address as its destination. So (iv) can leave only after the DNS reply arrives. Order so far: (ii), (iv).
3. The SYN starts the three-way handshake. Once the connection is established, the browser sends the HTTP GET for the index page (i).

4. The image URLs appear only inside the index page's HTML. The browser must receive and parse that page before it can request any image.
5. So the GET for an image object (iii) leaves last, over the same single TCP connection.
6. Chronological order: $(\text{ii}) \rightarrow (\text{iv}) \rightarrow (\text{i}) \rightarrow (\text{iii})$, which is option (C).

KEY IDEA

Each packet depends on an earlier one. The IP address from DNS is needed for the SYN. An established connection is needed for the GET. The index page must arrive before the image URLs are known.

WATCH OUT FOR

Sending the TCP SYN before the DNS lookup, as in options (A) and (D). Another mistake is asking for an image before the index page, as in options (A) and (B). The image URLs are inside the index page, so it must come first.

Q.17 MCQ · 1 MARK

Data Structures and Algorithms › Complexity

Given an integer array of size N, we want to check if the array is sorted (in either ascending or descending order). An algorithm solves this problem by making a single pass through the array and comparing each element of the array only with its adjacent elements. The worst-case time complexity of this algorithm is

(A) both $O(N)$ and $\Omega(N)$

(B) $O(N)$ but not $\Omega(N)$

(C) $\Omega(N)$ but not $O(N)$

(D) neither $O(N)$ nor $\Omega(N)$

Official answer: **A**

SOLUTION

1. The algorithm makes a single left-to-right pass through the N-element array, comparing each element only with its adjacent neighbor(s).
2. Such a pass performs at most N-1 comparisons, regardless of the input, giving an unconditional upper bound of O(N) on the running time.
3. To determine with certainty whether the array is sorted (ascending or descending), the algorithm must inspect every adjacent pair: if any pair were left unchecked, an adversarial input could hide a violation there while every checked pair still looks consistent.
4. Consider the worst-case inputs for this requirement: an array that actually is fully sorted, or one whose only violation of an otherwise sorted run occurs at the very last adjacent pair. In either case, a single-pass, adjacent-only algorithm cannot safely stop before reaching that last pair.
5. Hence, in the worst case, the algorithm must perform all N-1 comparisons, giving a lower bound of Omega(N) on the worst-case running time.
6. A running time that is simultaneously O(N) and Omega(N) is Theta(N) in the worst case.
7. Therefore the worst-case time complexity is both O(N) and Omega(N), option (A).

KEY IDEA

A single pass over an N-element array that only compares adjacent pairs makes at most N-1 comparisons, so the running time is bounded above by O(N) in every case. In the worst case (e.g. an array that actually is sorted, or one whose only violation is at the very last adjacent pair) the algorithm cannot conclude the answer without inspecting every adjacent pair, so it must also perform Omega(N) work. A bound that is simultaneously O(N) and Omega(N) is Theta(N), so the worst-case complexity satisfies both O(N) and Omega(N).

WATCH OUT FOR

Assuming that because the algorithm might exit early on finding a violation, its worst-case lower bound could be less than Omega(N), but the worst case includes inputs (e.g. an already-sorted array) where no early exit is possible, forcing all N-1 comparisons, so the lower bound Omega(N) still holds for the worst case.

Q.18 MCQ · 1 MARK

Programming in C › C basics

Consider the following C program:

```
#include <stdio.h>
int main(){
   int a = 6;
   int b = 0;
   while(a < 10) {
      a = a / 12 + 1;
      a += b;}
   printf("%d", a);
   return 0;}
```

Which one of the following statements is CORRECT?

(A) The program prints 9 as output

(B) The program prints 10 as output

(C) The program gets stuck in an infinite loop

(D) The program prints 6 as output

Official answer: **C**

SOLUTION

1. Initial state: a = 6, b = 0 (b is never modified anywhere in the program).
2. Check loop condition: a < 10, i.e. 6 < 10, true; enter the loop body.
3. Evaluate `a = a / 12 + 1`: with int division (truncation toward zero, both operands non-negative), 6 / 12 = 0, so a = 0 + 1 = 1.
4. Evaluate `a += b`: b = 0, so a stays 1.
5. Recheck loop condition: a = 1 < 10, true; enter the loop body again.
6. Evaluate `a = a / 12 + 1`: 1 / 12 = 0 (int division), so a = 0 + 1 = 1.
7. Evaluate `a += b`: b is still 0, so a stays 1.
8. From this point the state (a = 1, b = 0) exactly recreates itself every iteration: a is reset to 1 by the assignment and b never changes, so a can never reach 10 and the `while` condition never becomes false. The program is stuck in an infinite loop and never reaches the `printf` statement.

KEY IDEA

Integer division truncates toward zero in C, so once `a` becomes small (≤ 11), `a / 12` always evaluates to 0 and `a` collapses to the fixed point 1; since `b` is never modified, the loop state never changes and the loop cannot terminate.

WATCH OUT FOR

Assuming `a` will eventually grow toward 10 because of the `+= b` step, without noticing that `b` is initialized to 0 and never updated anywhere in the program, so it contributes nothing and `a` is permanently trapped at 1 after the first iteration.

Q.19 MCQ · 1 MARK

Programming in C › Recursion

Q.19 Consider the following C program:

```
#include <stdio.h>
void fX();
int main(){
    fX();
    return 0;}
```

```
void fX(){
 char a;
 if((a=getchar()) != '\n')
     fX();
 if(a != '\n')
     putchar(a);}
```

Assume that the input to the program from the command line is `1234` followed by a newline character. Which one of the following statements is CORRECT?

(A) The program will not terminate

(B) The program will terminate with no output

(C) The program will terminate with `4321` as output

(D) The program will terminate with `1234` as output

Official answer: **C**

SOLUTION

1. Each call of fX reads one character into its own local variable a, so every call keeps its own character: the first four calls hold '1', '2', '3' and '4'.
2. Input 1234 followed by a newline: the first call reads '1' and, since it is not '\n', calls fX again; the next calls read '2', '3' and '4' in turn and each calls fX again.
3. The fifth call reads '\n', so it does not recurse. Its test a != '\n' is false, so it prints nothing and returns.
4. Control now returns through the earlier calls in reverse order. Each one reaches putchar(a) after its recursive call has finished, so the call holding '4' prints first, then '3', then '2', then '1'.

5. The recursion stops at the newline, so the program terminates, and the output is 4321. The answer is (C).

KEY IDEA

Each call keeps its own local character and prints it only after the deeper call returns, so characters read left to right are printed right to left: the recursion reverses the input.

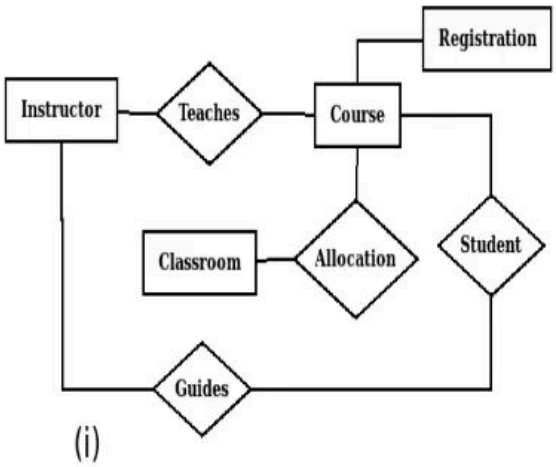
WATCH OUT FOR

Expecting the characters in the order they were read (1234, option D), or thinking the newline stops all output (option B). The newline only stops the recursion and is itself never printed; every earlier call still prints its own character on the way back.

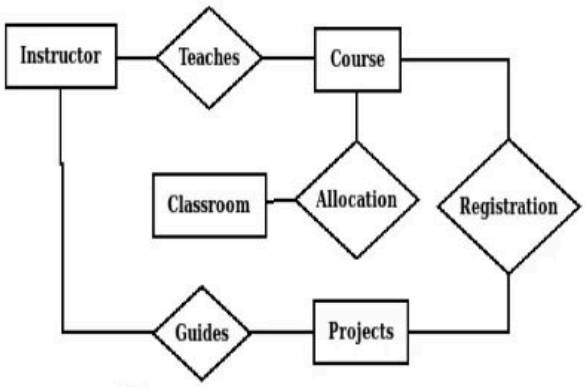
Q.20 MCQ · 1 MARK

Database Management and Warehousing › ER model

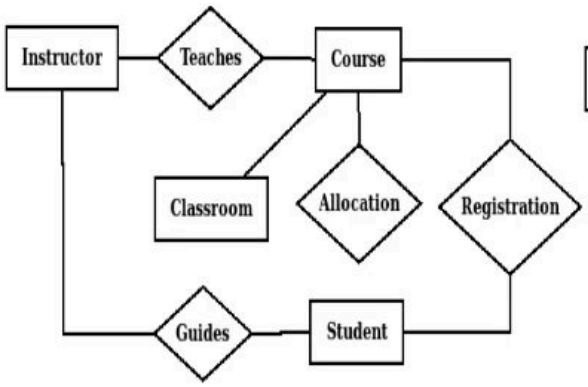
Q.20 Let S be the specification: "Instructors teach courses. Students register for courses. Courses are allocated classrooms. Instructors guide students." Which one of the following ER diagrams CORRECTLY represents S?



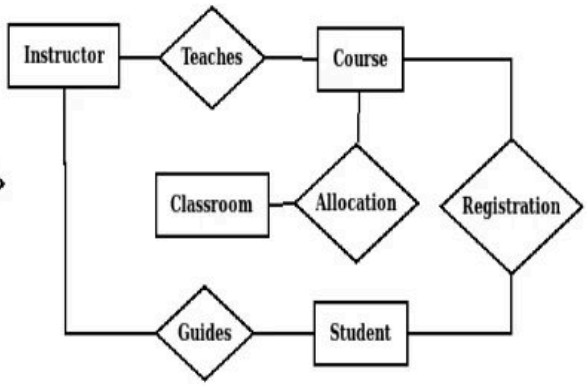
(i)



(ii)



(iii)



(iv)

(A) (i)

(B) (ii)

(C) (iii)

(D) (iv)

Official answer: **D**

SOLUTION

1. Read S as entities and relationships. Entities (nouns): Instructor, Course, Student, Classroom. Relationships (verbs): Teaches (Instructor-Course), Registration (Student-Course), Allocation (Course-Classroom), Guides (Instructor-Student).
2. A correct diagram therefore has four rectangles (entities) and four diamonds (relationships). Each diamond must be joined to exactly the two entities it relates, and two entities must never be joined directly by a plain line.
3. Diagram (i): Student is drawn as a diamond and Registration as a rectangle, so an entity and a relationship are swapped. Wrong.

4. Diagram (ii): there is no Student entity. The Registration and Guides diamonds connect to a 'Projects' entity that S never mentions. Wrong.
5. Diagram (iii): Classroom is joined to Course by a direct line, and the Allocation diamond is not joined to Classroom. Entities cannot be linked directly, and Allocation then does not relate Course and Classroom. Wrong.
6. Diagram (iv): Instructor-Teaches-Course, Course-Registration-Student, Course-Allocation-Classroom and Instructor-Guides-Student. Every entity and relationship in S appears once, and every diamond sits between the right pair of entities. Correct.

KEY IDEA
Nouns in the specification become entities (rectangles) and verbs become relationships (diamonds). Each diamond must join exactly the two entities it relates, and entities are never joined directly.

WATCH OUT FOR
Overlooking the small structural defects in the wrong diagrams: Student drawn as a diamond, an invented entity such as Projects, or Classroom joined straight to Course with Allocation left dangling.

Q.21 MCQ · 1 MARK

Database Management and Warehousing › Indexing

In a B+ tree, the requirement of at least half-full (50%) node occupancy is relaxed for which one of the following cases?

(A) Only the root node
(B) All leaf nodes
(C) All internal nodes
(D) Only the leftmost leaf node

Official answer: **A**

SOLUTION

1. In a B+ tree of order m, every node other than the root must be at least half full. An internal node holds at least $\lceil m/2 \rceil$ children, and a leaf holds at least $\lceil (m-1)/2 \rceil$ keys.
2. The root is the exception. A root that is a leaf may hold as few as 1 key, which is needed while the tree holds very few records.
3. An internal root needs only 2 children, however large m is.
4. So the occupancy rule is relaxed for the root alone. Options (B) and (C) are wrong: leaves and internal nodes that are not the root must stay at least half full, and splits, borrowing and merges maintain this.
5. Option (D) is wrong: no leaf is exempt because of its position. The leftmost leaf follows the same minimum as every other non-root leaf.
6. Hence the 50% occupancy requirement is relaxed only for the root node, which is option (A).

KEY IDEA
Every non-root node of a B+ tree must be at least half full, and only the root is exempt: it can be a leaf with a single key, or an internal node with just 2 children.

WATCH OUT FOR
Thinking that leaf nodes, or the leftmost leaf, can be under-full because they sit at an edge of the tree. Only the root is exempt, wherever a node sits.

Q.22 MSQ · 1 MARK

Database Management and Warehousing › Normal forms

Which of the following statements about a relation **R** in first normal form (1NF) is/are TRUE ?

(A) **R** can have a multi-attribute key
(B) **R** cannot have a foreign key
(C) **R** cannot have a composite attribute

(D) **R** cannot have more than one candidate key

Official answer: **A, C**

SOLUTION

1. First normal form requires every attribute value to be atomic: no multivalued attributes, no nested relations and no composite attributes. It places no restriction on keys, foreign keys or the number of candidate keys.
2. (A) A key may be made of several attributes, e.g. $\{\text{StudentID}, \text{CourseID}\}$ in an enrolment relation, and the relation can still be in 1NF. TRUE.
3. (B) Foreign keys are allowed in a 1NF relation, e.g. an enrolment relation that references a student relation. So "cannot have a foreign key" is FALSE.
4. (C) A composite attribute such as Address (street, city, PIN) is not atomic, so a relation in 1NF cannot have one; it must be split into simple attributes. TRUE.
5. (D) A relation can have several candidate keys and still be in 1NF (even in BCNF), e.g. $\text{Emp}(\text{EmpID}, \text{Email}, \text{Name})$ with both EmpID and Email as candidate keys. So the statement is FALSE.
6. Hence the true statements are (A) and (C).

KEY IDEA
1NF only demands atomic attribute values, which rules out composite and multivalued attributes. Keys, foreign keys and the number of candidate keys are not restricted.

WATCH OUT FOR
Treating 1NF as if it limited keys, for example thinking a multi-attribute key, a foreign key or several candidate keys break it. Those concerns belong to higher normal forms or to the relational model in general.

Q.23 MSQ · 1 MARK

Theory of Computation › Regular languages

Let L_1, L_2 be two regular languages and L_3 a language which is not regular. Which of the following statements is/are always TRUE?

(A) $L_1 = L_2$ if and only if $L_1 \cap \overline{L_2} = \phi$

(B) $L_1 \cup L_3$ is not regular

(C) $\overline{L_3}$ is not regular

(D) $\overline{L_1} \cup \overline{L_2}$ is regular

Official answer: **C, D**

SOLUTION

1. Regular languages are closed under complement, union and intersection. Test each statement against these closure properties and, where it fails, a counterexample.
2. (A) $L_1 \cap \overline{L_2} = \phi$ means no string of L_1 lies outside L_2, that is $L_1 \subseteq L_2$. This does not give equality. Take $L_1 = \{a\}$, $L_2 = \{a, b\}$: the intersection is empty but $L_1 \neq L_2$. So the 'only if' direction fails and (A) is not always true.
3. (B) Take $L_1 = \Sigma^*$ (regular) and any non-regular L_3. Then $\Sigma^* \cup L_3 = \Sigma^*$, which is regular. So (B) is not always true.
4. (C) If $\overline{L_3}$ were regular, then $L_3 = \overline{\overline{L_3}}$ would be regular by closure under complement. That contradicts the hypothesis, so $\overline{L_3}$ is not regular. (C) is always true.
5. (D) $\overline{L_1}$ and $\overline{L_2}$ are regular by closure under complement, and their union is regular by closure under union. (D) is always true.
6. Correct options: (C) and (D).

KEY IDEA
Regular languages are closed under complement and union, so complementing a non-regular language keeps it non-regular, and a union of complements of regular languages stays regular.

WATCH OUT FOR
Reading $L_1 \cap \overline{L_2} = \phi$ as $L_1 = L_2$. It only says $L_1 \subseteq L_2$. Another slip is assuming a union with a non-regular language is non-regular, when $L_1 = \Sigma^*$ absorbs it.

Q.24 MSQ · 1 MARK

Operating Systems › Processes & threads

Which of the following statements about threads is/are TRUE?

(A) Threads can only be implemented in kernel space

(B) Each thread has its own file descriptor table for open files

(C) All the threads belonging to a process share a common stack

(D) Threads belonging to a process are by default not protected from each other

Official answer: **D**

SOLUTION

1. Recall what the threads of one process share and what each keeps private. Shared: code, global data, heap, open files (the file descriptor table), signal handlers and the address space. Private: program counter, registers, stack and thread ID.
2. (A) False. Threads can be implemented in user space (user-level thread libraries, the many-to-one model) as well as in kernel space, or as a hybrid of the two. "Only in kernel space" is wrong.
3. (B) False. The file descriptor table belongs to the process, so all its threads share it. A file opened by one thread can be used by the others.
4. (C) False. Each thread needs its own stack for its call frames, local variables and return addresses. Only the heap and global data are shared.
5. (D) True. All threads of a process run in one address space with no memory protection between them. One thread can read or overwrite another's data, and a stray write can corrupt the whole process. Any protection has to come from the programmer, for example with locks.
6. Only (D) is correct.

KEY IDEA
Threads of a process share code, global data, heap and open files. Each thread has its own stack, registers and program counter. Because they share one address space, the operating system does not protect them from each other.

WATCH OUT FOR
Thinking each thread has its own file descriptor table (B), or that threads must live in the kernel (A). The table belongs to the process. Thread libraries can also run entirely in user space.

Q.25 MSQ · 1 MARK

Operating Systems › Processes & threads

Which of the following process state transitions is/are NOT possible?

(A) Running to Ready

(B) Waiting to Running

(C) Ready to Waiting

(D) Running to Terminated

Official answer: **B, C**

SOLUTION

1. Use the five-state process model: New, Ready, Running, Waiting (Blocked), Terminated. The legal transitions are New to Ready, Ready to Running (dispatch), Running to Ready (preemption or time-slice expiry), Running to Waiting (I/O or event wait), Waiting to Ready (I/O or event completion) and Running to Terminated (exit).
2. (A) Running to Ready: possible. A timer interrupt or the arrival of a higher-priority process preempts the running process and puts it back in the ready queue.
3. (B) Waiting to Running: not possible. When the awaited event completes, the process moves to Ready and must wait for the scheduler to dispatch it. It never goes from Waiting straight onto the CPU.
4. (C) Ready to Waiting: not possible. A process blocks only by executing a blocking call, such as an I/O request. A Ready process is not executing, so it cannot make that call.
5. (D) Running to Terminated: possible. The process finishes or calls exit.
6. The transitions that are NOT possible are (B) and (C).

KEY IDEA
In the five-state model a blocked process can only go Waiting to Ready, and only a Running process can block. So Waiting to Running and Ready to Waiting cannot occur.

WATCH OUT FOR
Treating a process that has just received its I/O as running at once (Waiting to Running), or thinking a Ready process can be moved to Waiting. Blocking is something a running process does to itself.

Q.26 MSQ · 1 MARK

Compiler Design › Parsing

Which of the following is/are Bottom-Up Parser(s)?

(A) Shift-reduce Parser

(B) Predictive Parser

(C) LL(1) Parser

(D) LR Parser

Official answer: **A, D**

SOLUTION

1. Bottom-up parsing builds the parse tree from the leaves (the input tokens) up to the start symbol, tracing a rightmost derivation in reverse.
2. (A) A shift-reduce parser shifts tokens onto a stack and reduces handles to non-terminals. This is the general bottom-up method, so it qualifies.
3. (B) A predictive parser is a recursive-descent or table-driven parser that chooses a production from the lookahead. It expands from the start symbol, so it is top-down.
4. (C) An LL(1) parser is the table-driven predictive parser: Left-to-right scan, Leftmost derivation, 1 lookahead. It is top-down.
5. (D) An LR parser (LR(0), SLR, LALR, CLR) is a shift-reduce parser driven by an automaton of items $A \to \alpha \cdot \beta$. It is bottom-up.
6. So the bottom-up parsers are (A) and (D).

KEY IDEA
Bottom-up parsers (shift-reduce, and the LR family built on it) reduce the input to the start symbol. Top-down parsers (predictive, recursive descent, LL(1)) expand from the start symbol.

WATCH OUT FOR
Choosing the predictive or LL(1) parser because it is table-driven and uses a stack. Both build the tree from the root downwards, so they are top-down.

Q.27 MSQ · 1 MARK

Engineering Mathematics › Probability & statistics

Let A and B be two events in a probability space with $P(A) = 0.3$, $P(B) = 0.5$, and $P(A \cap B) = 0.1$. Which of the following statements is/are TRUE?

(A) The two events A and B are independent

(B) $P(A \cup B) = 0.7$

(C) $P(A \cap B^c) = 0.2$, where B^c is the complement of the event B

(D) $P(A^c \cap B^c) = 0.4$, where A^c and B^c are the complements of the events A and B, respectively

Official answer: **B, C**

SOLUTION

1. Independence needs $P(A \cap B) = P(A)P(B)$. Here $P(A)P(B) = 0.3 \times 0.5 = 0.15 \neq 0.1$, so A and B are not independent. (A) is false.
2. By inclusion-exclusion, $P(A \cup B) = P(A) + P(B) - P(A \cap B) = 0.3 + 0.5 - 0.1 = 0.7$. (B) is true.
3. The event A splits into the disjoint parts $A \cap B$ and $A \cap B^c$. So $P(A \cap B^c) = P(A) - P(A \cap B) = 0.3 - 0.1 = 0.2$. (C) is true.
4. By De Morgan, $A^c \cap B^c = (A \cup B)^c$. So $P(A^c \cap B^c) = 1 - P(A \cup B) = 1 - 0.7 = 0.3 \neq 0.4$. (D) is false.
5. The true statements are (B) and (C).

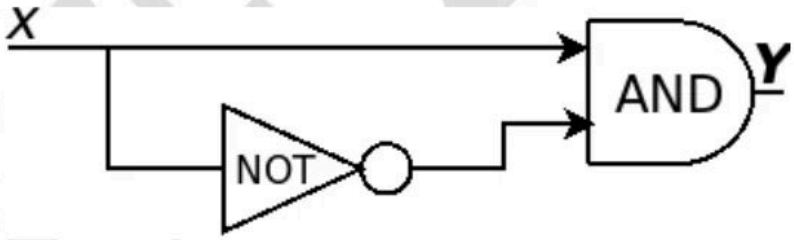
KEY IDEA
Use inclusion-exclusion for the union, split an event by the other event for the intersection with a complement, and use De Morgan for the intersection of two complements.

WATCH OUT FOR
Taking $P(A^c \cap B^c)$ as $1 - P(A) - P(B) + P(A \cap B)$ with a sign slip, or as $(1 - 0.3)(1 - 0.5) = 0.35$ by wrongly assuming independence. Neither gives 0.3, and neither supports option (D).

Q.28 MSQ · 1 MARK

Digital Electronics › Combinational

Q.28 Consider the circuit shown below where the gates may have propagation delays. Assume that all signal transitions occur instantaneously and that wires have no delays. Which of the following statements about the circuit is/are CORRECT?



(A) With no propagation delays, the output ***Y*** is always logic Zero

(B) With no propagation delays, the output ***Y*** is always logic One

(C) With propagation delays, the output ***Y*** can have a transient logic One after ***X*** transitions from logic Zero to logic One

(D) With propagation delays, the output ***Y*** can have a transient logic Zero after ***X*** transitions from logic One to logic Zero

Official answer: **A, C**

SOLUTION

1. Read the circuit: X goes straight to one input of the AND gate and also through a NOT gate to the other input, so $Y = X \cdot \overline{X}$. The gates may have a propagation delay t_d.
2. With no delays, $X \cdot \overline{X} = 0$ for both $X = 0$ and $X = 1$, so Y is always logic Zero. (A) is true and (B) is false.
3. Now include delay and take $X : 0 \to 1$. The AND's direct input rises at once. The NOT output still holds its old value $\overline{0} = 1$ for a time t_d.
4. During that interval both AND inputs are 1, so $Y = 1$ briefly (a static-0 hazard glitch). After t_d the NOT output falls to 0 and Y returns to 0. So (C) is true.
5. Take $X : 1 \to 0$. The direct input falls at once to 0, so the AND has a 0 input and Y stays 0. The NOT output rises to 1 only after t_d, but the other input is already 0.
6. Y never leaves 0, so it has no transient Zero (it is already Zero) and (D) is false.
7. Correct options: A and C.

KEY IDEA

The circuit computes $X \cdot \overline{X}$, which is 0 without delays. With a delayed NOT, a rising X briefly sees both AND inputs at 1, giving a short pulse of 1 (static-0 hazard).

WATCH OUT FOR

Assuming a glitch also appears on a falling X. On $1 \to 0$ the direct input drops first and forces the AND output to 0 before the NOT output rises, so no glitch occurs.

Q.29 MSQ · 1 MARK

Computer Networks › TCP

TCP client P successfully establishes a connection to TCP server Q. Let N_P denote the sequence number in the SYN sent from P to Q. Let N_Q denote the acknowledgement number in the SYN ACK from Q to P. Which of the following statements is/are CORRECT?

(A) The sequence number N_P is chosen randomly by P

(B) The sequence number N_P is always 0 for a new connection

(C) The acknowledgement number N_Q is equal to N_P

(D) The acknowledgement number N_Q is equal to $N_P + 1$

Official answer: **A, D**

SOLUTION

1. In the three-way handshake, the client P sends a SYN carrying its initial sequence number (ISN), N_P.
2. TCP does not start every connection at 0. It picks the ISN unpredictably, from a clock-driven or pseudo-random generator. This keeps delayed segments from an earlier connection from being mistaken for the new one, and it makes sequence-number guessing attacks harder. So (A) is true and (B) is false.
3. The server Q replies with a SYN-ACK. A SYN occupies one sequence number, so Q acknowledges the next sequence number it expects from P: $N_Q = N_P + 1$.
4. So (C), $N_Q = N_P$, is false, and (D), $N_Q = N_P + 1$, is true.
5. Correct statements: (A) and (D).

KEY IDEA

A SYN segment consumes one sequence number. The acknowledgement number in the SYN-ACK is therefore the client's ISN plus 1. The ISN itself is chosen unpredictably, not fixed at 0.

WATCH OUT FOR

Thinking the ACK in the SYN-ACK echoes the client's sequence number ($N_Q = N_P$). It actually names the next number expected, which is $N_P + 1$, because the SYN uses up one sequence number. Another slip is assuming a new connection always starts at sequence number 0.

Q.30 MSQ · 1 MARK

Computer Organization and Architecture › Pipelining

Consider a 5-stage pipelined processor with Instruction Fetch (IF), Instruction Decode (ID), Execute (EX), Memory Access (MEM), and Register Writeback (WB) stages. Which of the following statements about *forwarding* is/are CORRECT?

(A) In a pipelined execution, forwarding means the result from a source stage of an earlier instruction is passed on to the destination stage of a later instruction

(B) In forwarding, data from the output of the MEM stage can be passed on to the input of the EX stage of the next instruction

(C) Forwarding cannot prevent all pipeline stalls

(D) Forwarding does not require any extra hardware to retrieve the data from the pipeline stages

Official answer: **A, B, C**

SOLUTION

1. Definition check (A): Forwarding means a result computed at a source stage of an earlier instruction is passed directly to the destination stage of a later instruction that needs it, without waiting for it to be written back to the register file. This is the standard definition, so (A) is correct.
2. Check (B): The MEM/WB pipeline latch holds the output of the MEM stage. This value can be forwarded to the EX stage input (e.g., an ALU operand) of a following instruction: this is the classic MEM/WB → EX forwarding path used in a 5-stage pipeline. So (B) is correct.
3. Check (C): Even with full EX/MEM and MEM/WB forwarding in place, a load-use hazard (the instruction right after a load needs the loaded value in its own EX stage, before the load has produced it) cannot be resolved by forwarding: the pipeline must still insert at least one stall cycle. So forwarding reduces stalls but does not prevent all of them, making (C) correct.

4. Check (D): Forwarding needs dedicated bypass datapaths from stage-output latches to stage inputs, multiplexers to select between the forwarded value and the normal register-file value, and forwarding-unit control logic to detect hazards and drive those multiplexers. This is genuine extra hardware, so the claim of 'no extra hardware' is false, making (D) incorrect.
5. Since this is an MSQ, the correct set is every true statement: A, B, and C.

KEY IDEA
Forwarding (bypassing) routes a computed result directly from the pipeline stage that produces it to the pipeline stage that needs it, bypassing the register file/writeback, but it needs dedicated bypass wires, multiplexers, and forwarding-unit control logic, and it still cannot eliminate a load-use hazard (where the needed value does not exist yet when the dependent instruction reaches EX).

WATCH OUT FOR
Assuming forwarding is a "free" software-level trick with no hardware cost (missing why D is false), or assuming forwarding alone eliminates every stall and forgetting the load-use hazard, which still forces at least one bubble even with full forwarding (which is why C is true, not false).

Q.31 MSQ · 1 MARK

Computer Networks › IPv4

Which of the following fields is/are modified in the IP header of a packet going out of a network address translation (NAT) device from an internal network to an external network?

(A) Source IP

(B) Destination IP

(C) Header Checksum

(D) Total Length

Official answer: **A, C**

SOLUTION

1. A NAT device sits between a private (internal) network and the Internet. For a packet going out, it replaces the private source address with its own public address, and usually rewrites the source port in the TCP/UDP header as well.
2. Source IP: the private source address is replaced by the NAT's public address, so this field is modified. Option (A) is correct.
3. Destination IP: the outgoing packet is still addressed to the same external host, so the field is unchanged. (The destination address is rewritten only on the reply, on its way back in.) Option (B) is not modified.
4. Header Checksum: the IPv4 header checksum is computed over the whole IP header, including the source address. Once the source address changes, the old checksum no longer matches, so the NAT must recompute it. This field is modified. Option (C) is correct.
5. Total Length: translating the address does not change the size of the header or payload (an IPv4 address is 4 bytes before and after), so the field stays the same. Option (D) is not modified.
6. Fields modified: Source IP and Header Checksum, which are options (A) and (C).

KEY IDEA
An outgoing NAT rewrites the source address (and port). Any field that depends on the rewritten bytes must also be updated, so the IP header checksum changes with it.

WATCH OUT FOR
Forgetting that the header checksum covers the source address, so it must be recomputed after translation. Another slip is to include the destination address, which changes only on the reverse (incoming) path.

Q.32 NAT · 1 MARK

Discrete Mathematics › Sets & relations

Let A and B be non-empty finite sets such that there exist one-to-one and onto functions (i) from A to B and (ii) from $A \times A$ to $A \cup B$. The number of possible values of $|A|$ is _____

Official answer: **2**

SOLUTION

1. A bijection between two finite sets exists exactly when they have the same number of elements. From (i), $|A| = |B| = n$ with $n \geq 1$, since the sets are non-empty.
2. From (ii), $|A \times A| = |A \cup B|$, so $|A \cup B| = n^2$.
3. For any two finite sets, $\max(|A|, |B|) \leq |A \cup B| \leq |A| + |B|$. Here this reads $n \leq |A \cup B| \leq 2n$.
4. Combining, $n \leq n^2 \leq 2n$. The left inequality holds for every $n \geq 1$. Dividing the right inequality by $n > 0$ gives $n \leq 2$. So $n \in \{1, 2\}$.
5. Check $n = 1$: take $A = B = \{a\}$. Then $|A \cup B| = 1 = 1^2$, and bijections $A \to B$ and $A \times A \to A \cup B$ both exist. Valid.
6. Check $n = 2$: take $A = \{1, 2\}$ and $B = \{3, 4\}$, which are disjoint. Then $|A \cup B| = 4 = 2^2$, and both bijections exist. Valid.
7. Hence $|A|$ can be 1 or 2 and nothing else, so there are 2 possible values.

KEY IDEA

A bijection exists exactly when the two sets have equal size, so the conditions reduce to $|A| = |B| = n$ and $n^2 = |A \cup B|$. The union is squeezed between n and $2n$, which forces $n \leq 2$.

WATCH OUT FOR

Assuming A and B must be disjoint, or must be equal, which fixes $|A \cup B|$ at $2n$ or n and leaves only one value. The union size can be anything from n to $2n$. Another slip is forgetting that $n = 1$ works, because $A = B$ gives $1 = 1^2$.

Q.33 NAT · 1 MARK

Programming in C › C basics

Q.33 Consider the operator precedence and associativity rules for the *integer* arithmetic operators given in the table below.

Operator	Precedence	Associativity
+	Highest	Left
−	High	Right
*	Medium	Right
/	Low	Right

The value of the expression $3 + 1 + 5 * 2 / 7 + 2 - 4 - 7 - 6 / 2$ as per the above rules is

Official answer: **6**

SOLUTION

1. Step 1: Identify precedence order from the table (tighter-binding to loosest): + (Highest) > − (High) > *(Medium) > / (Low). Associativity: + Left, − Right,* Right, / Right.
2. Step 2: The lowest-precedence operator (/) forms the outermost split of the whole expression. Two '/' occur, and / is right-associative, so the expression is A / (B / C), where A = '3+1+5*2', B = '7+2-4-7-6',

C = '2'.

3. Step 3: Evaluate A: it contains only + and *; since + binds tighter than* here, all the + operations collapse first: 3+1+5 = 9, then the remaining * applies: A = 9*2 = 18.
4. Step 4: Evaluate B: it contains + and −; + binds tighter, so 7+2 = 9 collapses first, leaving the right-associative chain 9−4−7−6 = 9−(4−(7−6)) = 9−(4−1) = 9−3 = 6, so B = 6.
5. Step 5: C = 2.
6. Step 6: Combine at the top level (right-associative /): A / (B / C) = 18 / (6 / 2) = 18 / 3 = 6.

KEY IDEA

Precedence decides which operator's operand-groups are formed first (tighter binding = evaluated/collapsed first); the *lowest*-precedence operator present at any level is the one whose split happens outermost/last. Associativity only matters for resolving a chain of *equal*-precedence operators.

WATCH OUT FOR

Assuming "+" being listed first / "Highest" automatically means it is evaluated last (as in standard math where low precedence like + is outermost); here the table explicitly reassigns precedence so + is tightest-binding and / is loosest-binding. Using everyday PEMDAS intuition instead of the table gives a different (wrong) grouping and answer.

Q.34 NAT · 1 MARK

Discrete Mathematics › Graph theory

The number of spanning trees in a complete graph of 4 vertices labelled A, B, C, and D is _____

Official answer: **16**

SOLUTION

1. The graph is the complete graph K_4 on the labelled vertices A, B, C, D. Each spanning tree has 4 vertices and therefore exactly 3 edges.
2. Cayley's formula: the number of labelled trees on n vertices, which equals the number of spanning trees of K_n, is n^{n-2}.
3. For $n = 4$: $4^{4-2} = 4^2 = 16$.
4. Check by counting. A tree on 4 vertices is either a star or a path. Stars: the centre can be chosen in 4 ways, and the other three vertices attach to it, giving 4 trees. Paths: the 4 vertices can be listed in $4! = 24$ orders, and each path appears twice (once read in each direction), giving $24/2 = 12$ trees. No other tree shape exists on 4 vertices, so the total is $4 + 12 = 16$.
5. Second check using edge counts. K_4 has 6 edges, so there are $\binom{6}{3} = 20$ ways to choose 3 edges. Exactly 4 of these choices form a triangle (one for each triple of vertices), and a triangle leaves the fourth vertex isolated, so it is not a tree. Any other 3-edge choice has no cycle and is a spanning tree. That gives $20 - 4 = 16$.

KEY IDEA

Spanning trees of K_n are exactly the labelled trees on n vertices, and Cayley's formula counts them as n^{n-2}. For K_4 this gives $4^2 = 16$.

WATCH OUT FOR

Counting trees only up to isomorphism gives 2 (a path and a star). The vertices are labelled, so each labelling counts separately: 4 stars and 12 paths.

Q.35 NAT · 1 MARK

Database Management and Warehousing › Relational algebra

Q.35 Consider the following two relations, ***R(A,B)*** and ***S(A,C)***:

R	
A	***B***
10	20
20	30
30	40
30	50
50	95

S	
A	***C***
10	90
30	45
40	80

The total number of tuples obtained by evaluating the following expression

$$\sigma_{B<C}(R \bowtie_{R.A=S.A} S)$$

is

Official answer: **2**

SOLUTION

1. The join condition $R.A = S.A$ pairs a tuple of R with a tuple of S only when their A values are equal.
2. $A = 10$: R has $(10, 20)$ and S has $(10, 90)$. This gives the joined tuple $(A, B, C) = (10, 20, 90)$.
3. $A = 20$: R has $(20, 30)$, but S has no tuple with $A = 20$. Nothing joins.
4. $A = 30$: R has $(30, 40)$ and $(30, 50)$, and S has $(30, 45)$. This gives $(30, 40, 45)$ and $(30, 50, 45)$.
5. $A = 50$: R has $(50, 95)$, but S has no tuple with $A = 50$. Nothing joins. The tuple $(40, 80)$ of S has no partner in R either.
6. The join therefore has 3 tuples: $(10, 20, 90)$, $(30, 40, 45)$ and $(30, 50, 45)$.
7. Apply $\sigma_{B<C}$ to each tuple: $20 < 90$ is true, so it is kept. $40 < 45$ is true, so it is kept. $50 < 45$ is false, so it is dropped.
8. Two tuples remain, so the answer is 2.

KEY IDEA

Evaluate the equi-join first by matching equal A values. Then filter the joined tuples with the selection condition $B < C$.

WATCH OUT FOR

Counting the join result (3 tuples) and forgetting the selection $B < C$. Another slip is comparing B with C before joining, or treating the non-matching tuples ($A = 20$, $A = 50$, $A = 40$) as part of the join.

Q.36 MCQ · 2 MARKS

Computer Networks › Performance

Consider a network path P—Q—R between nodes P and R via router Q. Node P sends a file of size 10^6 bytes to R via this path by splitting the file into chunks of 10^3 bytes each. Node P sends these chunks one after the other without any wait time between the successive chunk transmissions. Assume that the size of extra headers added to these chunks is negligible, and that the chunk size is less than the MTU.

Each of the links P—Q and Q—R has a bandwidth of 10^6 bits/sec, and negligible propagation latency. Router Q immediately transmits every packet it receives from P to R, with negligible processing and queueing delays. Router Q can simultaneously receive on link P—Q and transmit on link Q—R.

Assume P starts transmitting the chunks at time $t = 0$.
Which one of the following options gives the time (in seconds, rounded off to 3 decimal places) at which R

receives all the chunks of the file?

(A) 8.000

(B) 8.008

(C) 15.992

(D) 16.000

Official answer: **B**

SOLUTION

1. The file has 10^6 bytes and each chunk has 10^3 bytes, so there are $10^6/10^3 = 1000$ chunks. Each chunk is $10^3 \times 8 = 8000$ bits.
2. Each link runs at 10^6 bits/s, so one chunk takes $\frac{8000}{10^6} = 0.008\ \text{s}$ to transmit on one link. Propagation, processing and queueing delays are negligible.
3. P sends the chunks back to back. Chunk k is fully received at Q at $0.008k$ seconds, so the last chunk (k = 1000) is fully received at Q at $8.000\ \text{s}$.
4. Q receives a whole chunk before forwarding it (store and forward), and it can receive and transmit at the same time. Chunk k reaches Q at $0.008k$, and the previous chunk finished leaving Q at $0.008(k-1) + 0.008 = 0.008k$. Link Q—R is therefore free exactly when each chunk arrives, and no chunk waits in a queue.
5. So Q starts sending the last chunk to R at $8.000\ \text{s}$. This takes $0.008\ \text{s}$ more.
6. R has the whole file at $8.000 + 0.008 = 8.008\ \text{s}$. In general, for n chunks over 2 equal-rate links the time is $(n+1)\frac{L}{B} = 1001 \times 0.008 = 8.008\ \text{s}$. The answer is option (B).

KEY IDEA

With store-and-forward over two equal-rate links, the transmissions overlap like a pipeline. The total time is (n + 1) chunk-transmission times, not 2n.

WATCH OUT FOR

Adding the two links' transmission times for the whole file (8 + 8 = 16 s). This forgets that Q forwards each chunk while it is still receiving the next one, so the second link only adds one chunk time (0.008 s).

Q.37 MCQ · 2 MARKS

Compiler Design › Syntax-directed translation

Q.37 Consider the following syntax-directed definition (SDD).

$S \rightarrow DHTU$	$\{ S.val = D.val + H.val + T.val + U.val; \}$
$D \rightarrow$ "M"D_1	$\{ D.val = 5 + D_1.val; \}$
$D \rightarrow \epsilon$	$\{ D.val = -5; \}$
$H \rightarrow$ "L"H_1	$\{ H.val = 5 * 10 + H_1.val; \}$
$H \rightarrow \epsilon$	$\{ H.val = -10; \}$
$T \rightarrow$ "C"T_1	$\{ T.val = 5 * 100 + T_1.val; \}$
$T \rightarrow \epsilon$	$\{ T.val = -5; \}$
$U \rightarrow$ "K"	$\{ U.val = 5; \}$

Given "MMLK" as the input, which one of the following options is the CORRECT value computed by the SDD (in the attribute $S.val$)?

(A) 45

(B) 50

(C) 55

(D) 65

Official answer: **A**

SOLUTION

1. Parse "MMLK" with $S \rightarrow DHTU$: D derives "MM", H derives "L", T derives ε (there is no "C" in the input), and U derives "K".
2. Every attribute is synthesised, so evaluate bottom-up: each parent takes its value from its children.
3. D: the innermost $D \rightarrow \varepsilon$ gives -5. Then $D \rightarrow$ "M"D_1 applies twice: $5 + (-5) = 0$, then $5 + 0 = 5$. So $D.\text{val} = 5$.
4. H: $H \rightarrow \varepsilon$ gives -10. Then $H \rightarrow$ "L"H_1 gives $5 \cdot 10 + (-10) = 40$. So $H.\text{val} = 40$.
5. T: $T \rightarrow \varepsilon$ gives $T.\text{val} = -5$.
6. U: $U \rightarrow$ "K" gives $U.\text{val} = 5$.
7. Root: $S.\text{val} = 5 + 40 + (-5) + 5 = 45$, which is option (A).

KEY IDEA
Each nonterminal's value is computed bottom-up from its parse subtree. An ε production still contributes its own constant (here a negative base value), and that constant is added back into the parent's total.

WATCH OUT FOR
Forgetting that the ε productions contribute -5, -10 and -5. For example, treating the missing T as 0 instead of -5 gives 50. Counting D as 10, with no -5 base, is another slip.

Q.38 MCQ · 2 MARKS

Compiler Design › Parsing

Consider the following grammar G, with S as the start symbol. The grammar G has three incomplete productions denoted by (1), (2), and (3).

$$S \to daT \mid \underline{(1)}$$
$$T \to aS \mid bT \mid \underline{(2)}$$
$$R \to \underline{(3)} \mid \varepsilon$$

The set of terminals is $\{a, b, c, d, f\}$. The FIRST and FOLLOW sets of the different non-terminals are as follows.

$\text{FIRST}(S) = \{c, d, f\}$, $\text{FIRST}(T) = \{a, b, \varepsilon\}$, $\text{FIRST}(R) = \{c, \varepsilon\}$

$\text{FOLLOW}(S) = \text{FOLLOW}(T)$ is the set of the three symbols c, f and the end marker $, and $\text{FOLLOW}(R) = \{f\}$.

Which one of the following options CORRECTLY fills in the incomplete productions?

(A) (1) $S \to Rf$, (2) $T \to \varepsilon$, (3) $R \to cTR$

(B) (1) $S \to fR$, (2) $T \to \varepsilon$, (3) $R \to cTR$

(C) (1) $S \to fR$, (2) $T \to cT$, (3) $R \to cR$

(D) (1) $S \to Rf$, (2) $T \to cT$, (3) $R \to cR$

Official answer: **A**

SOLUTION

1. $\text{FIRST}(S) = \{c, d, f\}$. The production $S \to daT$ supplies only d, so production (1) must supply both c and f.
2. $S \to fR$ has $\text{FIRST} = \{f\}$, which misses c. So options B and C are out.
3. $S \to Rf$: R is nullable, so $\text{FIRST}(Rf) = (\text{FIRST}(R) \setminus \{\varepsilon\}) \cup \{f\} = \{c, f\}$. This works. It also puts f in $\text{FOLLOW}(R)$, as given.
4. $\text{FIRST}(T) = \{a, b, \varepsilon\}$ contains neither c nor anything else that $T \to cT$ could add. $T \to cT$ would add c to $\text{FIRST}(T)$, and ε would still have no source. Option D is out.
5. $T \to \varepsilon$ is the only choice that supplies the ε in $\text{FIRST}(T)$. Production (2) is $T \to \varepsilon$.
6. Check (3) $R \to cTR$. $\text{FIRST}(R) = \{c, \varepsilon\}$ is correct. For $\text{FOLLOW}(T)$, take $\text{FIRST}(R) \setminus \{\varepsilon\} = \{c\}$. Since R is nullable, add $\text{FOLLOW}(R) = \{f\}$. From $S \to daT$, $\text{FOLLOW}(T) \supseteq \text{FOLLOW}(S)$, and $ is in $\text{FOLLOW}(S)$ because S is the start symbol. From $T \to aS$, $\text{FOLLOW}(S) \supseteq \text{FOLLOW}(T)$. Together these give $\text{FOLLOW}(S) = \text{FOLLOW}(T) = \{c, f\}$ plus the end marker $, as stated. Also $\text{FOLLOW}(R)$ stays $\{f\}$: $R \to cTR$ only adds $\text{FOLLOW}(R)$ to itself.
7. Answer: option A.

KEY IDEA

A production for a non-terminal must contribute exactly the missing terminals to its FIRST set, and a nullable non-terminal contributes its FOLLOW set to whatever precedes it. Use FIRST to pick (1) and (2), then verify (3) and the FOLLOW sets.

WATCH OUT FOR

Writing $S \to fR$ and forgetting that FIRST of a string starting with a nullable R must include FIRST(R) minus ε and so reaches past it. Also, adding $T \to cT$ without noticing it puts c into FIRST(T), which the given set forbids.

Q.39 MCQ · 2 MARKS

Compiler Design › Optimisation

Consider the following pseudo-code.

```
 L1:  t1 = -1
L2:  t2 = 0
L3:  t3 = 0
L4:  t4 = 4 * t3
L5:  t5 = 4 * t2
L6:  t6 = t5 * M
L7:  t7 = t4 + t6
L8:  t8 = a[t7]
L9:  if t8 <= max goto L11
L10: t1 = t8
L11: t3 = t3 + 1
L12: if t3 < M goto L4
L13: t2 = t2 + 1
L14: if t2 < N goto L3
L15: max = t1
```

Which one of the following options CORRECTLY specifies the number of basic blocks and the number of instructions in the largest basic block, respectively ?

(A) 6 and 6

(B) 6 and 7

(C) 7 and 7

(D) 7 and 6

Official answer: **D**

SOLUTION

1. A basic block starts at a leader: the first instruction, any instruction that is the target of a jump, and any instruction that immediately follows a jump (conditional or unconditional).
2. The first instruction is L1. The jump targets are L11 (from L9), L4 (from L12) and L3 (from L14).
3. The instructions immediately after a jump are L10 (after L9), L13 (after L12) and L15 (after L14).
4. The leaders are therefore L1, L3, L4, L10, L11, L13, L15. That is 7 leaders, so there are 7 basic blocks.
5. Each block runs from its leader up to the instruction before the next leader:

Block	Instructions	Count
B_1	$L1$–$L2$	2
B_2	$L3$	1
B_3	$L4$–$L9$	6
B_4	$L10$	1
B_5	$L11$–$L12$	2
B_6	$L13$–$L14$	2
B_7	$L15$	1

6. The counts add up to $2+1+6+1+2+2+1=15$, which matches the 15 instructions, so no instruction is missed or counted twice.
7. The largest block is B_3 (L4 to L9) with 6 instructions. So there are 7 basic blocks and the largest has 6 instructions, which is option (D).

KEY IDEA

Find the leaders (the first instruction, every jump target, every instruction right after a jump). Each basic block runs from one leader up to the instruction before the next leader.

WATCH OUT FOR

Forgetting that L3 is a jump target (from L14) and so starts its own block, separate from L1 - L2 . That gives 6 blocks and the wrong option. Another slip is to leave out the instruction after a conditional jump, which merges L10 into the block before it and makes the largest block 7.

Q.40 MCQ · 2 MARKS

Operating Systems › Synchronisation

Consider the following two threads T1 and T2 that update two shared variables a and b. Assume that initially a = b = 1. Though context switching between threads can happen at any time, each statement of T1 or T2 is executed atomically without interruption.

T1:

```
 a = a + 1;
b = b + 1;
```

T2:

```
 b = 2 * b;
a = 2 * a;
```

Which one of the following options lists all the possible combinations of values of a and b after both T1 and T2 finish execution?

(A) (a = 4, b = 4); (a = 3, b = 3); (a = 4, b = 3)

(B) (a = 3, b = 4); (a = 4, b = 3); (a = 3, b = 3)

(C) (a = 4, b = 4); (a = 4, b = 3); (a = 3, b = 4)

(D) (a = 2, b = 2); (a = 2, b = 3); (a = 3, b = 4)

Official answer: **A**

SOLUTION

1. Name the statements: T1 has s_1: a = a + 1 then s_2: b = b + 1 . T2 has t_1: b = 2 * b then t_2: a = 2 * a . Each statement is atomic, so the outcomes come from the interleavings that keep s_1 before s_2 and t_1 before t_2. There are $\binom{4}{2}=6$ of them. Start with $a=b=1$.
2. $s_1 s_2 t_1 t_2$: $a=2$, $b=2$, then $b=4$, $a=4$. Result $(a,b)=(4,4)$.
3. $s_1 t_1 s_2 t_2$: $a=2$, $b=2$, $b=3$, $a=4$. Result $(4,3)$.
4. $s_1 t_1 t_2 s_2$: $a=2$, $b=2$, $a=4$, $b=3$. Result $(4,3)$.
5. $t_1 s_1 s_2 t_2$: $b=2$, $a=2$, $b=3$, $a=4$. Result $(4,3)$.
6. $t_1 s_1 t_2 s_2$: $b=2$, $a=2$, $a=4$, $b=3$. Result $(4,3)$.
7. $t_1 t_2 s_1 s_2$: $b=2$, $a=2$, $a=3$, $b=3$. Result $(3,3)$.
8. The set of possible final values is $\{(4,4),(4,3),(3,3)\}$, which is option (A).
9. $(3,4)$ never occurs. It would need a to end on the $+1$, so s_1 runs after t_2, giving the order $t_1 t_2 s_1 s_2$. In that order s_2 is the last write to b, so b ends on the $+1$ and equals 3, not 4. Options (B), (C) and (D) each contain a pair that is not in the set, namely $(3,4)$ or $(2,2)$ or $(2,3)$, so they are wrong.

KEY IDEA
List every interleaving that keeps each thread's statement order, which gives 6 of them, and evaluate each one. Only three distinct pairs of final values appear.

WATCH OUT FOR
Assuming each variable can end on either operation independently, so that (3, 4) looks possible. For a to end on +1, T1's first statement must run after T2 has finished. That forces T1's second statement to be the last write to b.

Q.41 MCQ · 2 MARKS

Data Structures and Algorithms › Trees & heaps

An array [82, 101, 90, 11, 111, 75, 33, 131, 44, 93] is heapified. Which one of the following options represents the first three elements in the heapified array?

(A) 82, 90, 101

(B) 82, 11, 93

(C) 131, 11, 93

(D) 131, 111, 90

Official answer: **D**

SOLUTION

1. Build-max-heap (Floyd's bottom-up heapify), 0-indexed, n=10: process nodes from the last internal node index $\lfloor n/2 \rfloor - 1 = 4$ down to index 0, sifting each down against its children (indices $2i+1$, $2i+2$).
2. Start: $[82, 101, 90, 11, 111, 75, 33, 131, 44, 93]$.
3. $i = 4$ (value 111): only child is index 9 (93). Since $111 > 93$, no swap.
4. $i = 3$ (value 11): children are index 7 (131) and index 8 (44); larger child 131 exceeds 11, so swap → $[82, 101, 90, 131, 111, 75, 33, 11, 44, 93]$. Sifting continues at index 7, which has no children, so it stops.
5. $i = 2$ (value 90): children are index 5 (75) and index 6 (33); 90 already exceeds both, so no swap.
6. $i = 1$ (value 101): children are index 3 (131) and index 4 (111); larger child 131 exceeds 101, so swap → $[82, 131, 90, 101, 111, 75, 33, 11, 44, 93]$. Sifting continues at index 3 (now 101): children index 7 (11) and index 8 (44); 101 exceeds both, so it stops.
7. $i = 0$ (value 82): children are index 1 (131) and index 2 (90); larger child 131 exceeds 82, so swap → $[131, 82, 90, 101, 111, 75, 33, 11, 44, 93]$. Sifting continues at index 1 (now 82): children index 3 (101) and index 4 (111); larger child 111 exceeds 82, so swap → $[131, 111, 90, 101, 82, 75, 33, 11, 44, 93]$. Sifting continues at index 4 (now 82): its only child, index 9 (93), exceeds 82, so swap → $[131, 111, 90, 101, 93, 75, 33, 11, 44, 82]$. Index 9 has no children, so it stops.
8. Final heapified array: $[131, 111, 90, 101, 93, 75, 33, 11, 44, 82]$.
9. The first three elements are $131, 111, 90$, matching option (D).

KEY IDEA
Building a max-heap from an unordered array (Floyd's build-heap / bottom-up heapify) processes nodes from the last internal node (index $\lfloor n/2 \rfloor - 1$) down to the root (index 0), sift-down (percolate-down) each one against its children, swapping with the larger child whenever a child exceeds the parent, and continuing the sift-down at the new position until no child is larger.

WATCH OUT FOR
Sifting down only once per node instead of continuing the sift-down after a swap (e.g. stopping after swapping index 0 with index 1 without also sifting the displaced value further down through index 4), or processing nodes top-down instead of bottom-up, both of which leave the heap property violated deeper in the tree and can change which elements land at the top.

Q.42 MCQ · 2 MARKS

Data Structures and Algorithms › Complexity

Consider the following recurrence relation:

$$T(n) = \begin{cases} \sqrt{n}\,T(\sqrt{n}) + n & \text{for } n \geq 1, \\ 1 & \text{for } n = 1. \end{cases}$$

Which one of the following options is CORRECT?

(A) $T(n) = \Theta(n \log \log n)$

(B) $T(n) = \Theta(n \log n)$

(C) $T(n) = \Theta(n^2 \log n)$

(D) $T(n) = \Theta(n^2 \log \log n)$

Official answer: **A**

SOLUTION

1. Substitute $n = 2^m$, so $S(m) := T(2^m) = T(n)$, and $\sqrt{n} = 2^{m/2}$, so $T(\sqrt{n}) = S(m/2)$.
2. The recurrence $T(n) = \sqrt{n}\,T(\sqrt{n}) + n$ becomes $S(m) = 2^{m/2} S(m/2) + 2^m$.
3. Divide both sides by 2^m: $\dfrac{S(m)}{2^m} = \dfrac{S(m/2)}{2^{m/2}} + 1$. Let $R(m) := S(m)/2^m$; then $R(m) = R(m/2) + 1$.
4. This is a Master-theorem case with $a = 1$, $b = 2$, $f(m) = \Theta(1)$: since $\log_b a = 0$ matches the degree of $f(m)$, $R(m) = \Theta(\log m)$.
5. So $S(m) = 2^m \cdot \Theta(\log m)$. Since $m = \log n$ and $2^m = n$, this gives $T(n) = n \cdot \Theta(\log(\log n)) = \Theta(n \log \log n)$.
6. This matches option (A).

KEY IDEA

A recurrence with a $\sqrt{n}$ argument is solved by the double substitution $n = 2^m$ (turning $\sqrt{n}$ into a halving of m), then dividing through by the multiplicative factor $2^{m/2}$ so the recurrence becomes a plain additive one in m, solvable directly or via the Master theorem.

WATCH OUT FOR

Stopping after finding $T(n) = n \cdot \Theta(\log m)$ and forgetting to re-substitute $m = \log n$ into the n-side, so students report $\Theta(n \log n)$ (option B) instead of correctly reducing $\log m = \log(\log n)$ to get $\Theta(n \log \log n)$.

Q.43 MCQ · 2 MARKS

Data Structures and Algorithms › Trees & heaps

Consider a binary min-heap containing 105 distinct elements. Let k be the index (in the underlying array) of the maximum element stored in the heap. The number of possible values of k is

(A) 53

(B) 52

(C) 27

(D) 1

Official answer: **A**

SOLUTION

1. Min-heap property: for any parent–child pair, key(parent) ≤ key(child). If a node has at least one child, that child's key is ≥ the node's key, and since all 105 elements are distinct, the child's key must be strictly greater. So a node with a child can never hold the maximum element.
2. Therefore the maximum element must sit at a node with no children: a leaf of the heap.
3. Using 1-indexed array representation for $n = 105$ nodes, node i has a child at $2i$, which exists only when $2i \leq n$, i.e. $i \leq \lfloor n/2 \rfloor = \lfloor 105/2 \rfloor = 52$. So indices 1 through 52 are internal (non-leaf) nodes.

4. Hence leaf indices run from 53 to 105, giving $105 - 53 + 1 = 53$ leaf positions.
5. For each of these 53 leaf positions, a valid min-heap can be constructed with the maximum element placed at that leaf: put the largest value there and arrange the remaining 104 values anywhere else consistent with the min-heap order (always possible, since a leaf imposes no downward constraint and its parent only needs to be $\leq$ the maximum value, which trivially holds). So every leaf index is achievable.
6. Therefore the number of possible values of k equals the number of leaf positions $= 53$, which is option (A).

KEY IDEA

In a min-heap the maximum element can never occupy any internal (non-leaf) node, because every internal node is $\leq$ each of its children, so a node with a child cannot hold the overall maximum; hence the maximum must sit at some leaf, and every leaf position is achievable by some valid heap on the multiset.

WATCH OUT FOR

Assuming the maximum must be at the "last" leaf or at one of the two children of the root (confusing max-heap intuition), or miscounting leaves by using $n/2$ instead of $\lfloor n/2 \rfloor$ for the boundary between internal nodes and leaves, or forgetting to check that every leaf position, not just some of them, is actually achievable.

Q.44 MSQ · 2 MARKS

Database Management and Warehousing › Normal forms

The symbol $\rightarrow$ indicates functional dependency in the context of a relational database. Which of the following options is/are TRUE?

(A) $(X, Y) \rightarrow (Z, W)$ implies $X \rightarrow (Z, W)$

(B) $(X, Y) \rightarrow (Z, W)$ implies $(X, Y) \rightarrow Z$

(C) $((X, Y) \rightarrow Z$ and $W \rightarrow Y)$ implies $(X, W) \rightarrow Z$

(D) $(X \rightarrow Y$ and $Y \rightarrow Z)$ implies $X \rightarrow Z$

Official answer: **B, C, D**

SOLUTION

1. Check each option with Armstrong's axioms (reflexivity, augmentation, transitivity) and the rules derived from them (decomposition, union).
2. (A) $XY \rightarrow ZW$ says that X and Y together fix Z and W. It does not say that X alone does. Counterexample: take a relation with the rows $(X, Y, Z, W) = (1, 1, 1, 1)$ and $(1, 2, 2, 2)$. The two rows differ on Y, so $XY \rightarrow ZW$ holds. The two rows agree on X but differ on Z, so $X \rightarrow ZW$ fails. A is false.
3. (B) Decomposition: from $XY \rightarrow ZW$ we get $XY \rightarrow Z$ and $XY \rightarrow W$. B is true.
4. (C) Given $W \rightarrow Y$, augment with X to get $XW \rightarrow XY$. Given $XY \rightarrow Z$, transitivity gives $XW \rightarrow Z$. C is true.
5. (D) Transitivity: $X \rightarrow Y$ and $Y \rightarrow Z$ give $X \rightarrow Z$. D is true.
6. Options B, C and D are true. Option A is false.

KEY IDEA

A functional dependency can be weakened on the right side (decomposition) and chained (transitivity, with augmentation), but it cannot be weakened on the left side: dropping an attribute from the determinant is not allowed.

WATCH OUT FOR

Treating $(X, Y) \rightarrow (Z, W)$ as implying $X \rightarrow (Z, W)$, that is, dropping an attribute from the left side. This is not an Armstrong rule. Only the right side can be split.

Q.45 MSQ · 2 MARKS

Data Structures and Algorithms › Graph algorithms

Let G be a directed graph and T a depth first search (DFS) spanning tree in G that is rooted at a vertex v. Suppose T is also a breadth first search (BFS) tree in G, rooted at v. Which of the following statements is/are TRUE for every such graph G and tree T ?

(A) There are no back-edges in G with respect to the tree T

(B) There are no cross-edges in G with respect to the tree T

(C) There are no forward-edges in G with respect to the tree T

(D) The only edges in G are the edges in T

Official answer: **C**

SOLUTION

1. Since T is simultaneously the DFS tree and the BFS tree of G rooted at v, the depth of any vertex in T equals its BFS level, i.e. its shortest-path distance from v in G.
2. BFS invariant: for every edge (u, w) of G (tree or non-tree), $\text{level}(w) \leq \text{level}(u) + 1$; otherwise going through u would give a shorter path to w than its assigned level.
3. Forward-edge test (option C): a forward edge (u, w) has w as a proper descendant of u that is not its child, so $\text{depth}(w) \geq \text{depth}(u) + 2$. Since depth = level here, this forces $\text{level}(w) \geq \text{level}(u) + 2$, contradicting the BFS invariant. So a forward edge can never occur; statement (C) is TRUE for every valid (G, T).
4. Back-edge test (option A): a back edge (u, w) has w as an ancestor of u, so $\text{level}(w) \leq \text{level}(u)$, which never violates the invariant $\text{level}(w) \leq \text{level}(u) + 1$. So back edges are not ruled out automatically: a concrete example is needed.
5. Counterexample graph: vertices v, a, b, c, edges $v \to c,\ v \to a,\ a \to b,\ b \to v,\ b \to c$. DFS from v (visiting c before a): tree edges $v \to c,\ v \to a,\ a \to b$ (this is T); $b \to v$ is a back edge (v is still on the stack), and $b \to c$ is a cross edge (c is already finished).
6. BFS on the same graph with the same adjacency order gives $\text{level}(v) = 0,\ \text{level}(c) = \text{level}(a) = 1,\ \text{level}(b) = 2$, and the BFS tree edges are exactly $v \to c,\ v \to a,\ a \to b$ (the same T). Both non-tree edges satisfy the BFS invariant ($0 \leq 2 + 1$ and $1 \leq 2 + 1$), so T is simultaneously a valid DFS tree and a valid BFS tree of this G.
7. This T contains the back edge $b \to v$, so statement (A) is FALSE for this instance; hence not true for every (G, T).
8. The same T contains the cross edge $b \to c$, so statement (B) is also FALSE for this instance; hence not true for every (G, T).
9. The same example has non-tree edges ($b \to v$, $b \to c$), so statement (D) ('the only edges in G are the edges in T') is false in general.
10. Hence, of the four statements, only (C) holds for every graph G and tree T satisfying the hypothesis.

KEY IDEA
When one tree is simultaneously a DFS tree and a BFS tree rooted at v, tree-depth equals BFS-level for every vertex. The BFS invariant level(w) ≤ level(u)+1 (for every edge (u,w) of G) then rules out forward edges outright, since a forward edge would force level(w) ≥ level(u)+2, but it does not rule out back edges or cross edges, both of which are easy to realize with an explicit small graph.

WATCH OUT FOR
Assuming that once T is forced to be both the DFS and BFS tree, ALL non-tree edges are excluded (picking only C, or picking C together with A on the mistaken belief that DFS-of-a-tree-that-is-also-BFS can only ever have back edges). The safe check is to actually build a small counterexample graph and see which of A, B, D can be made false: only the forward-edge argument (C) is a genuine, unconditional proof; A, B and D each need, and have, explicit counterexamples.

Q.46 MSQ · 2 MARKS

Database Management and Warehousing › Transactions

Consider the following read-write schedule S over three transactions T_1, T_2, and T_3, where the subscripts in the schedule indicate transaction IDs:

$S : r_1(z);\ w_1(z);\ r_2(x);\ r_3(y);\ w_3(y);\ r_2(y);\ w_2(x);\ w_2(y);$

Which of the following transaction schedules is/are conflict equivalent to S ?

(A) $T_1T_2T_3$

(B) $T_1T_3T_2$

(C) $T_3T_2T_1$

(D) $T_3T_1T_2$

Official answer: **B, C, D**

SOLUTION

1. Two operations conflict if they belong to different transactions, access the same item, and at least one is a write. A serial schedule is conflict equivalent to S if it orders every conflicting pair the same way as S.
2. Item z: only T_1 touches it ($r_1(z)$, $w_1(z)$), so there is no conflict.
3. Item x: only T_2 touches it ($r_2(x)$, $w_2(x)$), so there is no conflict.
4. Item y: the operations in S are $r_3(y), w_3(y), r_2(y), w_2(y)$, in that order. The conflicting pairs from different transactions are $r_3(y)$ before $w_2(y)$, $w_3(y)$ before $r_2(y)$ and $w_3(y)$ before $w_2(y)$. Every one of them puts T_3 before T_2.
5. Precedence graph: the only edge is $T_3 \rightarrow T_2$. T_1 is isolated, so it can go anywhere in the order.
6. Serial orders with T_3 before T_2: $T_1T_3T_2$, $T_3T_1T_2$, $T_3T_2T_1$. These are options B, D and C.
7. Option A, $T_1T_2T_3$, puts T_2 before T_3, which reverses the y conflicts, so it is not conflict equivalent to S.

KEY IDEA
Draw the precedence graph. Only item y has conflicts between different transactions, and they give the single edge $T_3 \rightarrow T_2$. Every serial order that respects that edge is conflict equivalent to S, and T_1 can sit anywhere.

WATCH OUT FOR
Seeing T_1 appear first in S and assuming it must come first in the serial order. T_1 only touches z, so it conflicts with nobody and can go anywhere. The other slip is missing that $w_3(y)$ before $r_2(y)$ is also a conflict.

Q.47 MSQ · 2 MARKS

Digital Electronics › Boolean logic

Consider a Boolean expression given by $F(X, Y, Z) = \sum(3, 5, 6, 7)$.

Which of the following statements is/are CORRECT?

(A) $F(X, Y, Z) = \prod(0, 1, 2, 4)$

(B) $F(X,Y,Z) = XY + YZ + XZ$

(C) $F(X,Y,Z)$ is independent of input Y

(D) $F(X,Y,Z)$ is independent of input X

Official answer: **A, B**

SOLUTION

1. Take X as the most significant bit. $F(X,Y,Z) = \sum m(3,5,6,7)$, so $F = 1$ on $011, 101, 110, 111$ and $F = 0$ on minterms $0, 1, 2, 4$.

X	Y	Z	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

2. (A) A product of maxterms $\prod M(\dots)$ lists the minterm numbers where $F = 0$. Those are $0, 1, 2, 4$, so $F = \prod(0,1,2,4)$. **Correct.**
3. (B) On the K-map, the 1s at minterms 3 and 7 form the pair YZ. The 1s at 5 and 7 form the pair XZ. The 1s at 6 and 7 form the pair XY. These three pairs cover all four 1s, so $F = XY + YZ + XZ$ (the majority function). Each product term has $F = 1$ only on minterms in the set, so no extra minterm is introduced. **Correct.**
4. (C) Flip Y and compare: $F(0,0,1) = 0$ but $F(0,1,1) = 1$. F changes with Y, so it depends on Y. **Incorrect.**
5. (D) Flip X and compare: $F(0,1,0) = 0$ but $F(1,1,0) = 1$. F changes with X, so it depends on X. **Incorrect.**
6. The correct statements are (A) and (B).

KEY IDEA
A function given as $\sum m$ equals $\prod M$ over the missing minterm numbers. The function is the 3-input majority function, which depends on every input.

WATCH OUT FOR
Reading $\prod(0,1,2,4)$ as a list of minterms where $F = 1$, or taking the pair terms XY, YZ, XZ to mean the function ignores one variable.

Q.48 MSQ · 2 MARKS

Programming in C › C basics

Consider the following C function definition.

```
int f(int x, int y) {
 for (int i=0; i<y; i++) {
   x=x+x+y;
 }
 return x;
}
```

Which of the following statements is/are TRUE about the above function?

(A) If the inputs are x=20, y=10, then the return value is greater than 2^{20}

(B) If the inputs are x=20, y=20, then the return value is greater than 2^{20}

(C) If the inputs are x=20, y=10, then the return value is less than 2^{10}

(D) If the inputs are x=10, y=20, then the return value is greater than 2^{20}

Official answer: **B, D**

SOLUTION

1. The `for (int i=0; i<y; i++)` loop executes exactly y times (i = 0 to y−1), and each iteration performs `x=x+x+y`, i.e. $x \leftarrow 2x + y$.
2. This defines the linear recurrence $x_{n+1} = 2x_n + y$ with $x_0 = x$ (the input value of x).
3. Solving by unrolling/induction: $x_1 = 2x_0 + y$, $x_2 = 4x_0 + 3y$, and in general $x_n = 2^n x_0 + y\,(2^n - 1)$.
4. So the value returned after the loop (n = y iterations) is
$$x_y = 2^y x_0 + y\,(2^y - 1)\,.$$
5. (A) x=20, y=10: $x_{10} = 2^{10}(20) + 10(2^{10} - 1) = 20480 + 10230 = 30710$. Since $2^{20} = 1{,}048{,}576$ and $30710 < 2^{20}$, statement (A) is FALSE.
6. (B) x=20, y=20: $x_{20} = 2^{20}(20) + 20(2^{20} - 1) = 20{,}971{,}520 + 20{,}971{,}500 = 41{,}943{,}020$, which is greater than $2^{20} = 1{,}048{,}576$. Statement (B) is TRUE.
7. (C) x=20, y=10: $x_{10} = 30710$ (from step 5), and $2^{10} = 1024$. Since $30710 > 1024$, the claim 'less than 2^{10}' is FALSE.
8. (D) x=10, y=20: $x_{20} = 2^{20}(10) + 20(2^{20} - 1) = 10{,}485{,}760 + 20{,}971{,}500 = 31{,}457{,}260$, which is greater than $2^{20} = 1{,}048{,}576$. Statement (D) is TRUE.
9. Only (B) and (D) are TRUE.

KEY IDEA

The loop runs exactly y times, and each iteration replaces `x` by `x=x+x+y`, i.e. the linear recurrence $x_{n+1} = 2x_n + y$. Unrolling this recurrence gives a closed form $x_n = 2^n x_0 + y(2^n - 1)$, which lets every option be checked by a single arithmetic comparison instead of by re-running the loop by hand.

WATCH OUT FOR

Reading `x=x+x+y` as `x = 2x` and forgetting the `+y` term each iteration (or, symmetrically, treating the recurrence as $x_{n+1} = 2(x_n + y)$ instead of $2x_n + y$), which changes the growth rate and can flip (D) to false when it is actually true; also common is comparing the final value to 2^y instead of the specific power asked for in each option (e.g. checking against 2^{10} for option A instead of 2^{20}).

Q.49 MSQ · 2 MARKS

Engineering Mathematics › Linear algebra

Let A be any $n \times m$ matrix, where $m > n$. Which of the following statements is/are TRUE about the system of linear equations $A\mathbf{x} = \mathbf{0}$?

(A) There exist at least $m - n$ linearly independent solutions to this system

(B) There exist $m - n$ linearly independent vectors such that every solution is a linear combination of these vectors

(C) There exists a non-zero solution in which at least $m - n$ variables are 0

(D) There exists a solution in which at least n variables are non-zero

Official answer: **A**

SOLUTION

1. Rank–nullity: $\operatorname{rank}(A) + \dim N(A) = m$. Since A has n rows, $\operatorname{rank}(A) \le n$, so $\dim N(A) = m - \operatorname{rank}(A) \ge m - n \ge 1$.
2. (A) The null space has dimension at least $m - n$, so it contains at least $m - n$ linearly independent solutions. TRUE for every such A.
3. (B) This needs the solutions to be spanned by exactly $m - n$ vectors, i.e. $\dim N(A) = m - n$, which holds only when $\operatorname{rank}(A) = n$. For $A = 0$ the nullity is $m > m - n$, so no $m - n$ vectors span the

solution set. FALSE in general.

4. (C) Take $A = \begin{bmatrix} 1 & 1 \end{bmatrix}$ ($n = 1, m = 2$). Every non-zero solution is $(t, -t)$ with $t \neq 0$, so none has $m - n = 1$ zero variable. FALSE in general.
5. (D) Take $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ ($n = 2, m = 3$). The solutions are $(0, 0, t)$, which have at most 1 non-zero variable, fewer than $n = 2$. FALSE in general.
6. Only (A) holds for every $n \times m$ matrix with $m > n$.

KEY IDEA

By rank–nullity, the solution space of Ax = 0 has dimension m − rank(A) ≥ m − n, so it always holds at least m − n independent solutions. Any statement that needs exactly m − n, or a particular zero pattern, can fail for a special A.

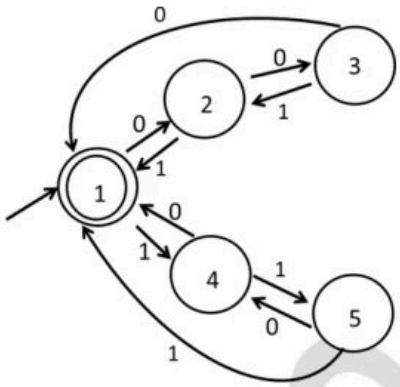
WATCH OUT FOR

Reading (B) as true because the generic case rank(A) = n gives nullity exactly m − n. The statement must hold for any A, and a rank-deficient A such as the zero matrix breaks it.

Q.50 MSQ · 2 MARKS

Theory of Computation › Finite automata

Q.50 Consider the 5-state DFA M accepting the language $L(M) \subset (0+1)^*$ shown below. For any string $w \in (0+1)^*$ let $n_0(w)$ be the number of $0's$ in w and $n_1(w)$ be the number of $1's$ in w.



Which of the following statements is/are FALSE?

(A) States 2 and 4 are distinguishable in M

(B) States 3 and 4 are distinguishable in M

(C) States 2 and 5 are distinguishable in M

(D) Any string w with $n_0(w) = n_1(w)$ is in $L(M)$

Official answer: **B, C**

SOLUTION

1. Read the transitions from the figure (start state 1, the only accepting state): $\delta(1,0) = 2,\ \delta(1,1) = 4$; $\delta(2,0) = 3,\ \delta(2,1) = 1$; $\delta(3,0) = 1,\ \delta(3,1) = 2$; $\delta(4,0) = 1,\ \delta(4,1) = 5$; $\delta(5,0) = 4,\ \delta(5,1) = 1$.
2. Let $d(w) = (n_0(w) - n_1(w)) \bmod 3$. Assign states 1, 2, 3 to $d = 0, 1, 2$ and states 4, 5 to $d = 2, 1$. Every transition then agrees: a 0 adds 1 to d and a 1 subtracts 1. For example, $\delta(1,1) = 4$ gives $0 - 1 \equiv 2$; $\delta(4,1) = 5$ gives $2 - 1 = 1$; $\delta(5,1) = 1$ gives $1 - 1 = 0$; $\delta(4,0) = 1$ gives $2 + 1 \equiv 0$; $\delta(5,0) = 4$ gives $1 + 1 = 2$; $\delta(3,1) = 2$ gives $2 - 1 = 1$. So $L(M) = \{w : n_0(w) - n_1(w) \equiv 0 \pmod 3\}$.

3. States with the same d are equivalent. States 3 and 4 both have $d = 2$: $\delta(3, \cdot) = (1, 2)$ and $\delta(4, \cdot) = (1, 5)$, and $2 \equiv 5$ (both $d = 1$). States 2 and 5 both have $d = 1$: $\delta(2, \cdot) = (3, 1)$ and $\delta(5, \cdot) = (4, 1)$, and $3 \equiv 4$. The minimal DFA has 3 states.
4. (A) States 2 ($d = 1$) and 4 ($d = 2$) differ: the string 1 takes 2 to the accepting state 1, but takes 4 to the non-accepting state 5. They are distinguishable, so the statement is true.
5. (B) States 3 and 4 are equivalent, so they are not distinguishable. The statement is FALSE.
6. (C) States 2 and 5 are equivalent, so they are not distinguishable. The statement is FALSE.
7. (D) If $n_0(w) = n_1(w)$ then $d = 0$, so w ends in state 1 and is accepted. The statement is true.
8. The FALSE statements are (B) and (C).

KEY IDEA

The DFA tracks $(n_0 - n_1) \bmod 3$ using five states, but only three residues matter. States with the same residue are equivalent, and states with different residues are distinguishable.

WATCH OUT FOR

Treating the five states as five distinct classes and calling every pair distinguishable. Another slip is picking the true statements (A, D) when the question asks for the FALSE ones.

Q.51 MSQ · 2 MARKS

Discrete Mathematics › Graph theory

The chromatic number of a graph is the minimum number of colours used in a proper colouring of the graph. Let G be any graph with n vertices and chromatic number k. Which of the following statements is/are always TRUE?

(A) G contains a complete subgraph with k vertices

(B) G contains an independent set of size at least n/k

(C) G contains at least $k(k-1)/2$ edges

(D) G contains a vertex of degree at least k

Official answer: **B, C**

SOLUTION

1. Take a proper colouring of G with exactly k colours, where $k = \chi(G)$. Each colour class is an independent set, and the k classes partition the n vertices.
2. (A) Test with the odd cycle C_5: $\chi(C_5) = 3$, but C_5 has no triangle, so it has no K_3. (A) is not always true.
3. (B) The k classes cover n vertices, so by the pigeonhole principle one class has at least $\lceil n/k \rceil \geq n/k$ vertices. That class is an independent set. (B) is always true.
4. (C) Suppose two colour classes had no edge between them. Merging them would give a proper colouring with $k - 1$ colours, which contradicts $\chi(G) = k$.
5. So every pair of the k classes has at least one edge between them. The pairs of classes number $\binom{k}{2} = \frac{k(k-1)}{2}$, and the edges counted for different pairs are distinct. Hence G has at least $\frac{k(k-1)}{2}$ edges. (C) is always true.
6. (D) Test with K_k: $\chi(K_k) = k$, but every vertex has degree $k - 1 < k$. (D) is not always true.
7. Hence the correct options are (B) and (C).

KEY IDEA
In a colouring with $\chi(G) = k$ colours, each colour class is an independent set. By pigeonhole, the largest class has at least n/k vertices. Minimality of k forces an edge between every two classes, so G has at least $\binom{k}{2}$ edges.

WATCH OUT FOR
Assuming a high chromatic number forces a large clique (A) or a high-degree vertex (D). Odd cycles have $\chi = 3$ with no triangle, and K_k has maximum degree only $k - 1$.

Q.52 MSQ · 2 MARKS

Discrete Mathematics › Groups

Consider the operators ◊ and □ defined by $a \diamondsuit b = a + 2b$, $a \square b = ab$, for positive integers. Which of the following statements is/are TRUE?

(A) Operator ◊ obeys the associative law

(B) Operator □ obeys the associative law

(C) Operator ◊ over the operator □ obeys the distributive law

(D) Operator □ over the operator ◊ obeys the distributive law

Official answer: **B, D**

SOLUTION

1. Write $a \diamondsuit b = a + 2b$ and $a \square b = ab$ on positive integers. Test each law with general symbols, and use a counterexample to refute a law.
2. (A) Associativity of $\diamondsuit$: $(a \diamondsuit b) \diamondsuit c = (a + 2b) + 2c = a + 2b + 2c$, while $a \diamondsuit (b \diamondsuit c) = a + 2(b + 2c) = a + 2b + 4c$. These differ because $c \neq 0$. For example $(1 \diamondsuit 1) \diamondsuit 1 = 3 \diamondsuit 1 = 5$ but $1 \diamondsuit (1 \diamondsuit 1) = 1 \diamondsuit 3 = 7$. So (A) is false.
3. (B) Associativity of □: $(a \square b) \square c = (ab)c = a(bc) = a \square (b \square c)$, because multiplication of integers is associative. So (B) is true.
4. (C) $\diamondsuit$ over □ requires $a \diamondsuit (b \square c) = (a \diamondsuit b) \square (a \diamondsuit c)$. The left side is $a + 2bc$; the right side is $(a + 2b)(a + 2c) = a^2 + 2ab + 2ac + 4bc$. With $a = b = c = 1$ the left side is 3 and the right side is 9. So (C) is false.
5. (D) □ over $\diamondsuit$ requires $a \square (b \diamondsuit c) = (a \square b) \diamondsuit (a \square c)$. The left side is $a(b + 2c) = ab + 2ac$; the right side is $ab + 2(ac) = ab + 2ac$. They are equal for all a, b, c. Since □ is commutative, the right-distributive form $(b \diamondsuit c) \square a = (b \square a) \diamondsuit (c \square a)$ holds too. So (D) is true.
6. The true statements are (B) and (D).

KEY IDEA
To test a law, expand both sides with general symbols and compare. One counterexample is enough to refute a law. Distributivity of one operator over another needs the left side to expand to exactly the right side.

WATCH OUT FOR
Calling $\diamondsuit$ associative because $a + 2b$ looks like addition, when the factor 2 on the second argument breaks it. Another slip is expanding $(a + 2b)(a + 2c)$ wrongly in (C) and finding a match.

Q.53 MSQ · 2 MARKS

Computer Organization and Architecture › Memory & cache

Consider two set-associative cache memory architectures: **WBC**, which uses the write back policy, and **WTC**, which uses the write through policy. Both of them use the LRU (*Least Recently Used*) block replacement policy. The cache memory is connected to the main memory. Which of the following statements is/are TRUE?

(A) A read miss in WBC never evicts a dirty block

(B) A read miss in WTC never triggers a write back operation of a cache block to main memory

(C) A write hit in WBC can modify the value of the dirty bit of a cache block

(D) A write miss in WTC always writes the victim cache block to main memory before loading the missed block into the cache

Official answer: **B, C**

SOLUTION

1. A read miss in WBC (write-back cache): LRU replacement picks the victim purely by recency, with no exemption for dirty blocks, so a dirty block CAN be evicted (triggering a write-back first). Statement (A) ("never evicts a dirty block") is FALSE.
2. A read miss in WTC (write-through cache): every write is immediately propagated to main memory, so no cache block in WTC is ever dirty. Evicting any block on a read miss therefore never requires a write-back to memory. Statement (B) is TRUE.
3. A write hit in WBC: the write updates the cached copy only (not memory), so the block's dirty bit is set to 1 to record that it now differs from memory and must be written back later. This is a modification of the dirty bit's value. Statement (C) is TRUE.
4. A write miss in WTC: since (as in step 2) no WTC block is ever dirty, the victim block being replaced already matches main memory, so it does NOT need to be written to memory before the new block is loaded. Statement (D) ("always writes the victim… before loading") is FALSE.
5. Hence the TRUE statements are (B) and (C).

KEY IDEA

In a write-back cache, memory is updated only when a dirty (modified) block is evicted, so the cache's dirty bit tracks which blocks are out of sync with memory; in a write-through cache every write goes to memory immediately, so no cache block is ever dirty and eviction never needs a write-back.

WATCH OUT FOR

Assuming LRU somehow "protects" dirty blocks from being chosen as the eviction victim on a read miss (it does not: LRU picks purely by recency, so a dirty block can absolutely be evicted, triggering a write-back first), and conversely forgetting that in write-through caches no block is ever dirty, so eviction is a silent, cost-free operation with respect to memory.

Q.54 NAT · 2 MARKS

Computer Organization and Architecture › Memory & cache

Consider a 512 GB hard disk with 32 storage surfaces. There are 4096 sectors per track and each sector holds 1024 bytes of data. The number of cylinders in the hard disk is

Official answer: **4096**

SOLUTION

1. Bytes per track = (sectors per track) × (bytes per sector) = 4096 × 1024 = 2^{12} \times 2^{10} = 2^{22} \text{ bytes}.
2. Disk geometry: one cylinder consists of the same track on every surface, so total capacity = (number of cylinders) × (number of surfaces) × (bytes per track).
3. 512 GB = 512 \times 2^{30} = 2^{9} \times 2^{30} = 2^{39} \text{ bytes} (using 1 GB = 2^{30} bytes).
4. Bytes per cylinder across all surfaces = 32 \times 2^{22} = 2^{5} \times 2^{22} = 2^{27} \text{ bytes}.
5. Number of cylinders = \dfrac{2^{39}}{2^{27}} = 2^{12} = 4096.

KEY IDEA
Disk capacity factors as (cylinders) × (surfaces, i.e. tracks per cylinder) × (sectors per track) × (bytes per sector). Compute bytes per track, multiply by the number of surfaces to get bytes per cylinder (across all surfaces), then divide total capacity by that to get the number of cylinders. Keeping everything in powers of 2 avoids arithmetic slips.

WATCH OUT FOR
Forgetting to divide by the number of surfaces (computing cylinders as total bytes ÷ bytes-per-track only, i.e. treating the disk as single-surface), or using 1 GB = 10^9 bytes instead of the GATE convention 2^30 bytes, both of which give a wrong power of two.

Q.55 NAT · 2 MARKS

Q.55 The baseline execution time of a program on a 2 GHz single core machine is 100 nanoseconds (ns). The code corresponding to 90% of the execution time can be fully parallelized. The overhead for using an additional core is 10 ns when running on a multicore system. Assume that all cores in the multicore system run their share of the parallelized code for an equal amount of time.

The number of cores that minimize the execution time of the program is _____

Official answer: **3**

Q.56 NAT · 2 MARKS

A given program has 25% load/store instructions. Suppose the ideal CPI (cycles per instruction) without any memory stalls is 2. The program exhibits 2% miss rate on instruction cache and 8% miss rate on data cache. The miss penalty is 100 cycles. The speedup (rounded off to two decimal places) achieved with a perfect cache (i.e., with NO data or instruction cache misses) is _____

Official answer: **3.00**

SOLUTION

1. CPI_ideal (with a perfect cache, i.e. no memory stalls at all) is given directly in the stem: $\text{CPI}_{ideal} = 2$.
2. Instruction-cache stall cycles per instruction: every instruction must be fetched, so the instruction-cache miss rate applies to all instructions. $\text{Stalls}_{instr} = 0.02 \times 100 = 2$ cycles per instruction.
3. Data-cache stall cycles per instruction: only load/store instructions access the data cache, so the data-cache miss rate is scaled by the load/store fraction. $\text{Stalls}_{data} = 0.25 \times 0.08 \times 100 = 2$ cycles per instruction.
4. Real CPI (with the actual instruction and data caches, misses included) is the ideal CPI plus both stall contributions:
$$\text{CPI}_{real} = \text{CPI}_{ideal} + \text{Stalls}_{instr} + \text{Stalls}_{data} = 2 + 2 + 2 = 6$$
5. Since the clock cycle time and the instruction count are unchanged between the real machine and the perfect-cache machine, execution time is proportional to CPI. So the speedup from a perfect cache is
$$\text{Speedup} = \frac{\text{CPI}_{real}}{\text{CPI}_{ideal}} = \frac{6}{2} = 3.00$$

KEY IDEA
Memory stall cycles per instruction are added on top of the ideal (memory-stall-free) CPI to get the real CPI; speedup from a perfect cache is the ratio of real CPI to ideal CPI, since clock rate and instruction count are unchanged.

WATCH OUT FOR
Applying the data-cache miss rate to every instruction instead of scaling it by the 25% load/store fraction (or forgetting to add both instruction-cache and data-cache stalls), which gives a wrong real CPI and hence a wrong speedup.

Q.57 NAT · 2 MARKS

Operating Systems › Processes & threads

Consider the following code snippet using the fork() and wait() system calls. Assume that the code compiles and runs correctly, and that the system calls run successfully without any errors.

```
int x = 3;
while(x > 0) {
 fork();
 printf("hello");
 wait(NULL);
 x--;
}
```

The total number of times the printf statement is executed is _____

Official answer: **14**

SOLUTION

1. Each `fork()` creates a child that is an exact copy of the parent, including the current value of `x`. After the fork, the parent and the child each run `printf`, then `wait(NULL)`, then `x--`, then test the loop condition again.
2. Let $f(x)$ be the total number of `printf` executions made by one process that enters the loop with value x, counting everything its descendants print. When $x = 0$ the loop does not run, so $f(0) = 0$.
3. In one iteration, the parent and the child each execute `printf` once. That gives 2 prints.
4. After the iteration both processes hold $x - 1$ and both return to the loop test. Each of them then produces $f(x - 1)$ prints. `wait(NULL)` only delays the parent until its child ends (the child, having no children yet, gets an immediate return from its own `wait`). It does not change the count.
5. So $f(x) = 2 + 2f(x - 1)$.
6. $f(1) = 2 + 2 \cdot 0 = 2$
7. $f(2) = 2 + 2 \cdot 2 = 6$
8. $f(3) = 2 + 2 \cdot 6 = 14$
9. Check by process count: iteration 1 has 1 process, which forks, so 2 prints. Iteration 2 has 2 processes, each forks, so 4 prints. Iteration 3 has 4 processes, each forks, so 8 prints. The total is $2 + 4 + 8 = 14$.

KEY IDEA

Every fork copies the process, including x, so each iteration doubles the number of processes, and every process prints once per iteration. Prints per iteration are 2, 4, 8, which total 14.

WATCH OUT FOR

Counting only the parent's prints, or thinking that `wait(NULL)` stops the child from reaching later iterations. Another slip is to assume the child restarts with the original x. The child keeps the current x and joins the loop with it.

Q.58 NAT · 2 MARKS

Computer Networks › IPv4

Q.58 Consider the entries shown below in the forwarding table of an IP router. Each entry consists of an IP prefix and the corresponding next hop router for packets whose destination IP address matches the prefix. The notation "/N" in a prefix indicates a subnet mask with the most significant N bits set to 1.

Prefix	Next hop router
10.1.1.0/24	R1
10.1.1.128/25	R2
10.1.1.64/26	R3
10.1.1.192/26	R4

This router forwards 20 packets each to 5 hosts. The IP addresses of the hosts are 10.1.1.16, 10.1.1.72, 10.1.1.132, 10.1.1.191, and 10.1.1.205 . The number of packets forwarded via the next hop router R2 is _____

Official answer: **40**

SOLUTION

1. Only the last octet differs, so work out the range each prefix covers in the last octet: $10.1.1.0/24$ covers 0 to 255 (R1); $10.1.1.128/25$ covers 128 to 255 (R2); $10.1.1.64/26$ covers 64 to 127 (R3); $10.1.1.192/26$ covers 192 to 255 (R4).
2. The router uses longest prefix match: among all entries that match a destination, the entry with the largest N is used.
3. 10.1.1.16 (last octet 16) lies only in the /24 range, so it goes to R1.
4. 10.1.1.72 (last octet 72) lies in 64 to 127, so it matches /24 and /26. The longest prefix is /26, so it goes to R3.
5. 10.1.1.132 (last octet 132) lies in 128 to 255 but below 192, so it matches /24 and /25. The longest prefix is /25, so it goes to R2.
6. 10.1.1.191 (last octet 191) is below 192, so it also matches only /24 and /25, and goes to R2.
7. 10.1.1.205 (last octet 205) lies in 192 to 255, so it matches /24, /25 and /26. The longest prefix is /26, so it goes to R4.
8. Two hosts (10.1.1.132 and 10.1.1.191) are forwarded via R2, so the packets via R2 are $2 \times 20 = 40$.

KEY IDEA

Routers pick the matching forwarding entry with the longest prefix. Convert each prefix to a range of the last octet and find the most specific range containing each host.

WATCH OUT FOR

Stopping at the first or shortest matching prefix. 10.1.1.205 also matches the /25 entry (R2), but the /26 entry is longer, so it goes to R4. Counting it under R2 would give 60.

Q.59 NAT · 2 MARKS

Theory of Computation › CFG & PDA

Let $G = (V, \Sigma, S, P)$ be a context-free grammar in Chomsky Normal Form with $\Sigma = \{a, b, c\}$ and V containing 10 variable symbols including the start symbol S. The string $w = a^{30}b^{30}c^{30}$ is derivable from S.

The number of steps (application of rules) in the derivation $S \to^* w$ is _____.

Official answer: **179**

SOLUTION

1. In Chomsky Normal Form every production is either $A \to BC$ (two variables) or $A \to a$ (one terminal).
2. The string has length $|w| = 30 + 30 + 30 = 90$.
3. Each rule $A \to a$ adds one terminal and no other symbol is ever turned into a terminal. So exactly 90 rules of this kind are used, one for each terminal of w.
4. Each rule $A \to BC$ replaces one symbol by two, so it raises the length of the sentential form by exactly 1. Rules $A \to a$ leave the length unchanged. We start from the single symbol S (length 1) and end at length 90, so exactly $90 - 1 = 89$ rules of the form $A \to BC$ are used.
5. Total number of steps $= 89 + 90 = 2|w| - 1 = 2 \times 90 - 1 = 179$.
6. The count depends only on $|w|$. The number of variables (10) and the particular derivation do not change it.

KEY IDEA

In a CNF grammar, deriving a string of length n always takes exactly $n - 1$ rules $A \to BC$ and n rules $A \to a$, so $2n - 1$ steps in all.

WATCH OUT FOR

Counting only the 89 binary rules, or only the 90 terminal rules. Another slip is trying to use the number of variables (10) or the shape $a^{30}b^{30}c^{30}$, neither of which affects the count.

Q.60 NAT · 2 MARKS

Data Structures and Algorithms › Graph algorithms

The number of edges present in the forest generated by the DFS traversal of an undirected graph G with 100 vertices is 40. The number of connected components in G is ____ (answer in integer)

Official answer: **60**

SOLUTION

1. DFS traversal of an undirected graph produces a spanning forest: it visits every vertex, and within each connected component the DFS tree edges form a spanning tree of that component.
2. A spanning tree on a component with k vertices has exactly $k - 1$ tree edges.
3. Summing over all connected components: total forest edges $= \sum_i (k_i - 1) = n - c$, where n is the total number of vertices and c is the number of connected components.
4. Given $n = 100$ and total forest edges $= 40$:
$$40 = 100 - c$$
5. Solving:
$$c = 100 - 40 = 60$$

KEY IDEA

A DFS (or BFS) traversal of an undirected graph produces a spanning forest with exactly one tree per connected component, and a spanning tree on k vertices has exactly $k - 1$ edges.

WATCH OUT FOR

Treating the graph's total edge count as equal to the forest's edge count (ignoring that non-tree/back edges are excluded from the DFS forest), or forgetting the -1 per component and computing $c = n/(\text{edges per component})$ instead of $c = n - (\text{forest edges})$.

Q.61 NAT · 2 MARKS

Theory of Computation › Finite automata

Consider the following two regular expressions over the alphabet $\{0, 1\}$:

$r = 0^* + 1^*$

$s = 01^* + 10^*$

The total number of strings of length less than or equal to 5, which are neither in r nor in s, is _____.

Official answer: **44**

SOLUTION

1. Count all strings over $\{0, 1\}$ of length 0 to 5: $2^0 + 2^1 + \cdots + 2^5 = 2^6 - 1 = 63$.
2. Strings in $r = 0^* + 1^*$ are ε together with the all-0 strings and all-1 strings. Each length 1 to 5 gives 2 such strings, so $|r| = 1 + 2 \times 5 = 11$.
3. Strings in 01^* of length at most 5 are $0, 01, 011, 0111, 01111$, one per length, so 5 strings. Strings in 10^* are $1, 10, 100, 1000, 10000$, also 5 strings.
4. The two parts of s start with different symbols, so they share no string and $|s| = 5 + 5 = 10$.
5. Overlap of s with r: a string of s lies in r only if it uses a single symbol. In 01^* that is only 0, and in 10^* only 1. Every other string of s contains both 0 and 1. So $|r \cap s| = 2$.
6. $|r \cup s| = 11 + 10 - 2 = 19$.
7. Strings of length at most 5 in neither r nor s: $63 - 19 = 44$.

KEY IDEA

Count the complement. Take all 63 strings of length at most 5 and subtract $|r \cup s|$, using inclusion-exclusion with the overlap $\{0, 1\}$.

WATCH OUT FOR

Forgetting ε in r, or forgetting that 0 and 1 belong to both r and s. This counts those two strings twice and gives $|r \cup s| = 21$, so the answer comes out 42.

Q.62 NAT · 2 MARKS

Operating Systems › Memory management

Consider a memory management system that uses a page size of 2 KB. Assume that both the physical and virtual addresses start from 0. Assume that the pages 0, 1, 2, and 3 are stored in the page frames 1, 3, 2, and 0, respectively. The physical address (in decimal format) corresponding to the virtual address 2500 (in decimal format) is _____

Official answer: **6596**

SOLUTION

1. The page size is $2\ \text{KB} = 2 \times 1024 = 2048$ bytes.
2. Page number of virtual address 2500: $\lfloor 2500/2048 \rfloor = 1$.
3. Offset within the page: $2500 \bmod 2048 = 2500 - 2048 = 452$.
4. The page table, read from the question (pages 0, 1, 2, 3 are in frames 1, 3, 2, 0):

Page	Frame
0	1
1	3
2	2
3	0

   Page 1 is therefore in frame 3.
5. Physical address $= \text{frame} \times \text{page size} + \text{offset} = 3 \times 2048 + 452 = 6144 + 452 = 6596$.

KEY IDEA

Split the virtual address into a page number and an offset by dividing by the page size. Look up the frame in the page table. The physical address is frame number times page size plus offset.

WATCH OUT FOR

Using the page number as the frame number (1 x 2048 + 452 = 2500), or reading the frame list in the wrong order. Another slip is taking 2 KB as 2000 bytes instead of 2048.

Q.63 NAT · 2 MARKS

Engineering Mathematics › Probability & statistics

A bag contains 10 red balls and 15 blue balls. Two balls are drawn randomly without replacement. Given that the first ball drawn is red, the probability (rounded off to 3 decimal places) that both balls drawn are red is _____

Official answer: **0.370 to 0.380**

SOLUTION

1. Let R_1 and R_2 be the events that the first and second balls drawn are red. The bag has 25 balls: 10 red and 15 blue. We want $P(R_1 \cap R_2 \mid R_1)$.
2. Since $R_1 \cap R_2 \subseteq R_1$, we have $P(R_1 \cap R_2 \mid R_1) = \dfrac{P(R_1 \cap R_2)}{P(R_1)} = P(R_2 \mid R_1)$.
3. $P(R_1) = \dfrac{10}{25}$ and $P(R_1 \cap R_2) = \dfrac{10}{25} \cdot \dfrac{9}{24}$, because the second draw is made without replacement.
4. Dividing: $\dfrac{\frac{10}{25} \cdot \frac{9}{24}}{\frac{10}{25}} = \dfrac{9}{24}$.
5. Check by direct counting: after one red ball is removed, the bag holds 9 red and 15 blue balls, 24 in all. So the second draw is red with probability $\dfrac{9}{24} = \dfrac{3}{8} = 0.375$.

KEY IDEA

Conditioning on the first ball being red leaves 24 balls, 9 of them red. The required probability is just the chance that the second draw is red from that reduced bag.

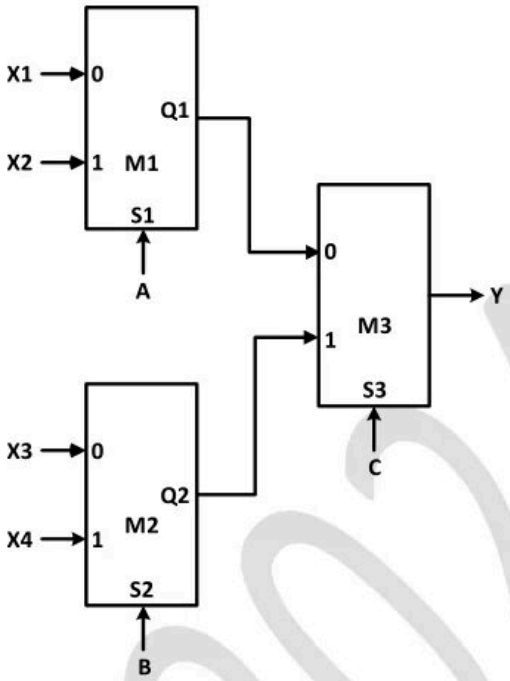
WATCH OUT FOR

Computing the unconditional probability of two reds, $\frac{10}{25} \cdot \frac{9}{24} = 0.15$. The question gives that the first ball is red, so the first factor must not be multiplied in. Another slip is forgetting that the draw is without replacement and using $\frac{10}{25} = 0.4$.

Q.64 NAT · 2 MARKS

Digital Electronics › Combinational

Q.64 Consider a digital logic circuit consisting of three 2-to-1 multiplexers M1, M2, and M3 as shown below. X1 and X2 are inputs of M1. X3 and X4 are inputs of M2. A, B, and C are select lines of M1, M2, and M3, respectively.



For an instance of inputs **X1=1, X2=1, X3=0**, and **X4=0**, the number of combinations of A, B, C that give the output **Y=1** is _____

Official answer: **4**

SOLUTION

1. A 2-to-1 multiplexer passes input 0 when its select is 0 and input 1 when its select is 1. From the figure, $Q_1 = \overline{A}X_1 + AX_2$, $Q_2 = \overline{B}X_3 + BX_4$ and $Y = \overline{C}\,Q_1 + C\,Q_2$.
2. M1: with $X_1 = X_2 = 1$, $Q_1 = \overline{A} + A = 1$ for both values of A.
3. M2: with $X_3 = X_4 = 0$, $Q_2 = 0$ for both values of B.
4. So $Y = \overline{C} \cdot 1 + C \cdot 0 = \overline{C}$.
5. $Y = 1$ exactly when $C = 0$, whatever A and B are.
6. Count the combinations: $C = 0$ is fixed and A, B each take 2 values, so $2 \times 2 = 4$ combinations of (A, B, C).

KEY IDEA

When both data inputs of a multiplexer are equal, its output equals that value no matter what the select line is. So A and B do not matter here, and only C decides Y.

WATCH OUT FOR

Counting only the combinations of C (giving 1), or forgetting that A and B are still free inputs. Each of them takes 2 values, so every choice of A and B counts as a separate combination.

Q.65 NAT · 2 MARKS

Computer Networks › IPv4

Consider sending an IP datagram of size 1420 bytes (including 20 bytes of IP header) from a sender to a receiver over a path of two links with a router between them. The first link (sender to router) has an MTU

(Maximum Transmission Unit) size of 542 bytes, while the second link (router to receiver) has an MTU size of 360 bytes. The number of fragments that would be delivered at the receiver is _____

Official answer: **6**

SOLUTION

1. The datagram carries $1420 - 20 = 1400$ bytes of data. Every fragment has its own 20-byte header, and the data in every fragment except the last must be a multiple of 8 bytes (the fragment offset counts in units of 8 bytes).
2. First link, MTU 542: each fragment has room for $542 - 20 = 522$ bytes of data. The largest multiple of 8 not above 522 is $\lfloor 522/8 \rfloor \times 8 = 65 \times 8 = 520$ bytes.
3. On the first link, $1400 = 520 + 520 + 360$. The router therefore receives 3 fragments, with data sizes 520, 520 and 360 bytes.
4. Second link, MTU 360: each fragment has room for $360 - 20 = 340$ bytes of data. The largest multiple of 8 not above 340 is $\lfloor 340/8 \rfloor \times 8 = 42 \times 8 = 336$ bytes.
5. A router only fragments; it never reassembles. Each incoming fragment is split on its own. $520 = 336 + 184$ gives 2 fragments, and $184 = 23 \times 8$ is a multiple of 8 even though it is not the last fragment overall. The second 520 also gives 2 fragments. $360 = 336 + 24$ gives 2 fragments, the last of which carries the final 24 bytes of the datagram.
6. The receiver gets $2 + 2 + 2 = 6$ fragments, with data sizes $336, 184, 336, 184, 336, 24$. These sum to 1400 bytes, which matches the original data.

KEY IDEA

Fragment on the first link with data in multiples of 8 bytes, giving 520, 520 and 360. The router then fragments each of these on its own for the smaller MTU. It never reassembles, so every first-link fragment is split again, and the receiver sees $2 + 2 + 2 = 6$.

WATCH OUT FOR

Ignoring the first link and fragmenting 1400 bytes directly for MTU 360, which gives $\lceil 1400/336 \rceil = 5$. Another slip is forgetting that fragment data must be a multiple of 8 bytes and using 340 or 522 instead of 336 or 520.

FROM IAE ACADEMY

Preparing for GATE CS?

IAE Academy teaches GATE CS in Ballygunge, Kolkata, and online. Counselling is free: ask us about batches, fees or your preparation.

FREE COUNSELLING

Chat on WhatsApp

Call +91 84204 36524

Open every day, 10:30 am to 7:30 pm.

Scan to chat on WhatsApp

THE GATE CS COURSE

Classroom batches in Ballygunge and online batches. Fees and details are on the course page.

Scan: course page

See the course

FREE ON IAEACADEMY.IN

More GATE CS papers, question by question with the official answers, and free notes by subject.

Scan: more papers

More solved papers

Found this useful? Send it to a friend preparing for GATE.

Send to a friend on WhatsApp

IAE Academy free study material. The questions are those of the GATE CS 2024 Set 1 question paper, organised by IISc Bengaluru; the answers are from its official answer key. Prepared and published by IAE Academy, Kolkata. Free to read and free to share, unaltered and with attribution. Not affiliated with, endorsed by or produced by IISc Bengaluru or the GATE committee.

IAE ACADEMY INSTITUTE FOR ENGINEERS

GATE coaching in Kolkata since 2012 · EC, EE, IN, CS and IT, DA, BT and CE · in the classroom and online

227A Rash Behari Ave, Ballygunge, Kolkata 700019

+91 84204 36524 · iaeacademy.in · Updated 27 October 2026

Get directions