DEEP PYQ ANALYSIS

GATE CSE 2027

High-Probability Question Bank
& 10 Predicted Mock Tests

MD ANISH AHAMAD

FREE PREVIEW

GATE CSE 2027 — Deep PYQ Analysis & High-Probability Question Bank

By MD ANISH AHAMAD · First Edition 2026

This preview contains 32 real, unedited pages from the 1,016-page book, exactly as they appear in the full PDF. Page numbers in the running heads and on the contents pages refer to the full book.

In this preview	In the full book	Pages
Title page	front matter	1
Contents	front matter	4
Preface	front matter	1
Executive summary	pages 1–3	3
What changed in the 2027 syllabus	pages 11–12	2
Subject-wise weightage 2009–2026	pages 16–17	2
Sample chapter: Operating System	page 307	1
Sample questions with four confidence scores	pages 36–39	4
Top 100 concepts: ranks 1–25	pages 494–495	2
How to take and score a mock test	page 532	1
Mock Test 01 (first pages)	pages 533–535	3
Mock Test 01 answer key	page 548	1
Mock Test 01 worked solutions (first pages)	pages 718–719	2
Last-minute revision sheet (first pages)	pages 934–935	2
Diagram-based questions (first pages)	pages 962–964	3

The full book also has: all eleven subject chapters, 470 practice questions with worked solutions and four confidence scores each, all ten mock tests with keys and solutions, the complete Top 100, the do-not-skip checklist, the 20-week roadmap and the full last-minute sheet.

GATE CSE 2027

Deep PYQ Analysis & High-Probability Question Bank

Evidence-based topic prediction, a 470-question bank and ten full-length predicted mock tests for the GATE Computer Science & Information Technology paper

MD ANISH AHAMAD

First Edition · 2026
For the examination of February 2027
gatecseguide.github.io

Contents

Preface

Most GATE preparation books answer the question "what is in the syllabus". This one tries to answer a harder and more useful question: **given twenty-seven years of question papers, what is actually likely to be asked in February 2027, and how confident can anyone honestly be about that?**

The difference matters. A candidate has perhaps twenty weeks and eleven subjects. Spending those weeks in proportion to the syllabus is not the same as spending them in proportion to the evidence. Cache mapping, CPU scheduling, TCP congestion control and eigenvalue properties have been examined in almost every recent paper. Several syllabus topics have not been examined in a decade. Both facts are knowable, and preparation should reflect them.

How this book was built

Four steps, in order:

1. **Verify the ground truth.** The official GATE 2027 syllabus, examination pattern and dates were taken directly from the organizing institute's portal, not from secondary sources. The syllabus PDF was extracted and compared word by word against the 2026 wording. The 2027 revision is real, and it matters: Computer Networks lost a third of its topic list, Computer Organization gained control-unit design and cache mapping as explicit items, and Digital Logic named the tabular minimisation method for the first time.
2. **Analyse the archive.** Thirty-seven GATE CS papers and sets from 2000 to 2026 were analysed at subject level for marks per paper, and at topic level for which concept was tested in which year and in which format.
3. **Score every topic.** An eight-factor model, set out in full in Section 7, converts that evidence into a single score per topic: long-term frequency, recent frequency, concept breadth, syllabus centrality, time since last appearance, format suitability, organizing-institute signal and other evidence.
4. **Write to the evidence.** The scores decided how many questions each subject received, which concepts got questions, and which variants of each concept appeared. A subject with more live concepts got more questions. Nothing was padded to make the chapters look even.

What this book refuses to do

It does not claim to know the paper. GATE questions are written by committees, and the year-to-year variation is real. Three commitments follow from that:

- **Facts, analysis and predictions are labelled separately** throughout. A statement about what happened in 2019 is marked FACT. A statement about what the data suggests is marked ANALYSIS. A statement about 2027 is marked PREDICTION. They are never blended.
- **Confidence is reported in four dimensions, not one.** Being confident that cache mapping will be examined is not the same as being confident about a particular cache question. Every question therefore carries Topic, Concept, Question-pattern and Exact-question confidence, and the last of these is capped at 35 % because GATE repeats concepts, not wording.
- **Uncertainty is stated, not hidden.** Where the evidence is thin, the book says so. Where two sources disagree, the disagreement is shown. Where the organizing institute tells us little, that is written down rather than dressed up.

Section 1 — Executive Summary

1.1 What this document is

A research-driven preparation resource for the **GATE 2027 Computer Science and Information Technology (CS)** paper. It was built in four steps: (1) the official 2027 syllabus, pattern and organizing institute were verified from gate2027.iitm.ac.in; (2) 37 GATE CS papers and sets from 2000 to 2026 were analysed at subject level (marks per paper) and at topic level (which concept was tested in which year, in which format); (3) a transparent eight-factor model scored every syllabus topic; (4) the scores drove a 440-question subject-wise question bank (plus 30 diagram-based questions in Appendix A, 470 in all) and ten full-length mock tests in the official 2027 pattern, each with an answer key and detailed solutions.

Throughout, three kinds of statement are kept apart: **FACT** (what happened in past papers or what the official documents say), **ANALYSIS** (what the data suggests) and **PREDICTION** (what the model expects for 2027). Every prediction-confidence figure is an analytical ranking, not a probability; nothing in this document is guaranteed to appear.

Every predicted question carries **four separate confidence scores** rather than one, because being sure a topic will be examined is not the same as being sure a particular question will be: **Topic** (will this topic appear at all), **Concept** (will this underlying concept be tested), **Question-pattern** (will a question of this shape appear) and **Exact-question** (will something very close to this question appear). The four always decrease from left to right, and Exact-question confidence is capped at 35 %, because what GATE repeats is the concept and the computation, not the wording. The rubric is in Section 7.6.

Every question also carries a **PYQ-lineage tag** naming the past papers its concept comes from, with the most recent one called out, for example "GATE 2005, 2008, 2012, 2019, 2021, 2024 · last tested 2024". Where a concept has no usable precedent the tag says so instead of listing years: General Aptitude items are written fresh each year, and the topics the 2027 revision newly made explicit have no PYQ history at all. Every question in this book is newly written; none is a reprint of a past-year question. Section 8.2 explains how to read the tag.

1.2 Verified facts that frame everything else

Item	Finding	Source
Organizer	IIT Madras (previous CS papers under IIT Madras: 2003, 2011, 2019)	gate2027.iitm.ac.in
Exam dates	6–7, 13–14, 20–21 February 2027; results 19 March 2027	gate2027.iitm.ac.in
Pattern	65 questions, 100 marks, 3 hours; GA 15 + Engineering Mathematics 13 + Subject 72; MCQ/MSQ/NAT; negative marking only on MCQ (−1/3, −2/3)	gate2027.iitm.ac.in/question_paper_pattern
Syllabus	Revised for 2027. Digital Logic reworded (minimization methods named, tabular method explicit); COA restructured (hardwired vs microprogrammed control, memory interfacing, memory performance and cache mapping explicit; main memory/secondary storage dropped); Computer Networks reduced (OSI/TCP-IP stack enumeration, framing, bridging, shortest-path/flooding routing, ARP, DHCP, ICMP, UDP, SMTP, FTP, Email removed; performance metrics and socket API added). Seven other sections and GA unchanged.	Official CS syllabus PDF, cross-checked with four independent change analyses
Recent paper style	2021–2026: about 34 MCQ, 11–15 MSQ, 16–19 NAT per paper; multi-step numericals; "familiar concept, unfamiliar setting"; 2026 rated moderate–difficult and lengthy	Analyst reports for 2021–2026

1.3 Key findings from 20+ years of papers

1. **Subject weights are stable at the level that matters.** Over 2009–2026 the average marks per paper are: Programming & DS 9.9, Operating System 8.8, Algorithms 8.5, COA 8.5, Computer Networks 8.2, Theory of Computation 7.8, Databases 6.9, Digital Logic 4.8, Compiler Design 4.6, with Mathematics settled at the official 13 (Discrete about 5–10, Linear Algebra/Calculus/Probability about 4–8) and GA fixed at 15. The only structural shifts since 2018 are COA up (7.7 → 9.2), Compiler Design up (3.8 → 5.4) and Discrete Mathematics up (6.0 → 7.8).
2. **The same 30–40 concepts produce most of the marks every year.** CPU scheduling, page replacement, paging/TLB, semaphores and deadlock; cache mapping, pipelining and AMAT; C output, recursion, BST and heaps; complexity/recurrences, MST, shortest paths, DP and hashing; DFA design, language classification, closure and decidability; LL/LR parsing, FIRST/FOLLOW and SDT; SQL output, normalization, serializability and B+ trees; TCP congestion control, CIDR, sliding windows and delay arithmetic; K-maps, counters, 2's complement and IEEE-754; eigenvalues, Bayes, expectation, graph counting and logic. These recur in new numerical clothing rather than as repeated questions.
3. **Question form has changed more than content.** Since 2014 (NAT) and 2021 (MSQ), roughly half of all marks carry no negative marking, and setters prefer questions with a unique numeric answer or a set of true/false statements. The concepts of 2005 are the concepts of 2026; the wording is now NAT/MSQ and questions more often combine two concepts.
4. **The 2027 syllabus revision concentrates Computer Networks.** Removed protocols accounted for a meaningful share of historical CN marks; the retained topics (TCP, IPv4/CIDR/NAT, routing algorithms, DLL/MAC, performance metrics, DNS/HTTP) should absorb the full 6–11 mark CN weight, making each of them denser.
5. **The organizer is a weak signal.** IIT Madras papers (2003, 2011, 2019) were PDS- and Discrete-Maths-heavy, lengthy and NAT-oriented, with above-average Digital Logic and (in 2019) tricky DBMS. The model gives this only 5 % weight; the organizer's decisive influence is the syllabus revision itself.

1.4 Most important subjects for 2027 (ANALYSIS)

Band	Subjects	Expected marks each	Why
High	Programming & DS; Algorithms; Engineering Mathematics (Discrete + LA/Calc/Prob); General Aptitude	9–12; 9–12; 13; 15	Largest and most stable blocks; PDS+Algorithms average 18 marks together
Medium	Operating System; COA; Computer Networks; Theory of Computation; Databases	6–10	Stable 7–9 mark averages; COA rising; CN concentrated by the syllabus cut
Low weight, high return	Compiler Design; Digital Logic	4–7	Formula-stable, predictable forms, best marks per hour of preparation

1.5 Highest-confidence topic predictions (PREDICTION, Tier 1 = model score ≥ 80)

- **Engineering Mathematics:** eigenvalues/eigenvectors; conditional probability and Bayes; graph degree/edge counting and Euler's formula; expectation and variance; named distributions; propositional logic.
- **Digital Logic:** Boolean algebra simplification; K-map minimization and prime-implicant counting; 2's complement range/overflow; IEEE-754 and custom floating-point formats.
- **COA:** cache mapping arithmetic; pipeline cycle counting with hazards and forwarding; AMAT for multi-level caches; pipeline speedup/throughput with latch delays.
- **Programming & DS:** C pointer/array output; recursion tracing; BST insertion/validity/successor/deletion; binary heap array operations; C operators and control flow; tree traversal reconstruction.

- **Algorithms:** loop/code complexity; Dijkstra execution (including modified rules); MST properties and counting; recurrences; shortest-path properties and Bellman–Ford/DAG; novel DP recurrences; DFS edge classification and timestamps; Kruskal/Prim execution.
- **Theory of Computation:** language-class identification; regular expressions; minimum-state DFA; decidability lists; CFG language/ambiguity/parse-tree counts; closure properties.
- **Compiler Design:** syntax-directed translation evaluation; LR conflict and grammar-class identification; LL(1) tables and conditions.
- **Operating System:** scheduling timelines (SRTF/RR/priority); synchronization-code property checks; paging and multi-level page-table arithmetic; page-replacement traces; inode arithmetic.
- **Databases:** SQL output on a printed instance; conflict serializability; normal forms and decomposition; B+ tree capacity/height.
- **Computer Networks:** TCP congestion-control trajectories; CIDR/subnetting; sliding-window arithmetic; delay and bandwidth-delay arithmetic.

Full ranked lists, scores and evidence are in Sections 5, 6 and 9; the "Do Not Skip" list in Section 10 covers what must be prepared regardless of ranking.

1.6 What the document contains

Section	Content	Size
2	Official 2027 syllabus hierarchy with 2026→2027 change table and syllabus history	—
3	Organizing-institute analysis (IIT Madras 2003/2011/2019) and its weight in the model	—
4	20+ years of paper analysis: master weightage table 2009–2026, aggregates, year narrative, type and difficulty evolution	—
5 / 8	Eleven subject chapters: syllabus tree, historical weightage, topic-wise PYQ concept map, frequency/recurrence/recency analysis, format and difficulty mapping, IIT Madras signal, prediction scoring, ranked summary, question patterns, and the question bank	440 questions: EM 54, DL 26, COA 40, PDS 46, Algorithms 50, TOC 40, CD 28, OS 46, DBMS 42, CN 40, GA 28
6	Cross-subject topic-wise deep analysis (Subject → Topic → Subtopic → Concept master table)	—
7	The prediction model: factors, weights, bands, limitations	—
9	Top 100 high-priority concepts, ranked with evidence	100+ entries
10	Do Not Skip list and checklist	—
11	Ten full-length mock tests in the official 2027 pattern (65 questions, 100 marks each) with distinct design themes	650 questions
12	Answer keys for all ten tests	—
13	Detailed solutions for all ten tests	—
14	Priority roadmap (A–D), subject-wise revision strategy, 20-week plan, attempt strategy	—
15	Last-minute sheet: top 50 concepts, formulas, algorithms, traps, confusions, edge cases	—
16	Sources	—

2.3 What changed: 2026 → 2027 (FACT, verified against the official PDF)

Section	2026 wording (summary)	2027 wording (official)	Effect
Digital Logic	Boolean algebra. Combinational and sequential circuits. Minimization. Number representations and computer arithmetic (fixed and floating point).	Boolean algebra and minimization – algebraic technique, Karnaugh map, tabular method. Design of combinational and sequential circuits. Number representation and arithmetic (fixed and floating point).	Minimization methods named explicitly (tabular/Quine–McCluskey now explicit). "Design of" emphasised. No topic removed.
COA	Machine instructions and addressing modes. ALU, data-path and control unit. Instruction pipelining, pipeline hazards. Memory hierarchy: cache, main memory and secondary storage; I/O interface (interrupt and DMA mode).	Instruction set and addressing modes. Design of ALU. Design of control unit – hardwired and microprogrammed. Memory interfacing and hierarchy: performance, cache memory mapping. I/O interface (interrupt and DMA). Instruction pipelining, pipeline hazards.	Newly explicit: hardwired vs microprogrammed control unit, memory interfacing, memory performance, cache mapping. Dropped from explicit list: main memory and secondary storage (disk).
Computer Networks	Concept of layering: OSI and TCP/IP stacks; packet/circuit/virtual-circuit switching; DLL: framing, error detection, MAC, Ethernet bridging; routing: shortest path, flooding, DV, LS; fragmentation, IPv4, CIDR, ARP/DHCP/ICMP, NAT; transport: flow/congestion control, UDP, TCP, sockets; application: DNS, SMTP, HTTP, FTP, Email.	Principles of Layering; switching (circuit, packet, virtual circuit) and performance metrics; DLL: error detection, MAC, Ethernet; DV and LS routing; IPv4 – fragmentation, CIDR, NAT; TCP – flow and congestion control, socket API; DNS and HTTP.	Largest change. Removed from explicit mention: OSI/TCP-IP stack enumeration, framing, Ethernet bridging, shortest-path and flooding routing, ARP, DHCP, ICMP, UDP, SMTP, FTP, Email. Newly explicit: performance metrics, socket API. Retained topics now carry the full CN weight.
All other sections	—	Identical to 2026	No change.
General Aptitude	—	Identical to 2021–2026	No change.

2.4 Terminology changes worth noting

- "Machine instructions" → "Instruction set"; "control unit" → "Design of control unit – hardwired and microprogrammed"; "memory hierarchy: cache, main memory and secondary storage" → "memory interfacing and hierarchy: performance, cache memory mapping".
- "Concept of layering: OSI and TCP/IP Protocol Stacks" → "Principles of Layering". Questions that merely ask which layer a protocol belongs to are therefore less likely; questions on why layering exists and on encapsulation remain fair.
- "sockets" → "socket API", suggesting questions on the order and role of socket calls rather than only the concept of a socket.
- "Minimization" → explicit "algebraic technique, Karnaugh map, tabular method". The tabular (Quine–McCluskey) method is now unambiguously examinable.

2.5 Historical syllabus changes (context for old PYQs)

Year	Change	Consequence for using old PYQs
2016	Removed Software Engineering, Web Technologies, Information Systems, Numerical Methods.	Ignore all pre-2016 questions on these topics.

Year	Change	Consequence for using old PYQs
2021	Added "Monoids"; made "pipeline hazards", "system calls", "CPU and I/O scheduling" explicit; added compiler data-flow analyses (constant propagation, liveness, CSE); removed IPv6, Wi-Fi basics, network security; GA restructured into Verbal/Quantitative/Analytical/Spatial; MSQ question type introduced.	Old network-security and IPv6 questions are out of scope. Data-flow analysis questions from 2021 onward define a new sub-topic.
2027	See 2.3.	Old CN questions on ARP/DHCP/ICMP/UDP/SMTP/FTP/OSI layers and old COA disk-geometry questions are now low priority.

Every other topic in the 2000–2026 question archive maps to a topic still in the 2027 syllabus, which is why a 20-year archive remains valid evidence for 2027.

Section 4 — Twenty-Plus Years of GATE CSE Papers: Year-Wise and Long-Term Analysis

4.1 The archive used (FACT)

Period	Papers	Pattern	How used in this book
2000–2005	6 papers	150 marks. 2000–2002: an objective Section A (1-mark and 2-mark MCQs) and a longer-answer Section B; 2003–2005: fully objective, about 90 MCQs. Only the objective questions survive in public archives as objective PYQs.	Qualitative: topic presence, long-term recurrence of concepts.
2006–2009	4 papers	2006–2008: 85 questions / 150 marks; 20 one-mark and 65 two-mark MCQs, including common-data and linked-answer pairs. 2009: 100 marks. No General Aptitude.	Topic presence and recurrence; approximate weightage (2009 tabulated).
2010–2013	4 papers	65 questions / 100 marks; General Aptitude (10 questions, 15 marks) introduced in 2010; all MCQ; linked-answer pairs continued to 2013.	Full subject-wise weightage.
2014–2017	10 papers (3 sets in 2014, 3 in 2015, 2 in 2016, 2 in 2017)	65 / 100; Numerical Answer Type (NAT) introduced 2014; online CBT; linked questions dropped. Old syllabus (with Software Engineering / Web Technologies) until 2015; new syllabus from 2016.	Full weightage per set; NAT trend.
2018–2020	3 papers (single session each)	65 / 100; MCQ + NAT.	Full weightage.
2021–2026	10 papers (2 shifts in 2021, single in 2022 and 2023, 2 shifts in 2024, 2025, 2026)	65 / 100; MSQ introduced 2021; revised syllabus 2021; GA restructured.	Full weightage per shift where available; type mix.

Total: 37 CS papers/sets across 27 examination years. The 2027 paper will be the 38th in this archive and the first under the 2027 syllabus.

Sources for subject-wise weightage: gateexam.info (2009–2019), GeeksforGeeks (2018–2023, 2025, 2026 shift 1 and 2), careers360 (2020–2023), goclasses.in (2023–2025), PW (2026 shift 2). Full URLs in Section 16.

4.2 Master weightage table, 2009–2026 (FACT — analyst compilations; approximate)

Marks per paper. Where a year had several sets, the value is the average across sets unless stated. Sources disagree by ±2–3 marks in some years because borderline questions are classified differently (recurrences: Discrete Maths vs Algorithms; C-code complexity: PDS vs Algorithms); the values below are the series used for all statistics in this book.

Subject	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026
Programming & DS	7	13	7	12	10	14.7*	18.7*	13	10	10	12	12	6	9	11	8	10	9
Algorithms	14	7	9	8	8	*	*	8	9	10	7	6	11	7	6	8	8	10
Operating System	16	7	8	9	12	10	7	6	9	8	10	10	7	9	7	10	8	6
Databases	10	7	5	11	7	4.3	5.7	4	5	6	8	8	8	7	5	8	8	7
Computer Networks	10	8	6	9	7	9	8	8	10	7	10	6	7	11	8	9	8	6
COA	5	9	9	7	11	7.3	5.3	10	6	11	4	11	10	9	12	9	8	9
Theory of Computation	9	7	9	6	8	5.3	5	10	9	8	8	9	9	7	9	7	9	7
Compiler Design	2	4	3	4	3	6	4	5	3	4	6	4	8	4	5	8	5	5
Digital Logic	5	7	8	3	3	2.3	5.7	3	3	4	9	4	4	3	6	5	6	5

Subject	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026
Discrete Mathematics	—	—	—	—	—	—	—	5	7	10	5	10	9	12	6	—	5	5
Engineering Maths (LA/Calc/Prob)	—	—	—	—	—	—	—	13	14	7	6	5	6	7	4	—	10	4
Mathematics total (DM + EM)	—	—	—	—	—	25.7	21	18	21	17	11	15	15	19	10	13	15	9
Old-syllabus topics (IS/SE/Web)	6	0	7	0	3	0.3	4.7	—	—	—	—	—	—	—	—	—	—	—
General Aptitude	0	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15

* In 2014 and 2015 the source merges Data Structures and Algorithms (14.7 and 18.7 average marks); the two are shown together in the PDS row. 2026 uses the GeeksforGeeks shift-1 marks table; shift 2 is only available as question counts (PDS 8, EM 7, COA 6, CN 6, OS 5, DBMS 4, CD 4, DL 4, Algorithms 4, TOC 3, DM 3). 2024 mathematics split is not available separately.

Note on 2009: the paper had no GA section and 100 marks; the very high OS (16) and Algorithms (14) values reflect that year's paper, not a trend.

4.3 Aggregate statistics per subject (ANALYSIS, computed from the table above)

Subject	Papers with data	Average marks	Min	Max	Average 2009–2017	Average 2018–2026	Direction
Programming & DS	16	9.9	6	13	10.3	9.7	stable
Algorithms	16	8.5	6	14	9.0	8.1	slightly falling
Operating System	18	8.8	6	16	9.3	8.3	slightly falling
Databases	18	6.9	4	11	6.6	7.2	stable
Computer Networks	18	8.2	6	11	8.3	8.0	stable
COA	18	8.5	4	12	7.7	9.2	rising
Theory of Computation	18	7.8	5	10	7.6	8.1	stable
Compiler Design	18	4.6	2	8	3.8	5.4	rising
Digital Logic	18	4.8	2.3	9	4.4	5.1	stable
Discrete Mathematics	10	7.4	5	12	6.0	7.8	rising
Engineering Maths (LA/Calc/Prob)	10	7.6	4	14	13.5	6.1	falling (2016–2017 outliers)
Mathematics total	13	16.1	9	25.7	21.4	13.8	falling to the official 13

Reading the table:

- **Programming & DS plus Algorithms** together average about 18 marks per paper, the largest block after mathematics and GA. Their split moves year to year (2021: 6 + 11; 2026 shift 1: 9 + 10), which is why the two must be prepared as one unit.
- **OS, COA, CN and TOC** form a stable middle tier at 8–9 marks each. COA has been the most reliably high recently (11, 10, 9, 12, 9, 8, 9 marks in 2020–2026 with 2019 the only dip).
- **DBMS** sits at 7–8 marks in the recent period with 2023 (5) the only weak year.
- **Compiler Design** has doubled its 2009–2013 level (2–4) to a recent 4–8; **Digital Logic** stays at 3–6 with occasional spikes (2011: 8, 2019: 9).
- **Mathematics** was over-weight in 2014–2017 (up to 30 marks in 2014 set 1) and has settled at the official 13 marks since 2018, split roughly 5–10 Discrete and 4–8 Linear Algebra/Calculus/Probability.

Classification of subjects by weight for 2027 (ANALYSIS):

Band	Subjects	Expected 2027 marks (each)
High	Programming & DS, Algorithms, Engineering Mathematics (DM + LA/Calc/Prob), General Aptitude	9–12 (DS/Algo each), 13 (Maths), 15 (GA)
Medium	OS, COA, CN, TOC, DBMS	6–10
Low	Compiler Design, Digital Logic	4–7

- Concept: number of processes created by n fork() calls (2^n − 1 children)
- Concept: fork inside loops and conditionals; return values (0 in child, PID in parent)
- Concept: copy-on-write address space; variable values after fork; stdio buffering and duplicated output
- Concept: exec replaces the image, PID unchanged; wait/zombie/orphan
- Subtopic: Processes
- Concept: process states (new, ready, running, waiting/blocked, terminated) and legal transitions
- Concept: PCB contents, what is saved on a context switch, dispatcher
- Concept: long-term / short-term / medium-term scheduler, degree of multiprogramming
- Subtopic: Threads
- Concept: what threads share (code, data, heap, open files) vs private (stack, registers, PC)
- Concept: user-level vs kernel-level threads; many-to-one, one-to-one, many-to-many
- Concept: blocking system call in a user-level thread blocks the process

- **Topic: Inter-process communication**
  - Subtopic: Shared memory vs message passing
  - Concept: pipes (unidirectional, related processes), named pipes/FIFOs, blocking semantics
  - Concept: synchronous vs asynchronous send/receive, direct vs indirect (mailbox)

- **Topic: Concurrency and synchronization**
  - Subtopic: Critical-section problem
  - Concept: mutual exclusion, progress, bounded waiting; checking a given algorithm
  - Concept: turn-variable (strict alternation), flag-variable, Peterson's algorithm and its wrong variants
  - Concept: race conditions; number of possible final values of a shared counter
  - Subtopic: Hardware support
  - Concept: test-and-set, swap, compare-and-swap, fetch-and-add; bounded waiting with a waiting[] array
  - Subtopic: Semaphores
  - Concept: binary vs counting; wait/signal (P/V) semantics, negative values = blocked count
  - Concept: ordering/precedence with semaphores, possible output strings
  - Concept: deadlock and ME violation from wrong ordering of wait() calls
  - Subtopic: Classical problems
  - Concept: bounded-buffer producer–consumer (empty/full/mutex)
  - Concept: readers–writers (readers-preference; writer starvation)
  - Concept: dining philosophers (deadlock-avoiding strategies)
  - Subtopic: Monitors and condition variables
  - Concept: implicit mutual exclusion, wait releases the lock, signal has no memory

- **Topic: Deadlock**
  - Subtopic: Characterization
  - Concept: four necessary conditions; resource-allocation graph; cycle vs deadlock for single/multi-instance resources
  - Subtopic: Prevention and avoidance
  - Concept: breaking each condition; minimum resources so that deadlock is impossible: sum(need_i − 1) + 1
  - Concept: Banker's algorithm: Need = Max − Allocation, safety algorithm, safe sequences, resource-request algorithm
  - Subtopic: Detection and recovery
  - Concept: wait-for graph; detection algorithm with Allocation/Request matrices; number of deadlocked processes

Topic / Subtopic	F1	F2	F3	F4	F5	F6	F7	F8	Score	Band
DM — Recurrence relations (solving, recurrences from counting)	8	7	8	9	7	8	8	7	77.5	High
LA — Systems of linear equations & rank/nullity	8	8	7	9	6	8	7	7	76.5	High
DM — Graph connectivity, trees, spanning trees, cut vertices	8	7	8	9	6	8	8	7	76.5	High
Calc — Limits (L'Hôpital, series, sequences)	8	8	6	9	6	9	7	6	74.5	High
DM — Graph colouring / chromatic number	7	7	7	9	7	8	7	7	73.5	High
LA — Determinants & matrix properties (symmetric, orthogonal, idempotent)	8	7	7	9	6	8	7	6	73.5	High
DM — Functions (counting, composition, injective/surjective)	7	8	7	8	6	9	7	6	72.5	High
Calc — Maxima and minima	7	7	6	9	7	9	6	6	71.0	High
DM — Groups & monoids (axioms, orders, Lagrange)	7	7	6	9	7	7	6	6	70.0	High
DM — Partial orders & lattices (Hasse, LUB/GLB, distributive)	7	6	7	9	7	6	7	6	69.5	High
Calc — Definite / improper integration	7	7	5	9	6	9	6	5	67.5	High
Calc — MVT, continuity & differentiability	6	6	5	9	7	7	6	5	63.5	Moderate
DM — Generating functions	6	6	5	9	6	7	5	6	63.0	Moderate
DM — Matching (perfect/maximum, Hall)	5	6	4	9	7	7	5	6	60.5	Moderate
LA — LU decomposition	4	5	3	9	7	7	4	6	54.5	Moderate
Stats — Mean, median, mode, standard deviation	4	5	3	9	6	8	4	5	53.0	Moderate

PREDICTION — explanation of key scores. Eigenvalues and Bayes theorem score 94 because they have appeared in effectively every paper since 2003 (F1 = F2 = 10), generate many variants (powers, inverses, polynomial-in-A; multi-source posteriors, sequential conditioning) and are perfectly suited to NAT. Graph counting via degree sums and Euler's formula scores 88.5 because it is the most frequently repeated DM computation and appears in all three IIT Madras papers. Propositional logic reaches Very High mainly on frequency and syllabus centrality even though it is usually a 1-mark MCQ, so its marks-yield is lower than its score suggests. FOL, counting, relations and recurrences cluster at 77–79 — each is nearly guaranteed to appear somewhere in the two shifts but not in every single shift. LU decomposition and descriptive statistics score only 53–55 despite explicit syllabus naming, because their historical frequency is low and the variants are few; they remain "cheap to prepare" and should not be skipped, but they should not displace Tier 1 practice. Matching is the one syllabus-named graph topic with a genuine "due" signal (F5 = 7) after sparse appearances.

Subject prediction summary (PREDICTION)

PREDICTION.

Rank	Topic	Subtopic	Historical Frequency	Recent Trend	Expected Format	Difficulty	Prediction Confidence
1	Linear Algebra	Eigenvalues, trace/det identities, matrix powers	Very high (every year)	Stable	NAT 1–2M	Easy–Moderate	Very High (94%)
2	Probability	Conditional probability, Bayes theorem	Very high (every year)	Stable	NAT 2M	Moderate	Very High (94%)
3	Discrete Maths	Graph counting: degrees, edges, Euler formula, planarity	Very high	Stable–rising	NAT 1–2M	Moderate	Very High (89%)
4	Probability	Expectation and variance (games, indicators, functions of RVs)	High	Rising	NAT 1–2M	Moderate	Very High (85%)
5	Probability	Named distributions (uniform, exponential, Poisson, binomial, normal)	High	Rising	NAT/MCQ 1–2M	Moderate	Very High (81%)
6	Discrete Maths	Propositional logic: tautology/equivalence/assignments	High	Stable	MCQ/MSQ 1M, NAT 1–2M	Easy–Moderate	Very High (81%)
7	Discrete Maths	First-order logic: translation, negation, quantifier order	High	Stable	MCQ 1–2M	Moderate	High (79%)
8	Discrete Maths	Counting: P&C, pigeonhole, inclusion–exclusion	High	Stable	NAT 1–2M	Moderate	High (79%)
9	Discrete Maths	Relations: counting with properties, equivalence, closures	High	Rising	NAT 1–2M, MSQ 1M	Moderate	High (79%)
10	Discrete Maths	Recurrence relations	High	Stable	NAT/MCQ 2M	Moderate–Difficult	High (78%)
11	Linear Algebra	Systems of equations, rank, nullity	High	Rising	MCQ/NAT 2M	Moderate	High (77%)
12	Discrete Maths	Connectivity, trees, spanning trees	High	Stable	NAT/MSQ 1–2M	Moderate	High (77%)
13	Calculus	Limits	High	Stable	NAT/MCQ 1M	Easy–Moderate	High (75%)
14	Discrete Maths	Colouring / chromatic number	Moderate–High	Stable	MCQ/NAT 1–2M	Moderate	High (74%)
15	Linear Algebra	Determinants and special matrices	High	Stable	NAT/MSQ 1–2M	Easy–Moderate	High (74%)
16	Discrete Maths	Functions: counting, composition	Moderate–High	Rising	NAT/MSQ 2M	Moderate	High (73%)

Rank	Topic	Subtopic	Historical Frequency	Recent Trend	Expected Format	Difficulty	Prediction Confidence
17	Calculus	Maxima and minima	Moderate–High	Stable	NAT 2M	Moderate	High (71%)
18	Discrete Maths	Groups and monoids	Moderate–High	Rising (MSQ)	NAT/MSQ 1–2M	Moderate	High (70%)
19	Discrete Maths	Posets and lattices	Moderate–High	Stable	MSQ/NAT 2M	Moderate–Difficult	High (70%)
20	Calculus	Integration	Moderate	Stable	NAT 1–2M	Moderate	High (68%)
21	Calculus	MVT, continuity and differentiability	Moderate	Stable	MCQ/NAT 1–2M	Moderate	Moderate (64%)
22	Discrete Maths	Generating functions	Moderate	Stable	MCQ/NAT 2M	Moderate–Difficult	Moderate (63%)
23	Discrete Maths	Matching	Low–Moderate	"Due"	NAT 2M	Moderate	Moderate (61%)
24	Linear Algebra	LU decomposition	Low	Flat	NAT 2M	Moderate	Moderate (55%)
25	Statistics	Mean, median, mode, SD	Low	Flat	NAT 1M	Easy	Moderate (53%)

Top 10 most important topics

1. Eigenvalues and eigenvectors — the one computation present in every paper since 2003; trace/determinant shortcuts make it a 60-second NAT.
2. Conditional probability and Bayes theorem — a 2-mark NAT in essentially every paper; framing changes, arithmetic does not.
3. Graph counting via degree sums and Euler's formula — the DM workhorse; combined-constraint variants are the recent pattern.
4. Expectation and variance — indicator-variable and "game" expectations recur yearly and link to Algorithms/OS.
5. Named distributions — uniform variance, exponential memorylessness and Poisson/binomial numerics cover the explicit syllabus list.
6. Propositional logic — reliable 1-mark MCQ/MSQ; low cost, high certainty.
7. First-order logic — translation and quantifier-order items are the most-repeated conceptual trap in DM.
8. Counting (pigeonhole, inclusion–exclusion, stars-and-bars) — the source of most 2-mark DM NATs.
9. Relations and functions counting — 2^(n(n−1)/2)-style formulas and onto-function counts recur almost annually since 2015.
10. Recurrence relations — always 2 marks, links directly to Algorithms; the characteristic-equation method is enough.

Top predicted question concepts

- Determinant or trace of a polynomial in A (A^2 + I, A^3 − 2A) from given eigenvalues, or from a 2×2 matrix's trace and determinant.
- Values of parameters λ, μ for which a 3×3 system has infinitely many / no solutions.
- Rank of A^T A or nullity of a rectangular matrix from given rank.
- Bayes posterior with three sources and unequal defect/error rates (3–4 decimal NAT).
- Conditional probability given a dice sum or given "at least one head".
- Expected number of distinct values / matches / trials using linearity of expectation.
- E[X^2] or Var(aX + b) for a uniform or exponential variable.
- Exponential memorylessness as a 1-mark MCQ; Poisson "exactly k events in t minutes" NAT.
- Edges of a planar graph with all faces of a given length; leaves of a tree from internal degree data.
- Chromatic number of a complement, a wheel, or K_n minus a matching.
- Number of perfect matchings in K_2n or K_n,n; spanning trees of K_m,n.
- Negation of a nested-quantifier statement; identifying the valid one among ∀∃/∃∀ implications.
- Number of truth assignments for which an implication-based formula is false.
- Number of reflexive-and-symmetric relations; number of equivalence relations with a given number of classes.
- Number of onto functions from a 6-set to a 3-set; whether g∘f is a bijection for given floor/ceiling maps.

- Order of an element from a^k = e data and |G|; number of elements of a given order in Z_n; which of several structures are groups.
- Whether a described 6-element poset is a lattice; whether D_n is Boolean/distributive.
- Closed-form value a_n for a second-order linear recurrence at n = 8–12; count of ternary strings avoiding "00".
- Coefficient extraction from (1 − x^k)/(1 − x)^3-type generating functions.
- Absolute maximum of a cubic on a closed interval where an endpoint ties an interior critical point; MVT c-value; ∫ x e^x dx.

Tier assignment

- **Tier 1 (Very High):** Eigenvalues/eigenvectors; Conditional probability & Bayes; Graph degree/edge counting & Euler formula; Expectation & variance; Named probability distributions; Propositional logic.
- **Tier 2 (High):** First-order logic; Counting (P&C, pigeonhole, inclusion–exclusion); Sets & relations; Recurrence relations; Systems of equations & rank; Graph connectivity/trees; Limits; Graph colouring; Determinants & special matrices; Functions; Maxima/minima; Groups & monoids; Posets & lattices; Integration.
- **Tier 3 (Moderate):** MVT/continuity/differentiability; Generating functions; Matching; LU decomposition; Mean/median/mode/SD.
- **Tier 4 (Lower):** Monoid-only theory questions (semigroups without group structure); multivariable calculus beyond single-variable extrema; statistics beyond the four named measures (regression, hypothesis testing — not in syllabus); Numerical Methods (removed 2016).

Question patterns per high-probability topic (PREDICTION)

PREDICTION. For each Tier 1–2 topic, the variations GATE has used and is expected to reuse.

- **Eigenvalues/eigenvectors**
- Given eigenvalues of A, find det or trace of A^k, A^-1, A + kI, or a polynomial in A.
- Given a 2×2 matrix, find eigenvalues from trace and determinant; find the eigenvector for the largest eigenvalue.
- Eigenvalues of triangular, rank-1 (all rows proportional), idempotent, nilpotent, orthogonal, or symmetric matrices.
- Cayley–Hamilton: express A^-1 or A^n as αA + βI and report α + β or an entry.
- MSQ: true statements about eigenvalues of symmetric/orthogonal matrices, diagonalizability, distinct eigenvalues ⇒ independent eigenvectors.
- **Conditional probability & Bayes**
- Multiple sources (machines, suppliers, students) with prior shares and defect/error rates; posterior of a source.
- Diagnostic test with sensitivity/specificity and disease prevalence.
- Dice: probability of an event given the sum or given "at least one six".
- Urns/cards without replacement; conditional on colour of the first draw.
- Independence checks: P(A ∩ B) = P(A)P(B) from P(A ∪ B) data (MSQ).
- **Graph degree/edge counting & Euler formula**
- Vertices or edges from a partial degree distribution; leaves of a tree from internal degrees.
- Regular graphs: edges of a k-regular graph on n vertices; existence conditions (nk even).
- Planar graph with all faces of length 3/4/5: find e or f; maximum edges e ≤ 3v − 6, bipartite e ≤ 2v − 4.
- Self-complementary graphs: n ≡ 0, 1 (mod 4); edges = n(n − 1)/4.
- Max edges in a disconnected graph; min edges for connectivity with k components.
- **Expectation & variance**
- Expected number of trials to first success (geometric); expected number of tosses to see a pattern.
- Linearity with indicators: expected fixed points, expected distinct faces, expected collisions.
- E[X^2] = Var + mean^2 for uniform/exponential; Var(aX + bY) for independent X, Y.
- Expected value of a game payoff with a fair/biased die.

- **Named distributions**
- Uniform: P(interval), variance $(b - a)^2/12$, $E[g(X)]$.
- Exponential: memorylessness $P(X > s + t \mid X > s)$, $P(X > \text{mean})$, min of two exponentials.
- Poisson: P(exactly k) with rate rescaled to the time window; P(at least 1).
- Binomial: P(at least k of n); mean np, variance $np(1 - p)$.
- Normal: expressions in Φ, symmetry $P(X > \mu) = 0.5$, mean = median = mode.
- **Propositional logic**
- Which of four formulas is (not) a tautology; which is equivalent to $p \to q$; contrapositive vs converse.
- Number of truth assignments making a given formula true/false.
- Valid inference: from premises choose the derivable conclusion.
- MSQ: statements about satisfiability/validity/equivalence.
- **First-order logic**
- Translate "every ... that ... is ..." / "some ... do not ..." with two or three predicates.
- Negate a nested ∀∃ statement with an implication inside.
- Which of $\forall x \exists y \to \exists y \forall x$, $\exists x(P \wedge Q) \to \exists xP \wedge \exists xQ$, $\forall x(P \vee Q) \leftrightarrow \forall xP \vee \forall xQ$ are valid (MSQ).
- **Counting**
- Minimum picks guaranteeing a pair with a fixed sum or same remainder (pigeonhole).
- Integers $\leq N$ divisible by none of several primes; derangement-type restricted permutations (inclusion–exclusion).
- Non-adjacent selections $C(n - k + 1, k)$; bit strings with constraints.
- Integer solutions with upper bounds (stars-and-bars minus overflow).
- **Sets & relations**
- Count relations that are reflexive and symmetric / antisymmetric / irreflexive.
- Equivalence relations with a given number of classes (Stirling numbers), Bell numbers for $n \leq 5$.
- Properties of $|a - b| \leq k$, a divides b, $a \equiv b \pmod{m}$.
- Size of transitive closure; composition $R \circ S$.
- **Recurrence relations**
- Second-order homogeneous with distinct/repeated roots; evaluate at n = 8–12.
- Set up and solve a recurrence for strings avoiding a pattern (binary/ternary).
- Non-homogeneous with constant or linear forcing term; particular solution.
- **Systems of equations & rank**
- λ, μ conditions for unique/infinite/no solutions in a 3×3 system.
- Nullity from rank; number of independent solutions of a homogeneous system; $\text{rank}(A^T A)$.
- Rank of a product ≤ min of ranks; rank of a matrix with parameter.
- **Graph connectivity & trees**
- MSQ on sufficient conditions for connectivity (degree $\geq n/2$, edges $> C(n - 1, 2)$).
- Spanning trees of K_n, $K_{m,n}$, C_n, ladder graphs; Cayley's formula.
- Cut vertices/bridges in described graphs; $\kappa(G) \leq \lambda(G) \leq \delta(G)$.
- **Limits**
- 0/0 forms with e^x, sin x, ln(1 + x), cos x via L'Hôpital or Maclaurin series.
- $(1 + a/n)^{bn} \to e^{ab}$; limits of rational functions at infinity.
- **Graph colouring**
- χ of cycles, wheels, complete graphs, K_n minus a matching, complements of cycles.
- Bounds: $\omega(G) \leq \chi(G) \leq \Delta + 1$; bipartite iff $\chi = 2$.
- **Determinants & special matrices**
- det(kA), det(AB), $\det(A^{-1} B A)$, effect of row operations; MSQ of true/false properties.

Subject	Entries in Top 100	Tier 1 entries (Ranks 1–50)	Expected 2027 marks (subject chapters)	Why the count looks like this
PDS	12	6	9–12	Highest single score in the table (96.5) and six Tier 1 rows; together with Algorithms it forms the 18–20-mark DS/Algo block.
OS	9	5	6–10	Five perennials at 81–96.5 plus four "one question in most papers" rows; the most literal recurrence of any subject.
CN	9	4	6–11	Four computation templates in Tier 1 and a deep High band; the 2027 syllabus cut concentrates marks on the retained rows.
COA	8	4	8–11	Sharpest top (cache 93, pipeline 90); the most reliably high-weight medium subject 2020–2026; two 2027-syllabus additions sit at 73.
DBMS	8	4	6–10	Three procedures (precedence graph, closure, SQL row tracing) plus B+ tree arithmetic form a tight, predictable core.
Digital Logic	7	4	4–7	Concentrated subject: four Very High rows map onto the three clauses of the syllabus text; low marks but the best marks-per-hour.
TOC	7	6	7–9	Six Tier 1 rows carry most TOC marks; everything below 76 is tested inside those rows rather than as separate items, hence only one Tier 2 entry.
Compiler Design	6	3	4–8	Steepest fall-off after its top three (SDT, LR, LL(1)); doubled its 2009–2013 weight but still the smallest technical subject.
Total	**100**	**50**	85 subject marks (+ 15 GA)	—

Counts follow the marks: the three high-weight technical blocks (Mathematics, Algorithms, PDS) supply 46 of the 100 entries; the five medium subjects supply 41; the two low-weight subjects supply 13. General Aptitude (15 marks) is not scored on the concept model and is treated in Section 10.

9.3 Ranks 1–25

PREDICTION. Ranks 1–25 are all Tier 1 (Very High). Every one of them scores 86.5 or more.

Rank	Subject	Topic	Subtopic	Concept	Why it matters	Historical evidence	Recent evidence (2018–2026)	Expected question type	Expected difficulty	Prediction confidence
1	PDS	Programming in C	Pointer / array / 2-D tracing	2-D array with mixed `int *` and `int (*)[n]` pointers; printed sum of selected elements; pointer arithmetic	Highest score in the master table; a pure 2-mark NAT present in every paper and every shift.	Pointer/array items in essentially every paper 2003–2026 (approx.).	2-D pointer NATs 2020, 2021, 2023 (approx.); present in every shift since 2018.	NAT 2M (+ MCQ 1M)	Moderate	Very High (96.5)
2	OS	CPU scheduling	Gantt-chart metrics	SRTF / RR / priority / FCFS / SJF with staggered arrivals; average waiting or turnaround; context-switch cost; response time	Most question-productive OS concept (F3 = 10); one 2-mark NAT in nearly every paper.	SRTF items 2006–2016 and RR items 2004–2017 in most papers (approx.).	SRTF 2019, 2020, 2022, 2025; RR 2020, 2021, 2024, 2025; MLQ variant asked 2026.	NAT 2M	Moderate–Difficult	Very High (96.5)
3	Eng. Maths	Linear Algebra	Eigenvalues and eigenvectors	det/trace of A^k, A^-1, A + kI or a polynomial in A from eigenvalues; Cayley–Hamilton; triangular/rank-1/idempotent/orthogonal matrices	Present in effectively every paper since 2003; trace/determinant shortcuts make it a 60-second NAT.	A^k / A + kI items 2013, 2015, 2017; A^-1 as polynomial 2014, 2015; rank-1 forms 2010, 2016.	2018, 2020, 2022, 2024 eigenvalue-power items; 2021 all-ones matrix.	NAT 1–2M	Easy–Moderate	Very High (94.0)
4	Eng. Maths	Probability	Conditional probability and Bayes	Posterior of one of 2–3 sources with prior shares and defect rates; dice/urn conditioning; independence checks	A 2-mark NAT in essentially every paper; the framing changes, the arithmetic does not.	Bayes posterior 2010, 2014, 2016; conditional dice 2011, 2012, 2016.	Posterior NATs 2018, 2020, 2022; conditional dice 2019, 2021.	NAT 2M	Moderate	Very High (94.0)

Rank	Subject	Topic	Subtopic	Concept	Why it matters	Historical evidence	Recent evidence (2018–2026)	Expected question type	Expected difficulty	Prediction confidence
5	OS	Synchronization	Code property checks	ME / progress / bounded waiting / deadlock of turn-flag or semaphore code; modified Peterson's; semaphore-gated output strings	Perennial since 2001; MSQ form gives no partial credit, so every property must be checked.	Two-process ME analysis 2001, 2003, 2006, 2007, 2009, 2010, 2013, 2015–2017.	Every paper 2018–2026; now a 2-mark MSQ; gated-output items 2018, 2020, 2023, 2026.	MSQ 2M	Moderate–Difficult	Very High (93.5)
6	COA	Memory hierarchy	Cache mapping arithmetic	Tag/index/offset bits; sets and lines; cache size or associativity from tag width; same-set addresses; tag-directory size	Named "cache memory mapping" in the 2027 syllabus; asked in almost every paper since 2014.	2011, 2012 metadata-size forms; 2014 field-width forms; 2017 tag-width inverse.	2018 symbolic tag width; 2019 fully associative widths; 2021 associativity from a 22-bit tag; cache item reported 2026.	NAT 1–2M	Easy–Moderate	Very High (93)
7	DBMS	SQL	Row count / meaning of nested queries	Correlated NOT EXISTS (division); NULL under NOT IN; > ALL over an empty subquery; correlated vs uncorrelated aggregate	Every paper since 2010; the most NAT-friendly DBMS topic; "tricky SQL" flagged by analysts 2023–2026.	2012 (> ALL), 2013, 2014 S3 (NOT EXISTS), 2016 S2, 2017 S2 (approx.).	Unbroken 2016–2022 streak; NULL items 2019, 2023, 2026; 2022 NOT EXISTS ... EXCEPT count.	NAT 2M	Moderate–Hard	Very High (93.0)
8	PDS	Recursion	Recursion tracing	Return value; call count with a global/static counter; print-before/after order; static accumulator	Second PDS concept scoring 9–10 on both frequency factors; a NAT generator in every shift.	Call-count items 2007, 2013, 2016; print-order items 2004, 2008, 2011, 2014.	Call counts 2018, 2020, 2024, 2026; static/global variants 2019, 2021, 2024; print order 2019, 2022.	NAT 2M	Moderate–Hard	Very High (92.0)
9	OS	Memory management	Paging arithmetic and multi-level page tables	Number of levels; bits per level; page-table memory for a sparse layout; single-level size; combined with TLB	Perennial; 2024–2026 papers combined multi-level paging with a TLB in a single NAT.	Multi-level paging 2003, 2008, 2013, 2015, 2017 (approx.).	2019, 2021, 2024, 2025, 2026 — rising and combined with TLB.	NAT 2M	Moderate–Difficult	Very High (90.5)
10	TOC	Regular and context-free languages	Language-class identification	Which of four languages are regular / DCFL / CFL / not CFL (free-middle, bounded-count, linear comparison, copy)	The only TOC form asked essentially every year since 2004; now a 2-mark MSQ favoured by setters.	Near-annual 2004–2017 (approx.).	Every paper 2018–2026; MSQ form since 2021.	MSQ 2M	Difficult	Very High (90.5)
11	COA	Pipelining	Cycle count with data hazards	4–6 instructions in a 5-stage pipeline with load-use dependence; forwarding on/off; multi-cycle EX; instruction-class cycle tables	"Pipeline hazards" is explicit in the syllabus; asked in consecutive years including 2026.	2008 (bypassing statement); 2010, 2015 multi-cycle forwarding chains.	2021 cycle count with stated forwarding; pipeline item reported 2026.	NAT 2M	Difficult	Very High (90)
12	CN	TCP	Congestion control	cwnd/ssthresh table with timeout and triple-dup-ACK rounds; RTTs to send F bytes; Tahoe vs Reno statements	"Congestion control" is explicit in the 2027 text; most papers since 2012; every retained CN topic gains share after the syllabus cut.	2008 slow-start growth; 2012 AIMD with a timeout in the fifth transmission; 2017 (approx.).	2020, 2022, 2024, 2025 (approx.); "TCP congestion phases" reported 2026.	NAT 2M; MSQ 1M	Moderate–Difficult	Very High (90.0)
13	CN	IPv4	CIDR / subnetting	Hosts per subnet; smallest block for a requirement; broadcast address with a non-octet mask; aggregation; allocation validity	Named "CIDR Notation" in the 2027 syllabus; every paper in some form.	2005, 2006, 2008, 2010, 2012 mask/allocation items.	2019 (/30 among three machines); 2020 block for 1,500 hosts; 2021, 2025 (approx.); subnetting reported 2026.	NAT or MCQ 1–2M	Easy–Moderate	Very High (90.0)
14	CD	Syntax-directed translation	Attribute evaluation	Synthesised value for a 4–6 token input with non-standard operators; print-order schemes; S- vs L-attributed	Appears in about three of four years; "syntax-directed translation" is named verbatim in the syllabus.	Embedded-action SDT 2005, 2009, 2013, 2016; S/L-attributed MCQs 2008, 2011, 2014.	2019, 2021, 2022, 2025 (approx.); near-annual.	NAT 2M (value) or MCQ	Moderate	Very High (90.0)
15	Digital Logic	Boolean algebra	Identities, XOR algebra, equivalence, consensus	The one invalid identity among four; equivalent forms of F (MSQ); consensus in POS; operator defined by a table	The natural 1-mark Digital Logic item; "higher than expected" Boolean weight reported in 2026 Shift 2.	2004, 2007, 2011, 2013, 2016 identity/equivalence items.	2018, 2019, 2020 invalid-identity items; 2026 Shift 2 emphasis.	MCQ 1M or MSQ	Easy–Moderate	Very High (89.5)

What you want	Where it is
The answer key for the test you just attempted	At the end of that test, under "Answer key — Mock Test NN". It lists all 65 answers with their question types, and repeats the marking scheme
The full working for any question	Section 13, under "Mock Test NN — Title" (for example "Mock Test 01 — Tier 1 Concentration"). Every solution ends with the four confidence scores for the concept it tests
All ten keys together, for quick scoring or revision	Section 12

The key page sits immediately after the last question of its own test, so treat the end of the question paper as a hard stop when you are attempting under time.

11.5 How to take and score a test

1. Attempt the full paper in 180 minutes without breaks; GA first (about 15 minutes), then the technical section.
2. Enter NAT answers with the stated precision; treat MSQ as all-or-nothing.
3. Score with the official marking scheme, using the key at the end of that test. A raw score of 70+ on these tests corresponds roughly to the top 300 ranks in recent years, 60–70 to the top 300–1,000 and 50–60 to the top 1,000–3,000 (see Section 14); the general-category qualifying cutoff has been about 27–30 (2026: 30, per published cutoffs).
4. Read the detailed solutions in Section 13 only after scoring.
5. Log every wrong or skipped question by subtopic; after three tests the log is the personal revision list.

Mock Test 01 — Tier 1 Concentration

Design focus: This test packs the maximum number of Tier 1 (Very High probability) concepts from every subject into their most standard GATE forms: SRTF timelines, LRU traces, direct-mapped cache bit-splits, forwarded-pipeline cycle counts, C pointer/2-D array tracing, BST preorder validity, bottom-up build-heap, Dijkstra extraction order, Kruskal totals, Master-theorem recurrences, minimum-state DFA, language-class identification, LL(1)/LR grammar classification, SDT attribute evaluation, SQL on a printed instance, conflict serializability, B+ tree capacity, TCP cwnd trajectories, CIDR sizing, sliding-window sequence bits, K-map prime implicants, IEEE-754 decoding, eigenvalue trace/determinant, Bayes theorem, degree-sum counting and propositional equivalence. The remaining slots are filled from Tier 1 first and Tier 2 second (recurrences, planarity, control-unit design, liveness, TCP handshake, DFS edge classes). Difficulty is moderate: no trick conventions, every convention needed for a unique answer is stated. Solve it as a calibration paper — a score of 60+ here means the Tier 1 core is secure.

Instructions: 65 questions, 100 marks, 180 minutes. Q1–Q10 General Aptitude. Negative marking applies to MCQ only (−1/3 for 1-mark, −2/3 for 2-mark). MSQ: select all correct options; no partial credit. NAT: enter the numeric value to the stated precision.

General Aptitude (Q1–Q10)

Q1 MCQ · 1 mark

Select the option that correctly completes the sentence.

Neither the manager nor the employees ______ aware of the change in policy until it was announced in the meeting.

(A) were
(B) was
(C) is
(D) has been

General Aptitude — verbal: verb–subject agreement

Q2 MCQ · 1 mark

Sculptor : Chisel :: Surgeon : ______

Select the word that completes the analogy in the same relationship as the first pair.

(A) Hospital
(B) Patient
(C) Scalpel
(D) Anaesthesia

General Aptitude — verbal: analogy

Q3 MCQ · 1 mark

A shopkeeper marks an article 40% above its cost price and then allows a discount of 25% on the marked price. The shopkeeper's profit, as a percentage of the cost price, is

(A) 5%
(B) 10%
(C) 15%
(D) 20%

General Aptitude — quantitative: percentages

Q4 MCQ · 1 mark

The idiom "to throw in the towel" most nearly means

(A) to start a quarrel
(B) to clean thoroughly

(C) to celebrate a victory
(D) to give up

General Aptitude — verbal: idioms in context

Q5 MCQ · 1 mark

The next term of the sequence 2, 6, 12, 20, 30, _____ is

(A) 40
(B) 42
(C) 44
(D) 46

General Aptitude — quantitative: series

Q6 MCQ · 2 marks

Read the passage and answer the question.

Cities are typically warmer than their rural surroundings, a phenomenon known as the urban heat island effect. Dark roofs and asphalt absorb sunlight during the day and release the stored heat at night. Recent studies suggest that increasing tree cover by ten percent can reduce local surface temperatures measurably — but only when the added trees are distributed across neighbourhoods rather than concentrated in a few large parks.

Which one of the following is the best inference from the passage?

(A) Large parks have no effect on urban temperatures.
(B) Spreading additional trees across a city reduces surface temperatures more effectively than concentrating the same trees in a few parks.
(C) Replacing asphalt with dark roofs reduces the urban heat island effect.
(D) Rural areas do not absorb heat from sunlight.

General Aptitude — verbal: reading comprehension

Q7 MCQ · 2 marks

The table gives the number of units sold of four products in two years.

Product	Units sold in 2024	Units sold in 2025
P	120	150
Q	200	180
R	80	120
S	100	110

Which product had the highest percentage increase in units sold from 2024 to 2025, and what is the percentage increase in the total units sold (all four products together) from 2024 to 2025?

(A) R; 12%
(B) P; 12%
(C) R; 10%
(D) Q; 12%

General Aptitude — quantitative: data interpretation

Q8 MCQ · 2 marks

Five people P, Q, R, S and T occupy five seats numbered 1 to 5 from left to right in a single row, one person per seat. The following are known:

(i) Q sits immediately to the right of P.
(ii) S sits in seat 1.
(iii) R is not adjacent to S.
(iv) T does not sit at either end.
(v) T is adjacent to Q.
Who sits in seat 4?

(A) P
(B) Q
(C) T
(D) R

General Aptitude — analytical: deduction / seating arrangement

Q9 NAT · 2 marks

A and B working together can complete a job in 12 days; B and C together can complete it in 15 days; A and C together can complete it in 20 days. Working alone, A completes the job in _____ days. (Enter an integer.)

General Aptitude — quantitative: work and time

Q10 MCQ · 2 marks

A square sheet of paper is folded in half along a vertical line (right half onto the left half), then folded in half along a horizontal line (bottom half onto the top half), and finally folded in half along a diagonal of the resulting small square. A single circular hole is then punched through all the layers at a point that does not lie on any fold line. When the sheet is completely unfolded, the number of holes in it is

(A) 4
(B) 6
(C) 8
(D) 16

General Aptitude — spatial: paper folding and punching

Technical Section — 1-mark questions (Q11–Q35)

Q11 MCQ · 1 mark

What is printed by the following C program?

```
#include <stdio.h>
int main() {
    int a = 7, b = 2, c;
    c = a / b * b + a % b;
    printf("%d ", c);
    c = (a > b) ? a & b : a | b;
    printf("%d ", c);
    c = a << 1 ^ b;
    printf("%d", c);
    return 0;
}
```

(A) 7 2 12
(B) 7 2 56
(C) 6 2 12
(D) 7 7 12

Programming and Data Structures — C operators and precedence

Q12 MCQ · 1 mark

Which one of the following propositional formulas is logically equivalent to $(p \rightarrow r) \wedge (q \rightarrow r)$?

Answer key — Mock Test 01

Attempt the whole paper before reading this page. Score with the official scheme: a correct 1-mark question gives +1 and a correct 2-mark question +2; a wrong MCQ costs 1/3 of a mark (1-mark) or 2/3 of a mark (2-mark); MSQ and NAT carry no negative marking, and an MSQ scores only if every correct option is selected and no incorrect one is. NAT answers are given to the precision stated in each question.

This paper contains 34 MCQ, 12 MSQ and 19 NAT questions (30 one-mark and 35 two-mark, 100 marks).

Q	Answer	Type	Q	Answer	Type	Q	Answer	Type
1	A	MCQ	23	A, C	MSQ	45	D	MCQ
2	C	MCQ	24	B	MCQ	46	6	NAT
3	A	MCQ	25	C	MCQ	47	A	MCQ
4	D	MCQ	26	A, C	MSQ	48	10	NAT
5	B	MCQ	27	11	NAT	49	A, C	MSQ
6	B	MCQ	28	6	NAT	50	17	NAT
7	A	MCQ	29	C	MCQ	51	A, B, D	MSQ
8	C	MCQ	30	126	NAT	52	C	MCQ
9	30	NAT	31	D	MCQ	53	B	MCQ
10	C	MCQ	32	B	MCQ	54	D	MCQ
11	A	MCQ	33	A, C, D	MSQ	55	A, C	MSQ
12	B	MCQ	34	D	MCQ	56	A, B, C	MSQ
13	16	NAT	35	A	MCQ	57	A, C, D	MSQ
14	A	MCQ	36	6.5	NAT	58	B	MCQ
15	D	MCQ	37	0.31 (accepted range 0.31 to 0.32)	NAT	59	B	MCQ
16	63	NAT	38	35	NAT	60	A, B, C	MSQ
17	A	MCQ	39	4	NAT	61	A, C, D	MSQ
18	7	NAT	40	12	NAT	62	A, C, D	MSQ
19	C	MCQ	41	3	NAT	63	C	MCQ
20	C	MCQ	42	B	MCQ	64	A, B, D	MSQ
21	38	NAT	43	11	NAT	65	B	MCQ
22	C	MCQ	44	47	NAT			

Detailed solutions for this test: Section 13, under the heading "Mock Test 01 — Tier 1 Concentration".
Each solution there shows the full working and ends with the four confidence scores (Topic, Concept, Question-pattern, Exact-question) for the concept it tests.

Consolidated keys for all ten tests: Section 12.

Section 13 — Detailed Mock Test Solutions

Each test's solutions follow, in question order. Attempt and score the paper (Section 12) before reading these.

Mock Test 01 — Tier 1 Concentration

Q1 ANSWER **A**

With "neither ... nor ...", the verb agrees with the subject nearer to it. The nearer subject is "the employees" (plural), and the sentence is in the past ("until it was announced"), so the verb is "were". "was"/"is" fail agreement; "has been" fails both agreement and tense.

Based on PYQs: General Aptitude items are written fresh for every paper; this is a standard GA form, not modelled on one past paper
Confidence: Topic 94% | Concept 85% | Question-pattern 76% | Exact-question 16%

Q2 ANSWER **C**

A chisel is the cutting tool with which a sculptor practises the craft; the scalpel is the cutting tool with which a surgeon practises surgery. Hospital (place), patient (object of work) and anaesthesia (a supporting agent, not the surgeon's own cutting instrument) do not preserve the "craftsperson : primary tool" relationship.

Based on PYQs: General Aptitude items are written fresh for every paper; this is a standard GA form, not modelled on one past paper
Confidence: Topic 90% | Concept 81% | Question-pattern 72% | Exact-question 14%

Q3 ANSWER **A**

Let cost price = 100. Marked price = 140. Selling price after 25% discount = 0.75 × 140 = 105. Profit = 5 on 100, i.e., 5%.

Based on PYQs: General Aptitude items are written fresh for every paper; this is a standard GA form, not modelled on one past paper
Confidence: Topic 93% | Concept 84% | Question-pattern 75% | Exact-question 15%

Q4 ANSWER **D**

"To throw in the towel" (from boxing, where a trainer throws a towel into the ring to stop the fight) means to admit defeat, to give up.

Based on PYQs: General Aptitude items are written fresh for every paper; this is a standard GA form, not modelled on one past paper
Confidence: Topic 90% | Concept 80% | Question-pattern 71% | Exact-question 13%

Q5 ANSWER **B**

Differences: 4, 6, 8, 10 — increasing by 2 each time, so the next difference is 12 and the next term is 30 + 12 = 42. Equivalently the n-th term is n(n + 1): 1·2, 2·3, 3·4, 4·5, 5·6, 6·7 = 42.

Based on PYQs: General Aptitude items are written fresh for every paper; this is a standard GA form, not modelled on one past paper
Confidence: Topic 88% | Concept 79% | Question-pattern 70% | Exact-question 12%

Q6 ANSWER **B**

The passage states that the temperature reduction occurs "only when the added trees are distributed across neighbourhoods rather than concentrated in a few large parks" — so distribution is more effective than concentration (B). A overstates ("no effect" is not claimed). C contradicts the passage (dark roofs absorb heat). D is not supported; the passage only says cities are warmer than rural areas.

Based on PYQs: General Aptitude items are written fresh for every paper; this is a standard GA form, not modelled on one past paper
Confidence: Topic 95% | Concept 87% | Question-pattern 78% | Exact-question 16%

Q7 ANSWER **A**

Percentage changes: P: (150 − 120)/120 = 25%; Q: (180 − 200)/200 = −10%; R: (120 − 80)/80 = 50%; S: (110 − 100)/100 = 10%. Highest increase: R.
Totals: 2024 = 120 + 200 + 80 + 100 = 500; 2025 = 150 + 180 + 120 + 110 = 560. Increase = 60/500 = 12%.

Based on PYQs: General Aptitude items are written fresh for every paper; this is a standard GA form, not modelled on one past paper
Confidence: Topic 92% | Concept 83% | Question-pattern 74% | Exact-question 15%

Q8 ANSWER **C**

S is in seat 1 (ii). R is not adjacent to S, so R is not in seat 2 (iii). T is not in seat 1 or 5 (iv). P and Q occupy consecutive seats with Q immediately right of P (i); possible (P, Q) placements among seats 2–5: (2,3), (3,4), (4,5).

- (P,Q) = (2,3): seats 4,5 left for R and T; T cannot be 5, so T = 4, R = 5. Check (v): T (seat 4) adjacent to Q (seat 3) — yes. Arrangement: S P Q T R.
- (P,Q) = (3,4): seats 2,5 for R and T; R cannot be 2, so R = 5, T = 2. T (seat 2) is adjacent to P (3), not Q (4) — violates (v).
- (P,Q) = (4,5): seats 2,3 for R and T; R cannot be 2, so R = 3, T = 2. T (2) is not adjacent to Q (5) — violates (v). Unique arrangement S P Q T R; seat 4 is T.

Based on PYQs: General Aptitude items are written fresh for every paper; this is a standard GA form, not modelled on one past paper
Confidence: Topic 88% | Concept 79% | Question-pattern 69% | Exact-question 12%

Q9 ANSWER **30**

Let a, b, c be the daily work rates. a + b = 1/12, b + c = 1/15, a + c = 1/20. Adding: 2(a + b + c) = 1/12 + 1/15 + 1/20 = (5 + 4 + 3)/60 = 12/60 = 1/5, so a + b + c = 1/10. Then a = (a + b + c) − (b + c) = 1/10 − 1/15 = (3 − 2)/30 = 1/30. A alone needs 30 days.

Based on PYQs: General Aptitude items are written fresh for every paper; this is a standard GA form, not modelled on one past paper
Confidence: Topic 91% | Concept 82% | Question-pattern 73% | Exact-question 14%

Q10 ANSWER **C**

Each fold doubles the number of layers: 1 → 2 (vertical) → 4 (horizontal) → 8 (diagonal). A hole punched through all 8 layers at a point not on any fold line produces one hole per layer, i.e., 8 distinct holes after unfolding (no two coincide because the punch is off every fold line).

Based on PYQs: General Aptitude items are written fresh for every paper; this is a standard GA form, not modelled on one past paper
Confidence: Topic 80% | Concept 71% | Question-pattern 62% | Exact-question 11%

Q11 ANSWER **A**

- `a / b * b + a % b`: `/`, `*`, `%` have equal precedence and associate left to right: (7/2)*2 = 3*2 = 6; 7 % 2 = 1; sum = 7.
- `(a > b) ? a & b : a | b`: `?:` has lower precedence than `&` and `|`, so this is (a > b) ? (a & b) : (a | b). a > b is true, so the value is 7 & 2 = 0111 & 0010 = 0010 = 2.
- `a << 1 ^ b`: shift binds tighter than `^`, so it is (7 << 1) ^ 2 = 14 ^ 2 = 1110 ^ 0010 = 1100 = 12. (Option B corresponds to the wrong grouping 7 << (1 ^ 2) = 56.)
  Output: `7 2 12`.

Based on PYQs: GATE 2019, 2021, 2024, 2026 · last tested 2026 · new question on the same concept, not a reprint
Confidence: Topic 92% | Concept 84% | Question-pattern 75% | Exact-question 20%

Section 15 — Last-Minute Revision Sheet

This sheet is meant to be read once a day in the final two weeks and once more on the morning of the examination. It contains only what GATE CS has repeatedly tested: the concepts, formulas, algorithms, traps and edge cases distilled from the recurrence analyses of the eleven subject chapters (Section 5). Every formula is written in plain text with its assumptions stated; where GATE keys have used a particular convention, the convention is named. Nothing here is a copied question.

Notation used throughout: log means log base 2 unless written ln; ceil(x) and floor(x) are the ceiling and floor; C(n, k) is the binomial coefficient; abs(x) is the absolute value; x^y is x to the power y; a mod b is the remainder.

Top 50 concepts to revise repeatedly

1. **Eigenvalues.** Sum = trace, product = determinant; eigenvalues of A^k, A^-1, A + cI are lambda^k, 1/lambda, lambda + c; triangular matrices show them on the diagonal.
2. **Bayes' theorem.** Posterior = (likelihood × prior) / (total probability); write the denominator as a sum over all sources before dividing.
3. **Linearity of expectation.** E[sum] = sum of E, with no independence needed; indicator variables count matches, fixed points, distinct faces.
4. **Named distributions.** Uniform variance (b − a)^2/12; exponential mean 1/lambda and memorylessness; Poisson mean = variance = lambda t; binomial np and np(1 − p).
5. **Handshake and Euler.** Sum of degrees = 2e; v − e + f = 2 for connected planar graphs; e ≤ 3v − 6, and e ≤ 2v − 4 when bipartite or triangle-free.
6. **Counting relations and functions.** Reflexive-and-symmetric relations 2^(n(n − 1)/2); onto functions by inclusion–exclusion; Bell numbers 1, 1, 2, 5, 15, 52 for equivalence relations.
7. **First-order logic translation.** Universal goes with implication, existential with conjunction; "not every A is B" is exists x (A(x) and not B(x)).
8. **Lagrange's theorem.** Subgroup order divides group order; element order divides group order; prime order implies cyclic; Z_n has phi(n) generators and tau(n) subgroups.
9. **Rank and solutions.** Unique solution iff rank(A) = rank([A | b]) = n; infinitely many iff both ranks equal and less than n; none iff rank(A) < rank([A | b]).
10. **C pointer arithmetic.** p + k scales by sizeof(*p); *p++ increments the pointer; a row pointer int (*q)[n] steps by n elements.
11. **C operator semantics.** a / b truncates toward zero and a % b takes the sign of a; the precedence table; missing break in switch; short-circuit skips side effects.
12. **Recursion tracing.** Table the values from the base upward; print-recurse-print gives a sequence and its reverse; a static accumulator's final value is what every frame reads after unwinding.
13. **BST facts.** Inorder is sorted; successor of a node with a right child is the minimum of that subtree; a search path is valid only if every key stays within the (low, high) bounds set so far.
14. **Heap facts.** 1-based parent floor(i/2), children 2i and 2i + 1; build-heap is Theta(n) with at most 2n comparisons; the k-th smallest of a min-heap lies at depth at most k − 1.
15. **Catalan numbers.** 1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796 for n = 0..10: BSTs on n keys, stack permutations, parenthesisations of n + 1 factors.
16. **Traversal reconstruction.** Preorder-plus-inorder, postorder-plus-inorder and level-order-plus-inorder determine a binary tree; preorder-plus-postorder does not unless the tree is full.
17. **Master theorem.** Compare f(n) with n^(log_b a): smaller by a polynomial factor gives n^(log_b a); equal up to log^k gives one more log; larger (with regularity) gives f(n).

18. **Standard recurrences.** T(n − 1) + n is n^2; T(n − 1) + log n is n log n; 2T(n/2) + n log n is n log^2 n; T(sqrt n) + 1 is log log n; T(n/3) + T(2n/3) + n is n log n.
19. **Loop counts.** j += i inner loops give n log n; i = i × i gives log log n; while (i × i ≤ n) gives sqrt n; doubling outer loops give n.
20. **MST theorems.** Adding a constant or squaring positive weights keeps the MST; the heaviest edge of a cycle is in no MST unless tied; distinct weights give a unique MST but the converse is false.
21. **Shortest-path theorems.** Adding a constant changes shortest paths (fewer-edge paths gain less); multiplying by a positive constant does not; Dijkstra can fail with a negative edge but does not always.
22. **DFS edge types.** Directed graphs have tree, back, forward and cross edges; undirected DFS has only tree and back edges; a connected undirected DFS has exactly e − (v − 1) non-tree edges.
23. **Hashing probes.** Count every slot examined including the empty one where the key lands; linear probing wraps around; expected probes 1 + alpha (chaining, unsuccessful).
24. **Quicksort behaviour.** Sorted input with first-element pivot is the worst case; partition output depends on the stated scheme; quicksort and heapsort are not stable; merge sort is not in place.
25. **Language classification.** A free middle segment makes w x w^R regular; one linear comparison of counts is CFL; two independent comparisons, ww, or a^n b^n c^n are not CFL.
26. **Minimum DFA states.** Independent residue conditions multiply; "ends with w" needs |w| + 1 states; "k-th symbol from the right" needs 2^k; count the dead state when it exists.
27. **Closure properties.** DCFL is closed under complement but not union; CFL under union but not intersection or complement; RE under everything except complement.
28. **Decidability.** Everything about DFAs is decidable; for CFGs only membership, emptiness and finiteness are decidable; for TMs only syntactic and step-bounded questions; Rice's theorem covers the rest.
29. **RE versus recursive.** L and its complement both RE implies recursive; halting is RE not recursive; its complement is not RE; "accepts some string" is RE, "accepts no string" is co-RE, "L(M) is regular" is neither.
30. **FIRST/FOLLOW rules.** epsilon is never in FOLLOW; $ is in FOLLOW of the start symbol; nullable chains push FIRST of later symbols and FOLLOW of the parent through.
31. **LL(1) conditions.** For A → alpha | beta: disjoint FIRST sets, at most one nullable, and if beta is nullable then FIRST(alpha) and FOLLOW(A) are disjoint; no left recursion; no ambiguity.
32. **LR hierarchy.** LR(0) ⊂ SLR(1) ⊂ LALR(1) ⊂ CLR(1); LALR has the same number of states as LR(0)/SLR; merging LR(1) states can create only reduce–reduce conflicts; an ambiguous grammar is not LR(k).
33. **SDT evaluation.** Build the parse tree with the grammar's associativity (E → E op T is left, E → T op E is right), then evaluate bottom-up; embedded actions fire in depth-first left-to-right order.
34. **Liveness.** live-in(B) = use(B) ∪ (live-out(B) − def(B)); a variable used and then defined in the same instruction is live before it; iterate backward.
35. **Scheduling metrics.** Turnaround = completion − arrival; waiting = turnaround − burst; response = first dispatch − arrival; in RR a new arrival enters the queue before the preempted process (GATE convention).
36. **Critical-section triad.** Strict alternation gives ME and bounded waiting but not progress; flag-only gives ME but can deadlock; Peterson's gives all three; TSL gives ME without bounded waiting.
37. **Semaphore arithmetic.** A negative semaphore value counts the blocked processes; wrong order of wait(mutex) and wait(empty) deadlocks the bounded buffer when it is full.
38. **Paging arithmetic.** Entries = 2^(VA bits − offset bits); entries per page-table page = page size / PTE size; levels = ceil(page-number bits / bits per level); the outer table always costs one page.
39. **Effective access time.** TLB: h × (t_TLB + t_mem) + (1 − h) × (t_TLB + (L + 1) × t_mem) for L levels, serial lookup; demand paging: (1 − p) × t_mem + p × t_fault.
40. **Deadlock arithmetic.** Deadlock is impossible iff resources ≥ sum over processes of (need_i − 1) + 1; Banker's checks Request ≤ Need and Request ≤ Available, then runs safety.
41. **Inode arithmetic.** Maximum file size = (d + p + p^2 + p^3) × B with p = B / pointer size; disk accesses with the inode cached: 1 direct, 2 single, 3 double, 4 triple.

Appendix A — Diagram-Based Questions

A.0 Why this appendix exists

FACT. A share of GATE CS questions is built around a figure rather than a table: a logic circuit whose output function must be derived, a finite-automaton state diagram, a weighted graph for a minimum spanning tree, a Gantt chart, an entity-relationship diagram, a B+ tree, a congestion-window plot. The figure is not decoration in those questions; it carries the data, and the question cannot be answered without reading it.

ANALYSIS. The main question bank in Section 5 states such questions in their text-equivalent form, because that form is unambiguous and because GATE itself frequently uses it:

Figure in the paper	Text-equivalent form used in Section 5
Automaton state diagram	Transition table, with the start state and final states named
Weighted graph	Edge list with weights, plus the vertex set
Binary tree or heap	Level-order array, or parent–child listing
Gantt chart	Process table with arrival and burst times, and the schedule as a text timeline
Cache or address layout	Address width, page or block size and field widths given numerically
ER diagram	Entity list with cardinality ratios and participation stated in words
Congestion-window plot	Round-by-round cwnd table

Both forms test the same concept, and a candidate who can only handle one of them is half prepared. Reading a diagram quickly is its own skill: finding the start state, spotting a self-loop, reading a double circle as accepting, tracing a bar in a Gantt chart to the right tick, distinguishing total from partial participation in an ER diagram.

PREDICTION. Expect several figure-based questions in the 2027 paper. They cluster in Digital Logic (circuits and K-maps), Theory of Computation (automata), Algorithms (graphs), Operating Systems (Gantt charts and resource-allocation graphs), Databases (ER diagrams and B+ trees) and Computer Networks (topologies and timelines).

A.1 How to use this appendix

The questions that follow are numbered QA.1 onwards and carry the same header block as the main bank: subject, topic and subtopic; question type and marks; the four confidence scores; the PYQ-lineage tag; and a full worked solution that refers back to the figure.

Work through them after finishing the corresponding subject chapter, and make a point of these habits while you do:

1. **Read every label before reading the question stem.** Edge weights, bit widths, tick values and cardinality marks are the data.
2. **Redraw the figure on your rough sheet.** For automata, copy the transition table out of the diagram; for graphs, write the edge list. Converting to text is usually faster than repeatedly re-reading the picture, and it is the form in which you already practise.
3. **Check what the figure does not say.** A missing transition in a DFA means a dead state; an unlabelled participation line means partial participation; an axis without a tick at the origin may not start at zero.
4. **Watch for figures that are deliberately larger than needed.** GATE often draws a graph with edges that no correct path uses.

Every figure here is drawn to the same conventions: start states carry an incoming arrow from empty space, accepting states are double circles, directed edges carry arrowheads, and all quantities are labelled directly on the drawing.

A.2 Digital Logic, Computer Organization, Programming and Data Structures

Real GATE CS papers routinely print a figure and ask a question that cannot be answered without reading it: a gate-level circuit, a Karnaugh map, a state or timing diagram, a datapath, a tree, a linked list. The main question bank (Section 5) states such items in text form — transition tables, edge lists, array contents — because text travels better through a book. This appendix restores the figures. Each of the nine questions below has its own figure, is self-contained, and is unanswerable from the prose alone: every number you need is inside the picture.

Conventions used throughout this appendix. Logic gates are drawn as labelled rectangles ("AND", "XOR", "NAND", "NOT") rather than as distinctive-shape symbols, because rectangles survive printing at small sizes; the label inside the rectangle is the gate function. In circuit diagrams a solid dot marks a connection between wires, and two wires that cross without a dot are not connected. Trees are drawn with the left child to the left of its parent and the right child to the right. All prediction-confidence figures follow the four-level rubric (Topic ≥ Concept ≥ Question-pattern ≥ Exact-question, with Exact-question capped at 35%) and are model-based analytical confidence, not probabilities.

Format mix in this sub-appendix: 5 MCQ and 4 NAT; 1 one-mark and 8 two-mark items.

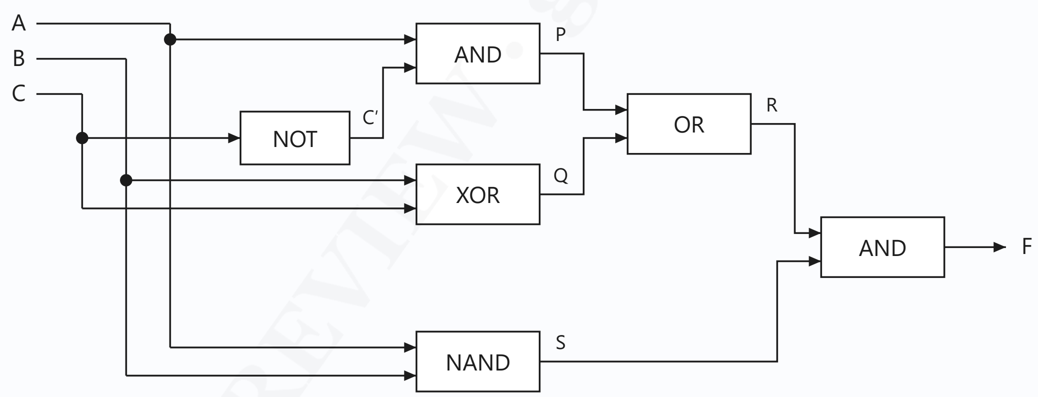
QA.1 NAT · 2 marks Moderate

Subject: Digital Logic | **Topic:** Design of combinational circuits | **Subtopic:** Circuit analysis — output function and truth-table counting
Prediction Confidence: Topic 88% | Concept 78% | Question-pattern 68% | Exact-question 18%
Based on PYQs: GATE 2002, 2007, 2014, 2021 · last tested 2021 · new question on the same concept, not a reprint

QUESTION

The combinational circuit shown below is built from six gates, each drawn as a labelled rectangle. The inputs are A, B and C, and the output is F. The intermediate nets are marked C′, P, Q, R and S. A solid dot marks a connection; wires that cross without a dot are not connected.

The number of distinct input combinations (A, B, C) for which F = 1 is _______.



Gates are labelled rectangles. A solid dot marks a connection; crossings without a dot are not joined.

ANSWER 4 (integer)

Concept Tested: Reading a gate-level schematic, propagating signals through it net by net, and converting the resulting Boolean expression into a truth table; XOR and NAND behaviour; counting satisfying assignments rather than simplifying.
Why This Question Is High Probability: "Analyse this circuit" is the oldest Digital Logic question shape in GATE and it survives because it cannot be answered by recall. Boolean algebra and combinational-circuit design sit in Tier 1 of the Digital Logic list, the 2027 syllabus says "Design of combinational and sequential circuits" explicitly, and the 2026 shift-2 analysis specifically noted that Boolean-algebra weight was higher than expected. The NAT phrasing ("how many input combinations give output 1") is the modern renewal of the old MCQ phrasing ("the output F is").
Historical Evidence: 2002 (output of a gate network in terms of the inputs); 2007 (circuit built from NAND/XOR, identify the

function realised); 2014 (the function implemented by a given multiplexer/gate network); 2021 (MSQ on the properties of a given Boolean network). The XOR-plus-NAND combination is a recurring setter favourite because the NAND branch silently kills the A = B = 1 rows.

SOLUTION

Read the nets off the figure:

$C' = \text{NOT } C$

$P = \text{AND}(A, C') = A{\cdot}C'$

$Q = \text{XOR}(B, C) = B \oplus C$

$R = \text{OR}(P, Q) = A{\cdot}C' + (B \oplus C)$

$S = \text{NAND}(A, B) = (A{\cdot}B)'$

$F = \text{AND}(R, S) = [A{\cdot}C' + (B \oplus C)] \cdot (A{\cdot}B)'$

Truth table (all eight rows):

A	B	C	C′	P = A·C′	Q = B⊕C	R = P+Q	S = (AB)′	F = R·S
0	0	0	1	0	0	0	1	0
0	0	1	0	0	1	1	1	1
0	1	0	1	0	1	1	1	1
0	1	1	0	0	0	0	1	0
1	0	0	1	1	0	1	1	1
1	0	1	0	0	1	1	1	1
1	1	0	1	1	1	1	0	0
1	1	1	0	0	0	0	0	0

F = 1 in exactly four rows: (0,0,1), (0,1,0), (1,0,0) and (1,0,1). Answer: 4.

As a check, minimise: $F = \Sigma m(1, 2, 4, 5) = A'(B \oplus C) + A{\cdot}B'$. The second term $A{\cdot}B'$ covers m4 and m5, the first covers m1 and m2, and the NAND branch $S = (AB)'$ is what forces m6 and m7 to 0 even though R = 1 for m6.

Common errors: forgetting that S = 0 for both A = B = 1 rows (gives 5); reading the rectangle labelled NAND as AND ($F = A{\cdot}B{\cdot}R$, true only for m6, gives 1); or treating XOR as OR, which makes Q = 1 for m3 and m7 and gives an answer of 5.

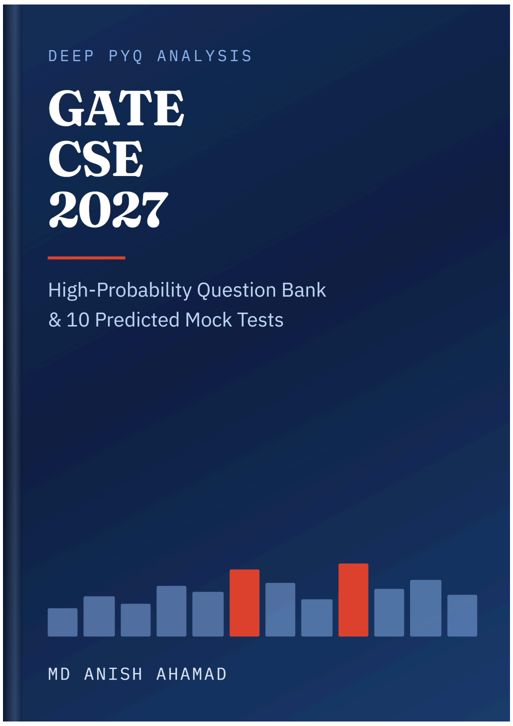
QA.2 NAT · 2 marks Moderate

Subject: Digital Logic | **Topic:** Boolean minimization | **Subtopic:** Karnaugh map with don't-cares — prime implicants and minimal literal count
Prediction Confidence: Topic 90% | Concept 82% | Question-pattern 73% | Exact-question 20%
Based on PYQs: GATE 2008, 2014, 2017, 2021 · last tested 2021 · new question on the same concept, not a reprint

QUESTION

The Karnaugh map of a four-variable Boolean function F(A, B, C, D) is shown below. The rows are labelled with AB and the columns with CD, both in reflected-Gray order. The small number at the top left of each cell is the minterm index; X denotes a don't-care.

The minimum number of literals in a minimal sum-of-products expression for F is _______.



END OF FREE PREVIEW

Get the full 1,016-page book

- 27 years of GATE CS papers analysed topic by topic
- Top 100 concepts ranked, with the reason for each rank
- 1,120 questions with worked solutions
- 10 full mock tests in the 2027 pattern, with keys
- 20-week study plan and last-minute revision sheet

₹250 ~~₹300~~

Buy the book — ₹250

See the book page

WhatsApp: +91 70339 35778 · **Web:** gateguide.co.in · **Email:** mdanisahamad2003@gmail.com

Free GATE CSE 2027 guides (syllabus changes, subject-wise important topics, formula sheet, study plans): gateguide.co.in/articles