TIFR GS Computer Science

Sample Paper – 6

Duration: 180 Minutes **Maximum Marks: 120**

Instructions

- This paper contains **30 Multiple Choice Questions** (single correct answer), divided into **Part A: Common (Q1–Q15)** and **Part B: Computer Science (Q16–Q30)**.
- **Part A** must be attempted by all candidates. **Part B** must be attempted for Computer Science (CS) programme admission.
- Each correct answer carries **+4 marks**. **1 mark is deducted** for each wrong answer. Unattempted questions carry **0 marks**.
- Only **one** option is correct per question.
- Use of calculators, mobile phones, laptops, tablets, or any electronic device is strictly **NOT permitted**. Rough sheets are provided at the examination centre.
- Syllabus level: **Graduate-level Computer Science** — Discrete Mathematics, Algorithms & Data Structures, Theory of Computation, Programming Languages, Operating Systems, Digital Circuits, Databases.

Part A: Common (Q1–Q15)

Q1. [Boolean Algebra – Shannon Expansion]

Shannon's expansion theorem states that any Boolean function can be decomposed with respect to a variable. Which of the following correctly states Shannon's expansion of $f(A, B, C)$ with respect to variable A?

(A) $f(A, B, C) = A \cdot f(1, B, C) + A' \cdot f(0, B, C)$

(B) $f(A, B, C) = A \cdot f(0, B, C) + A' \cdot f(1, B, C)$

(C) $f(A, B, C) = A + f(1, B, C)$ only if $A = 1$

(D) $f(A, B, C) = A \cdot B \cdot C + A' \cdot B' \cdot C'$

Q2. [Combinatorics – Inclusion-Exclusion]

Given $|A| = 40$, $|B| = 30$, $|C| = 20$, $|A \cap B| = 10$, $|B \cap C| = 8$, $|A \cap C| = 5$, and $|A \cap B \cap C| = 3$. What is $|A \cup B \cup C|$?

(A) 83

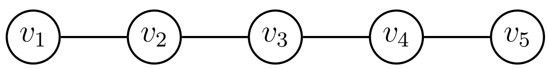
(B) 70

(C) 90

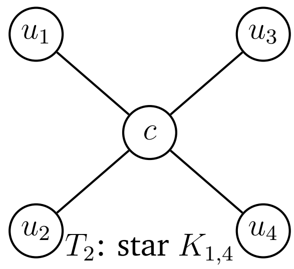
(D) 67

Q3. [Graph Theory – Tree Isomorphism]

Consider the two trees on 5 vertices shown below. T_1 is the path P_5 and T_2 is the star $K_{1,4}$ (centre c connected to 4 leaves). Are T_1 and T_2 isomorphic?



T_1: path P_5



T_2: star $K_{1,4}$

(A) Yes, both are trees on 5 vertices

(B) Yes, both have exactly 4 edges

(C) No, their degree sequences differ

(D) No, T_1 is planar but T_2 is not

Q4. [Number Theory – RSA Decryption Exponent]

In an RSA scheme with primes $p = 5$, $q = 11$: $n = 55$, $\phi(n) = 40$, public exponent $e = 3$. The private exponent d satisfies $3d \equiv 1 \pmod{40}$. What is d?

(A) 13

(B) 17

(C) 23

(D) 27

Q5. [Set Theory – Cantor's Diagonalization]

Cantor's diagonal argument shows that for any function $f : \mathbb{N} \to \{0,1\}^\omega$, one can construct a sequence $s \notin \mathrm{Image}(f)$ by setting $s_i = 1 - [f(i)]_i$ (flipping the i-th bit of $f(i)$). This proves:

(A) $|\mathbb{N}| < |\{0,1\}^\omega| = |\mathbb{R}|$

(B) $|\mathbb{N}| = |\mathbb{R}|$ (both countable)

(C) $|\{0,1\}^\omega|$ is countably infinite

(D) $|\mathbb{R}| < |\mathbb{N}|$

Q6. [Probability – Expectation of Binomial]

Let $X \sim \mathrm{Binomial}(n = 3,\ p = \frac{1}{2})$. What is $E[X]$?

(A) 1

(B) $3/2$

(C) 2

(D) 3

Q7. [Linear Algebra – LU Decomposition]

For $A = \begin{pmatrix} 2 & 1 \\ 4 & 3 \end{pmatrix}$, which is the correct LU decomposition $A = LU$ with L unit lower-triangular and U upper-triangular?

(A) $L = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 2 & 1 \\ 4 & 3 \end{pmatrix}$

(B) $L = \begin{pmatrix} 2 & 0 \\ 4 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & \frac{1}{2} \\ 0 & 1 \end{pmatrix}$

(C) $L = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}$

(D) $L = \begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 2 & 1 \\ 0 & -1 \end{pmatrix}$

Q8. [Predicate Logic – Minimal Herbrand Model]

Consider the clause set $\{P(a),\ Q(f(a)),\ \forall x\,(Q(x) \to Q(f(x)))\}$. In the minimal Herbrand model, which ground atoms are true?

(A) Only $P(a)$

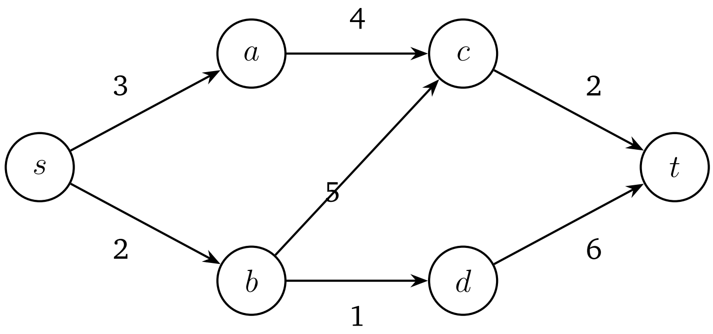
(B) Only $Q(f(a))$

(C) $P(a)$ and $P(f(a))$ only

(D) $P(a)$, $Q(f(a))$, $Q(f(f(a)))$, $\ldots$ (infinitely many Q atoms)

Q9. [Graph Theory – DAG Shortest Path]

Find the shortest-path distance from s to t in the weighted DAG below, using dynamic programming in topological order.



(A) 9

(B) 7

(C) 11

(D) 8

Q10. [Combinatorics – Bell Numbers]

The Bell number B_n counts the number of set-partitions of an n-element set. What is B_4 (the number of partitions of $\{1, 2, 3, 4\}$)?

(A) 10

(B) 15

(C) 20

(D) 12

Q11. [Probability – Central Limit Theorem]

Let $X_1, X_2, \ldots, X_n$ be IID random variables with mean μ and variance $\sigma^2 < \infty$. By the Central Limit Theorem, as $n \to \infty$, the distribution of $\dfrac{\bar{X}_n - \mu}{\sigma/\sqrt{n}}$ converges to:

(A) $\mathrm{Uniform}(0, 1)$

(B) $\text{Poisson}(1)$

(C) $N(0, 1)$ (standard normal)

(D) $\text{Exponential}(1)$

Q12. [Number Theory – Perfect Numbers]

A *perfect number* equals the sum of its proper divisors. The smallest perfect number is 6. What is the **next** perfect number?

(A) 12

(B) 16

(C) 24

(D) 28

Q13. [Relations – Partial Orders vs Total Orders]

The power set $\mathcal{P}(\{1, 2, 3\})$ ordered by set inclusion $\subseteq$ is a partially ordered set. Which property, required for a *total order*, does this poset **lack**?

(A) Totality: $\{1\}$ and $\{2\}$ are incomparable under $\subseteq$

(B) Reflexivity: $A \subseteq A$ fails for some set A

(C) Antisymmetry

(D) Transitivity

Q14. [Logic – Functional Completeness]

Which singleton set of Boolean connectives is *functionally complete* (able to express every Boolean function)?

(A) $\{\text{AND}\}$

(B) $\{\text{NAND}\}$

(C) $\{\text{OR}\}$

(D) $\{\text{XOR}\}$

Q15. [Recurrences – Linear Homogeneous]

Solve the recurrence $T(n) = 3T(n-1) - 2T(n-2)$ with $T(0) = 0$ and $T(1) = 1$. The closed-form solution is:

(A) $3^n - 2^n$

(B) $n \cdot 2^{n-1}$

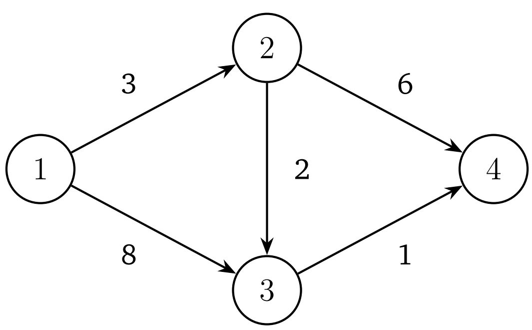
(C) $2^n - 1$

(D) $3 \cdot 2^n - 2$

Part B: Computer Science (Q16–Q30)

Q16. [Algorithms – Floyd-Warshall]

Run Floyd-Warshall on the directed weighted graph below. After processing intermediate vertex $k = 2$, what is the shortest-path estimate $d[1][4]$?



(A) 6

(B) 7

(C) 8

(D) 9

Q17. [Data Structures – Treap]

A *treap* augments a BST with random priorities, satisfying BST ordering on keys and max-heap ordering on priorities. The **expected height** of a treap on n keys is:

(A) $O(\log n)$

(B) $O(n)$

(C) $O(\log^2 n)$

(D) $O(\sqrt{n})$

Q18. [Theory of Computation – Myhill-Nerode Theorem]

Let $L = \{a^n b^n \mid n \geq 0\}$. Under the Myhill-Nerode equivalence for L, strings a^i and a^j with $i \neq j$ are:

(A) Equivalent (same extension behaviour with respect to L)

(B) Distinguishable: the suffix b^i witnesses it, since $a^i b^i \in L$ but $a^j b^i \notin L$

(C) Indistinguishable for any $i \neq j$

(D) Equivalent only if i and j have the same parity

Q19. [Algorithms – Fractional vs 0/1 Knapsack]

For the same set of items and capacity, let OPT_F be the optimal fractional knapsack value and OPT_{01} the optimal 0/1 knapsack value. Which relation **always** holds?

(A) $\mathrm{OPT}_{01} > \mathrm{OPT}_F$ always

(B) $\mathrm{OPT}_F = \mathrm{OPT}_{01}$ always

(C) $\mathrm{OPT}_F \geq \mathrm{OPT}_{01}$ always

(D) $\mathrm{OPT}_{01} > \mathrm{OPT}_F$ iff all items have equal weight

Q20. [Data Structures – Union-Find]

Using union by rank *without* path compression, the worst-case total time for any sequence of m union and find operations on n elements is:

(A) $O(m)$

(B) $O(m\sqrt{n})$

(C) $O(mn)$

(D) $O(m \log n)$

Q21. [Theory of Computation – Church-Turing Thesis]

The Church-Turing thesis asserts that:

(A) Every effectively computable function is Turing-computable (and equivalently λ-definable and μ-recursive)

(B) Turing machines can compute every function whatsoever, including non-computable ones

(C) Church's λ-calculus is strictly more powerful than Turing machines

(D) No problem is undecidable in principle

Q22. [Operating Systems – Process Scheduling]

For interactive systems, which scheduling algorithm best minimises response times for short CPU-burst (I/O-bound) processes?

(A) FCFS (First Come First Served)

(B) SRTF / Preemptive SJF (Shortest Remaining Time First)

(C) Priority scheduling without preemption

(D) Round Robin with a very large time quantum

Q23. [Compilers – Attribute Grammars]

In an attribute grammar, which statement correctly characterises synthesized and inherited attributes?

(A) Synthesized attributes are passed from parent to children

(B) Inherited attributes are computed from children to parent

(C) Synthesized attributes flow from children to parent; inherited attributes flow from parent or siblings to children

(D) All attributes must be synthesized for the grammar to be applicable

Q24. [Theory of Computation – Post Correspondence Problem]

The undecidability of the Post Correspondence Problem (PCP) is established by:

(A) A direct diagonalization argument on tile sequences

(B) Reduction from the Halting Problem where tiles abstractly represent TM steps

(C) Rice's theorem, since PCP is a semantic property of Turing machines

(D) A many-one reduction from the Halting Problem constructing domino tiles that faithfully simulate a TM computation history

Q25. [Algorithms – Randomized QuickSort Analysis]

The expected $O(n \log n)$ time of randomized QuickSort is most cleanly proved using:

(A) Indicator random variables X_{ij} for each pair (i, j) being compared; linearity of expectation gives $E[\text{total comparisons}] = O(n \log n)$

(B) Amortized analysis of merge operations within the recursion tree

(C) A potential function argument on the sequence of pivot selections

(D) The Master theorem applied to $T(n) = 2T(n/2) + n$

Q26. [Operating Systems – Lamport's Bakery Algorithm]

Lamport's bakery algorithm for mutual exclusion:

(A) Requires one atomic hardware instruction (e.g. test-and-set) per process during entry

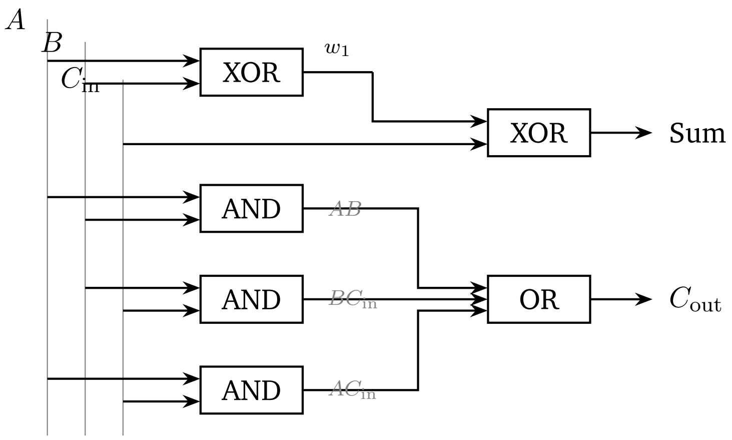
(B) Achieves mutual exclusion for N processes using only ordinary read/write registers (no special hardware)

(C) Requires a centralized coordinator process to grant access tokens

(D) Is correct only when exactly $N = 2$ processes compete

Q27. [Digital Circuits – Full Adder]

The gate-level diagram below shows a full adder with inputs A, B, C_{in} and outputs Sum and C_{out}. Which option gives the correct Boolean expressions?



(A) $\text{Sum} = A \cdot B + C_{\text{in}}, \quad C_{\text{out}} = A \oplus B$

(B) $\text{Sum} = A \oplus B, \quad C_{\text{out}} = A \cdot B \oplus C_{\text{in}}$

(C) $\text{Sum} = A \oplus B \oplus C_{\text{in}}, \quad C_{\text{out}} = A \cdot B + B \cdot C_{\text{in}} + A \cdot C_{\text{in}}$

(D) $\text{Sum} = A + B + C_{\text{in}}, \quad C_{\text{out}} = A \cdot B \cdot C_{\text{in}}$

Q28. [Databases – Strict 2PL]

Strict Two-Phase Locking (Strict 2PL) releases all locks held by a transaction:

(A) As soon as the growing phase ends (the lock point)

(B) Immediately after each individual read or write operation

(C) During the shrinking phase, just like basic 2PL

(D) Only when the transaction commits or aborts (preventing cascading rollbacks and dirty reads)

Q29. [Programming Languages – Continuation-Passing Style]

The CPS transform of the direct-style expression $(f\ 3)$ with continuation k is:

(A) $f\ 3\ k$ (pass continuation k to f as an extra argument)

(B) $(\lambda v.\ k\ v)$ (ignore f and 3 entirely)

(C) $k\ (f\ 3)$ (apply k to the direct-style result)

(D) $f\ (\lambda v.\ k\ v)$ (only valid when 3 itself is in CPS)

Q30. [Algorithms – Parallel Prefix Sums]

On a PRAM (CREW model) with n processors, computing the prefix sums (scan) of n elements takes:

(A) $O(n)$ time

(B) $O(\log n)$ time

(C) $O(\sqrt{n})$ time

(D) $O(1)$ time

Detailed Solutions

Q1.

Solution

Shannon's cofactor expansion theorem states: for any Boolean function f and variable A,

$$f(A,B,C) = A \cdot f(1,B,C) + A' \cdot f(0,B,C).$$

When $A = 1$ only the first cofactor $f(1,B,C)$ survives; when $A = 0$ only $f(0,B,C)$ survives. This is option (A).

Why others are wrong. (B) swaps the two cofactors. (C) is not a valid identity. (D) gives the AND of all literals, which is a specific function, not the general expansion theorem.

Answer: (A) ↑ Go Back to Q1

Q2.

Solution

By the inclusion-exclusion principle:

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C| + |A \cap B \cap C|$$

$$= 40 + 30 + 20 - 10 - 8 - 5 + 3 = 90 - 23 + 3 = \mathbf{70}.$$

Why others are wrong. (A) 83 omits the triple-intersection term $(90 - 7 = 83)$. (C) 90 is the raw sum before subtracting intersections. (D) 67 makes an arithmetic error in the subtraction.

Answer: (B) ↑ Go Back to Q2

Q3.

Solution

Two graphs are isomorphic only if they share the same degree sequence.

- $T_1 = P_5$: degree sequence $(1,1,2,2,2)$ – two leaves, three internal vertices of degree 2.
- $T_2 = K_{1,4}$: degree sequence $(1,1,1,1,4)$ – four leaves, one centre of degree 4.

The degree sequences differ, so T_1 and T_2 are **not isomorphic** – option (C).

Why others are wrong. (A) and (B) give necessary but not sufficient conditions for isomorphism. (D) is false: all trees are planar, including $K_{1,4}$.

Answer: (C) ↑ Go Back to Q3

Q4.

Solution

We need d with $3d \equiv 1 \pmod{40}$. Using the extended Euclidean algorithm:

$$40 = 13 \cdot 3 + 1 \implies 1 = 40 - 13 \cdot 3 \implies -13 \equiv 27 \pmod{40}.$$

Verify: $3 \times 27 = 81 = 2 \times 40 + 1 \equiv 1 \pmod{40}$. So $d = \mathbf{27}$.

Why others are wrong. $3 \times 13 = 39 \equiv -1$; $3 \times 17 = 51 \equiv 11$; $3 \times 23 = 69 \equiv 29 \pmod{40}$. None equal 1.

Answer: (D) ↑ Go Back to Q4

Q5.

Solution

Given any $f : \mathbb{N} \to \{0,1\}^\omega$, define s by $s_i = 1 - [f(i)]_i$ (flip the i-th bit of $f(i)$). Then $s \neq f(n)$ for every n (they differ at position n), so f is not surjective. Since this holds for *every* f, no surjection $\mathbb{N} \twoheadrightarrow \{0,1\}^\omega$ exists, giving $|\mathbb{N}| < |\{0,1\}^\omega|$. Via binary representations, $|\{0,1\}^\omega| = |\mathbb{R}|$.

Why others are wrong. (B) $|\mathbb{R}| > |\mathbb{N}|$, so they are not equal. (C) The uncountability of $\{0,1\}^\omega$ is precisely what the argument proves. (D) contradicts the conclusion.

Answer: (A) ↑ Go Back to Q5

Q6.

Solution

For $X \sim \text{Binomial}(n,p)$, the expectation is $E[X] = np$. With $n = 3$ and $p = \frac{1}{2}$:

$$E[X] = 3 \times \tfrac{1}{2} = \mathbf{\tfrac{3}{2}}.$$

Why others are wrong. (A) $np = 1$ requires, e.g., $n = 2, p = \frac{1}{2}$. (C) $np = 2$ requires $n = 4, p = \frac{1}{2}$. (D) $np = 3$ would require $p = 1$.

Answer: (B) ↑ Go Back to Q6

Q7.

Solution

Gaussian elimination on A: subtract $\ell_{21} = \frac{4}{2} = 2$ times row 1 from row 2:

$$U = \begin{pmatrix} 2 & 1 \\ 4 - 2 \cdot 2 & 3 - 2 \cdot 1 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}, \quad L = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}.$$

Verification: $LU = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 4 & 3 \end{pmatrix} = A.$ ✓

Why others are wrong. (A) uses $L = I$, so $U = A$ which is not upper triangular. (B) L is not unit lower-triangular (diagonal entries must be 1). (D) wrong multiplier: $\ell_{21} = 4$ would leave $U_{22} = 3 - 4 = -1$ only if row 2 subtracts $4\times$ row 1; but then $L_{21} = 4$, not 4: check LU gives $\begin{pmatrix} 2 & 1 \\ 8-2 & 4-2-1 \end{pmatrix}$ – incorrect.

Answer: (C) ↑ Go Back to Q7

Q8.

Solution

The Herbrand universe is $\mathcal{H} = \{a, f(a), f(f(a)), \ldots\}$. Build the minimal Herbrand model clause by clause:

- $P(a)$: ground fact – $P(a) = \mathbf{T}$.
- $Q(f(a))$: ground fact – $Q(f(a)) = \mathbf{T}$.
- $\forall x\,(Q(x) \to Q(f(x)))$: applying with $x = f(a)$ gives $Q(f(f(a))) = \mathbf{T}$; applying with $x = f(f(a))$ gives $Q(f(f(f(a)))) = \mathbf{T}$; and so on.

No rule derives new P atoms, so only $P(a)$ is true for P. The minimal model contains $P(a)$ together with *all* $Q(f^n(a))$ for $n \geq 1$ – infinitely many atoms – matching option (D).

Why others are wrong. (A) ignores the Q facts. (B) ignores $P(a)$ and the chain of Q atoms. (C) there is no rule producing $P(f(a))$.

Answer: (D) ↑ Go Back to Q8

Q9.

Solution

Process vertices in topological order s, a, b, c, d, t and relax edges:

$$d[s] = 0, \quad d[a] = 0 + 3 = 3, \quad d[b] = 0 + 2 = 2,$$

$$d[c] = \min(3 + 4,\ 2 + 5) = \min(7, 7) = 7, \quad d[d] = 2 + 1 = 3,$$

$$d[t] = \min(7 + 2,\ 3 + 6) = \min(9, 9) = \mathbf{9}.$$

The shortest s-t distance is 9, achieved by both paths $s \to a \to c \to t$ and $s \to b \to d \to t$.

Why others are wrong. (B) 7 is $d[c]$, not $d[t]$. (C) 11 would come from $s \to a \to c \to t$ with wrong weights. (D) 8 does not correspond to any path in the graph.

Answer: (A) ↑ Go Back to Q9

Q10.

Solution

Bell numbers count set partitions: $B_0 = 1, B_1 = 1, B_2 = 2, B_3 = 5, B_4 = 15$. The 15 partitions of $\{1, 2, 3, 4\}$ are: 1 partition into one block; 7 partitions into two blocks; 6 partitions into three blocks; 1 partition into four singletons. Total: $1 + 7 + 6 + 1 = \mathbf{15}$.

Why others are wrong. (A) $10 = \binom{5}{2}$, a binomial coefficient, not a Bell number. (C) $20 = \binom{6}{3}$. (D) 12 has no combinatorial interpretation here.

Answer: (B) ↑ Go Back to Q10

Q11.

Solution

The Central Limit Theorem (CLT) states: if $X_1, \ldots, X_n$ are IID with mean μ and finite variance σ^2, then as $n \to \infty$,

$$\frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}} \xrightarrow{d} N(0, 1).$$

The limit is the **standard normal distribution** $N(0, 1)$, regardless of the original distribution of the X_i.

Why others are wrong. (A) Uniform$(0, 1)$ and (B) Poisson(1) are not the CLT limit. (D) Exponential(1) arises in other contexts (e.g. inter-arrival times), not the CLT.

Answer: (C) ↑ Go Back to Q11

Q12.

Solution

The proper divisors of 28 are $\{1, 2, 4, 7, 14\}$. Their sum: $1 + 2 + 4 + 7 + 14 = \mathbf{28}$. So 28 is perfect.

To confirm 28 is the *next* after 6: check 7–27. Using the Euclid–Euler theorem, even perfect numbers have the form $2^{p-1}(2^p - 1)$ with $2^p - 1$ prime. For $p = 2$: $2(3) = 6$. For $p = 3$: $4(7) = 28$. No perfect number lies strictly between 6 and 28.

Why others are wrong. 12: divisors $1+2+3+4+6 = 16 \neq 12$. 16: $1+2+4+8 = 15 \neq 16$. 24: $1 + 2 + 3 + 4 + 6 + 8 + 12 = 36 \neq 24$.

Answer: (D) ↑ Go Back to Q12

Q13.

Solution

A total order requires all four properties: reflexivity, antisymmetry, transitivity, and **totality** (every pair is comparable). The poset $(\mathcal{P}(\{1,2,3\}), \subseteq)$ satisfies reflexivity ($A \subseteq A$), antisymmetry, and transitivity, but *not* totality: the sets $\{1\}$ and $\{2\}$ are incomparable (neither $\{1\} \subseteq \{2\}$ nor $\{2\} \subseteq \{1\}$). This is option (A).

Why others are wrong. (B) Reflexivity holds: $A \subseteq A$ for all A. (C) and (D): antisymmetry and transitivity both hold for $\subseteq$.

Answer: (A) ↑ Go Back to Q13

Q14.

Solution

A set S of connectives is functionally complete if every Boolean function can be expressed using only connectives in S. NAND alone is functionally complete: $\neg A = A \uparrow A$ (NAND with itself); $A \wedge B = (A \uparrow B) \uparrow (A \uparrow B)$; and OR follows by De Morgan's law. Thus $\{\text{NAND}\}$ is complete – option (B).

Why others are wrong. $\{\text{AND}\}$ cannot express negation (it is monotone). $\{\text{OR}\}$ is also monotone and cannot express negation. $\{\text{XOR}\}$ cannot express the constant function 1 or AND, so it is not complete.

Answer: (B) ↑ Go Back to Q14

Q15.

Solution

Characteristic equation: $r^2 = 3r - 2 \implies r^2 - 3r + 2 = 0 \implies (r-1)(r-2) = 0$, so roots $r = 1, 2$.

General solution: $T(n) = \alpha \cdot 1^n + \beta \cdot 2^n = \alpha + \beta \cdot 2^n$.

Applying initial conditions:

$$T(0) = 0 : \ \alpha + \beta = 0 \implies \alpha = -\beta.$$

$$T(1) = 1 : \ \alpha + 2\beta = 1 \implies -\beta + 2\beta = \beta = 1, \ \alpha = -1.$$

Thus $T(n) = 2^n - 1$.

Verification. $T(0) = 1 - 1 = 0\,\checkmark$; $T(1) = 2 - 1 = 1\,\checkmark$; $T(2) = 3$: check $3(1) - 2(0) = 3\,\checkmark$.

Why others are wrong. (A) $3^n - 2^n$: $T(2) = 9 - 4 = 5 \neq 3$. (B) $n \cdot 2^{n-1}$: $T(2) = 4 \neq 3$. (D) $3 \cdot 2^n - 2$: $T(0) = 3 - 2 = 1 \neq 0$.

Answer: (C) ↑ Go Back to Q15

Q16.

Solution

Initial distance matrix $D^{(0)}$ (edges as given; ∞ for missing edges):

$$D^{(0)} = \begin{pmatrix} 0 & 3 & 8 & \infty \\ \infty & 0 & 2 & 6 \\ \infty & \infty & 0 & 1 \\ \infty & \infty & \infty & 0 \end{pmatrix}.$$

After $k = 1$ (using vertex 1 as intermediate): since $d[2][1] = d[3][1] = d[4][1] = \infty$, no entry improves. $D^{(1)} = D^{(0)}$.

After $k = 2$ (using vertex 2 as intermediate): $d^{(2)}[i][j] = \min(d^{(1)}[i][j],\ d^{(1)}[i][2] + d^{(1)}[2][j])$.

- $d[1][3] = \min(8,\ 3+2) = 5$.
- $d[1][4] = \min(\infty,\ 3+6) = \mathbf{9}$.

Rows 3 and 4 have $d[3][2] = d[4][2] = \infty$, so no other updates occur. $d[1][4] = 9$ – option (D).

Why others are wrong. 6 would require a direct path of length 6 not going via vertex 2; none exists. 7 and 8 do not correspond to any path length.

Answer: (D) ↑ Go Back to Q16

Q17.

Solution

In a treap, each key is assigned an independent uniform random priority. The structure is equivalent to a random BST built by inserting keys in the order of their priorities. The expected height of a random BST on n keys is $\Theta(\log n)$ (the same as the expected depth of a random permutation's recursion tree). More precisely, the expected height is at most $4.311 \ln n$ with high probability. Thus the expected height is $O(\log n)$ – option (A).

Why others are wrong. (B) $O(n)$ is the worst-case height of an unbalanced BST, not the treap expectation. (C) $O(\log^2 n)$ and (D) $O(\sqrt{n})$ are not tight bounds for treaps.

Answer: (A) ↑ Go Back to Q17

Q18.

Solution

Two strings x and y are Myhill-Nerode equivalent for L iff for every z, $xz \in L \Leftrightarrow$

$yz \in L$. Take $x = a^i$ and $y = a^j$ with $i \neq j$, and let $z = b^i$:

$$a^i b^i \in L \quad \text{but} \quad a^j b^i \notin L \text{ (since } j \neq i\text{)}.$$

So $z = b^i$ *distinguishes* a^i from a^j; they are not equivalent – option (B). This also shows L has infinitely many equivalence classes, confirming (by the Myhill-Nerode theorem) that L is not regular.

Why others are wrong. (A), (C), (D) all claim some form of equivalence, which the witness $z = b^i$ refutes.

Answer: (B) ↑ Go Back to Q18

Q19.

Solution

The fractional knapsack relaxes the 0/1 integrality constraint: items may be taken in fractions. Any feasible 0/1 solution is also feasible for the fractional problem, so:

$$\mathrm{OPT}_F \geq \mathrm{OPT}_{01}.$$

Equality holds iff the greedy fractional solution happens to be integral. In general $\mathrm{OPT}_F > \mathrm{OPT}_{01}$ is possible – option (C).

Why others are wrong. (A) $\mathrm{OPT}_{01} > \mathrm{OPT}_F$ is impossible (0/1 is more constrained). (B) equality need not hold. (D) the condition for equality is not simply equal weights.

Answer: (C) ↑ Go Back to Q19

Q20.

Solution

Union by rank keeps trees balanced by always attaching the shorter tree under the taller one. Without path compression, a find on an element at depth d costs $O(d)$. The rank of any tree is at most $\lfloor \log_2 n \rfloor$, so the depth (and each find) costs at most $O(\log n)$. Over m operations the total cost is $O(m \log n)$ – option (D).

Why others are wrong. $O(m)$ would be optimal (amortized $O(1)$ per operation), which requires path compression as well. $O(m\sqrt{n})$ is the bound for neither strategy. $O(mn)$ corresponds to naive union without any rank heuristic.

Answer: (D) ↑ Go Back to Q20

Q21.

Solution

The Church-Turing thesis is a *thesis* (not a theorem): it claims that every **intuitively effectively computable** function coincides with the class of Turing-computable (equivalently λ-definable, equivalently μ-recursive) functions. It captures the informal notion of an algorithm. This is option (A).

Why others are wrong. (B) Turing machines cannot compute non-computable functions (e.g. the halting function); this contradicts the thesis itself. (C) λ-calculus and Turing machines are *equivalent* in power. (D) undecidable problems exist and are well established (e.g. the Halting Problem).

Answer: (A) ↑ Go Back to Q21

Q22.

Solution

SRTF (Shortest Remaining Time First), the preemptive version of SJF, always runs the process with the least remaining CPU time. For I/O-bound processes that perform short CPU bursts, SRTF immediately preempts longer jobs and schedules the short burst, resulting in the smallest possible average response time – option (B).

Why others are wrong. (A) FCFS can create long waits (convoy effect). (C) Non-preemptive priority scheduling cannot interrupt a running long job. (D) Round Robin with a very large quantum degenerates to FCFS.

Answer: (B) ↑ Go Back to Q22

Q23.

Solution

In an attribute grammar attached to a context-free grammar:

- A **synthesized attribute** of a non-terminal A in a production $A \to \alpha$ is computed from the attributes of the symbols in α (children in the parse tree). Information flows *up* from children to parent.
- An **inherited attribute** of a non-terminal B in $A \to \ldots B \ldots$ is computed from the attributes of A (the parent) or siblings. Information flows *down* or *sideways*.

This is option (C).

Why others are wrong. (A) and (B) swap the definitions. (D) S-attributed grammars use only synthesized attributes, but inherited attributes are perfectly valid and commonly used.

Answer: (C) ↑ Go Back to Q23

Q24.

Solution

PCP undecidability is proved by a **many-one reduction from the Halting Problem**: given a TM M and input w, one constructs a set of domino tiles whose top and bottom strings can be matched (forming a solution) iff M accepts w. The tiles encode TM configurations – transitions, initial, accepting states – so a PCP solution corresponds precisely to an accepting computation history. This is option (D).

Why others are wrong. (A) PCP involves combinatorial matching, not diagonalization. (B) overly vague – the reduction is concrete and many-one, not just a simulation argument. (C) Rice's theorem applies to semantic properties of TM *languages*, not to combinatorial matching problems like PCP.

Answer: (D) ↑ Go Back to Q24

Q25.

Solution

For elements with sorted ranks $i < j$ (out of n), define $X_{ij} = 1$ if elements i and j are compared. Two elements are compared iff one of them is chosen as pivot while both still lie in the same subarray, which happens iff it is the first element in $\{i, i+1, \ldots, j\}$ chosen as pivot:

$$E[X_{ij}] = \frac{2}{j-i+1}.$$

By linearity of expectation:

$$E\left[\sum_{i<j} X_{ij}\right] = 2\sum_{1\le i<j\le n} \frac{1}{j-i+1} = 2\sum_{k=1}^{n-1} \frac{n-k}{k+1} = O(n\log n).$$

This is option (A).

Why others are wrong. (B) amortised analysis is for data structures. (C) potential functions do not directly apply here. (D) $T(n) = 2T(n/2) + n$ only holds when the pivot *always* splits evenly – not guaranteed even in expectation for random pivots.

Answer: (A) ↑ Go Back to Q25

Q26.

Solution

Lamport's bakery algorithm assigns each process a "ticket number" and uses a lexicographic tie-breaking rule. Crucially, it relies solely on *ordinary read and write* operations on shared variables – no atomic instructions such as test-and-set, compare-and-swap, or fetch-and-add are needed. It achieves mutual exclusion for any $N \geq 2$ processes – option (B).

Why others are wrong. (A) hardware atomics are precisely what bakery avoids. (C) there is no coordinator; processes coordinate via their ticket arrays directly. (D) the algorithm works for any N, not just 2.

Answer: (B) ↑ Go Back to Q26

Q27.

Solution

A full adder computes the one-bit sum of three bits A, B, C_{in}. The standard Boolean expressions are derived as follows.

Sum is the parity (XOR) of all three inputs:

$$\text{Sum} = A \oplus B \oplus C_{\text{in}}.$$

Carry out is 1 whenever at least two of the three inputs are 1:

$$C_{\text{out}} = AB + BC_{\text{in}} + AC_{\text{in}}.$$

The circuit in the diagram realises this with two XOR gates (intermediate $w_1 = A \oplus B$ then $\text{Sum} = w_1 \oplus C_{\text{in}}$) and three AND gates whose outputs feed an OR gate for C_{out}. This is option (C).

Truth-table check for $A = B = C_{\text{in}} = 1$: Sum$= 1 \oplus 1 \oplus 1 = 1$, $C_{\text{out}} = 1 + 1 + 1 = 1$ $(2 + 1 = 3$ in binary: $11_2)$. ✓

Why others are wrong. (A) and (D) are not valid adder expressions. (B) $C_{\text{out}} = AB \oplus C_{\text{in}}$ is incorrect: for $A = B = C_{\text{in}} = 1$ it gives $1 \oplus 1 = 0 \neq 1$.

Answer: (C) ↑ Go Back to Q27

Q28.

Solution

Basic 2PL divides a transaction's lifetime into a growing phase (locks acquired) and a shrinking phase (locks released). Locks may be released during the shrinking phase, which can allow cascading rollbacks.

Strict 2PL tightens this: *all* locks (both shared and exclusive) are held until the transaction **commits or aborts**. This prevents dirty reads (reading uncommitted data) and cascading rollbacks, because no other transaction can access the written data until it is committed. This is option (D).

Why others are wrong. (A) describes the basic 2PL lock point, not Strict 2PL.

(B) releasing locks immediately after each operation violates even basic 2PL. (C) Strict 2PL does not release locks during the shrinking phase at all.

Answer: (D) ↑ Go Back to Q28

Q29.

Solution

In continuation-passing style (CPS), every function f is transformed to accept an extra *continuation* argument k – a function that represents "what to do next" with the result. The CPS transform of the call $(f\ 3)$ with continuation k is:

$$f\ 3\ k$$

which means: invoke the CPS version of f with argument 3 and continuation k; f will eventually call k with its result instead of returning normally. This is option (A).

Why others are wrong. (B) ignores both f and 3. (C) $k\,(f\,3)$ applies k to the *direct-style* result of f – this is meta-style and assumes f has not been CPS-transformed. (D) $f\,(\lambda v.\,k\,v)$ passes 3 implicitly but is missing 3 as an argument, and the form is wrong.

Answer: (A) ↑ Go Back to Q29

Q30.

Solution

The parallel prefix (scan) algorithm proceeds in two sweeps on a balanced binary tree of depth $\lceil \log_2 n \rceil$:

(a) **Up-sweep (reduce):** In $O(\log n)$ parallel steps, compute partial sums at each internal node.

(b) **Down-sweep (distribute):** In $O(\log n)$ parallel steps, propagate prefix values down to leaves.

Total parallel time: $O(\log n)$ with n processors on a CREW PRAM – option (B).

Why others are wrong. (A) $O(n)$ is the sequential time with one processor. (C) $O(\sqrt{n})$ arises in some restricted models, not CREW PRAM. (D) $O(1)$ is impossible: at least $\Omega(\log n)$ depth is needed due to the data-dependence chain.

Answer: (B) ↑ Go Back to Q30

Answer Key

Part A: Common (Q1–Q15)

Q	Ans	Q	Ans	Q	Ans	Q	Ans	Q	Ans
1	A	2	B	3	C	4	D	5	A
6	B	7	C	8	D	9	A	10	B
11	C	12	D	13	A	14	B	15	C

Part B: Computer Science (Q16–Q30)

Q	Ans	Q	Ans
16	D	17	A
18	B	19	C
20	D	21	A
22	B	23	C
24	D	25	A
26	B	27	C
28	D	29	A
30	B		

Scan the QR code on each page footer to access more TIFR GS sample papers and solutions on Collegedunia.