IAE ACADEMY INSTITUTE FOR ENGINEERS

GATE CS · 2025 · SET 1

IAE Academy free study material · Updated 27 October 2026

GATE CS 2025 Set 1 Solved Paper

INSIDE

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

Question paper and official answer key: IIT Roorkee

Online, with the questions linked to their topic notes: **iaeacademy.in/study/papers/cs-2025-1**

FREE COUNSELLING

Preparing for GATE CS? Talk to IAE Academy.

Or call **+91 84204 36524**

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

Chat on WhatsApp

Free. Not for sale. Pass it on.

About this paper

65 questions, 100 marks. Question types: 30 MCQ · 13 MSQ · 22 NAT.

Official answer key: IIT Roorkee · https://gate2025.iitr.ac.in/doc/2025/2025_Key/CS1_Keys.pdf

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

This paper online, with the questions linked to their topic notes:
iaeacademy.in/study/papers/cs-2025-1

Answer key

Q	TYPE	MARKS	ANSWER
Q.1	MCQ	1	**A**
Q.2	MCQ	1	**B**
Q.3	MCQ	1	**C**
Q.4	MCQ	1	**A**
Q.5	MCQ	1	**C**
Q.6	MCQ	2	**C**
Q.7	MCQ	2	**A**
Q.8	MCQ	2	**A**
Q.9	MCQ	2	**A**
Q.10	MCQ	2	**B**
Q.11	MCQ	1	**A**
Q.12	MCQ	1	**C**
Q.13	MCQ	1	**B**
Q.14	MCQ	1	**A**
Q.15	MCQ	1	**C**
Q.16	MCQ	1	**B**
Q.17	MCQ	1	**D**
Q.18	MCQ	1	**B**
Q.19	MCQ	1	**A**
Q.20	MCQ	1	**A**
Q.21	MSQ	1	**B, D**
Q.22	MSQ	1	**B, D**
Q.23	MSQ	1	**A, D**
Q.24	MSQ	1	**B, D**
Q.25	MSQ	1	**B, D**
Q.26	MSQ	1	**A, B**
Q.27	MSQ	1	**A, B, C**
Q.28	MSQ	1	**D**
Q.29	NAT	1	**21**
Q.30	NAT	1	**195**
Q.31	NAT	1	**-2.1 to -1.9**
Q.32	NAT	1	**0.49 to 0.51**
Q.33	NAT	1	**435**
Q.34	NAT	1	**25**
Q.35	NAT	1	**5**
Q.36	MCQ	2	**A**
Q.37	MCQ	2	**B**
Q.38	MCQ	2	**A**
Q.39	MCQ	2	**A**
Q.40	MCQ	2	**A**
Q.41	MCQ	2	**D**
Q.42	MCQ	2	**A**
Q.43	MCQ	2	**C**
Q.44	MCQ	2	**C**
Q.45	MCQ	2	**C**
Q.46	MSQ	2	**A, D**
Q.47	MSQ	2	**B, C**
Q.48	MSQ	2	**B, D**
Q.49	MSQ	2	**B**
Q.50	MSQ	2	**C**
Q.51	NAT	2	**65468**
Q.52	NAT	2	**6**
Q.53	NAT	2	**11.83 to 11.87**
Q.54	NAT	2	**6**
Q.55	NAT	2	**26**
Q.56	NAT	2	**0.949 to 0.952**
Q.57	NAT	2	**7**
Q.58	NAT	2	**0.300 to 0.302**
Q.59	NAT	2	**5**
Q.60	NAT	2	**7 to 8**
Q.61	NAT	2	**5**
Q.62	NAT	2	**5**
Q.63	NAT	2	**46**
Q.64	NAT	2	**5**
Q.65	NAT	2	**10 to 11**

General Aptitude Q.1 to Q.10

Q.1 MCQ · 1 MARK

General Aptitude › Grammar & vocabulary

Ravi had _____ younger brother who taught at _____ university. He was widely regarded as _____ honorable man.

Select the option with the correct sequence of articles to fill in the blanks.

(A) a; a; an

(B) the; an; a

(C) a; an; a

(D) an; an; a

Official answer: **A**

SOLUTION

1. Rule: use "a" before a word that begins with a consonant SOUND and "an" before a word that begins with a vowel SOUND. The sound decides it, not the spelling of the first letter.
2. Blank 1: "_____ younger brother". The article goes with "younger", which begins with the consonant sound /y/. The brother is not identified, so the indefinite article is needed: "a younger brother". "An" is wrong because "younger" does not begin with a vowel sound.
3. Blank 2: "_____ university". "University" starts with a vowel letter but is pronounced /yoo-/, which begins with the consonant sound /y/. So "a university".
4. Blank 3: "_____ honorable man". The h in "honorable" is silent, so the word is pronounced /on-/ and begins with a vowel sound. So "an honorable man".
5. Sequence: a; a; an. This is option (A). Option (B) starts with "the", but the brother is not a particular one already known, so "the" is wrong, and it also puts "an" before "university". Options (C) and (D) put "an" before "university" and "a" before "honorable", so both are wrong.

KEY IDEA

Choose a or an by the first SOUND of the next word, not its first letter. "University" starts with the consonant sound /y/, so it takes "a". "Honorable" has a silent h and starts with a vowel sound, so it takes "an".

WATCH OUT FOR

Going by spelling: "an university" because u is a vowel letter, or "a honorable" because h is a consonant letter. Both are wrong because the sound of the word decides the article.

Q.2 MCQ · 1 MARK

General Aptitude › Grammar & vocabulary

The CEO’s decision to downsize the workforce was considered myopic because it sacrificed long-term stability to accommodate short-term gains.

Select the most appropriate option that can replace the word “myopic” without changing the meaning of the sentence.

(A) visionary

(B) shortsighted

(C) progressive

(D) innovative

Official answer: **B**

SOLUTION

1. The word to replace is "myopic". Literally it means nearsighted; figuratively it means lacking foresight, thinking only of the present.

2. The sentence says the decision "sacrificed long-term stability to accommodate short-term gains". The decision therefore focused on the short term and ignored the long term.
3. The tone is critical ("was considered myopic because..."). The replacement must be a negative word about lack of foresight, and it must keep the meaning of the sentence.
4. Option (A) visionary means having foresight. It is the opposite of myopic. Reject.
5. Options (C) progressive and (D) innovative both describe positive, forward-looking change. Neither means short-term thinking, and both would make the sentence contradict itself. Reject.
6. Option (B) shortsighted means lacking concern for the long term. It fits the sentence exactly. Answer: (B).

KEY IDEA
Use the "because" clause as the context clue. It says the decision gave up long-term stability for short-term gains, so the missing word must mean lacking foresight. That is "shortsighted".

WATCH OUT FOR
Choosing a word that sounds like a business term, such as "progressive" or "innovative". Another mistake is picking "visionary" because it relates to sight and vision, when it means the opposite of myopic.

Q.3 MCQ · 1 MARK

General Aptitude › Numerical

The average marks obtained by a class in an examination were calculated as 30.8. However, while checking the marks entered, the teacher found that the marks of one student were entered incorrectly as 24 instead of 42. After correcting the marks, the average becomes 31.4. How many students does the class have?

(A) 25
(B) 28
(C) 30
(D) 32

Official answer: **C**

SOLUTION

1. Let the class have n students. The average is $\frac{\text{total}}{n}$, so a change in the total changes the average by (change in total)$/n$.
2. The marks were entered as 24 but should be 42, so the total rises by $42 - 24 = 18$.
3. The average rises from 30.8 to 31.4, an increase of $31.4 - 30.8 = 0.6$.
4. Equate the two expressions for the rise in the average: $\frac{18}{n} = 0.6$.
5. Solve: $n = \frac{18}{0.6} = 30$.
6. Check: old total $= 30.8 \times 30 = 924$; new total $= 924 + 18 = 942$; $\frac{942}{30} = 31.4$. This matches the given corrected average, so the answer is (C).

KEY IDEA
Correcting one entry changes the total by the difference between the right and wrong marks. Dividing that difference by the number of students gives the change in the average, so $n = \frac{\text{change in total}}{\text{change in average}}$.

WATCH OUT FOR
Using the old and new averages directly (for example 30.8 and 31.4) as if they were totals, or reading the correction as $24 - 42 = -18$ and getting a negative number of students. The marks go up by 18, so the average goes up as well.

Q.4 MCQ · 1 MARK

General Aptitude › Reasoning

Consider the relationships among P, Q, R, S, and T:
- P is the brother of Q.
- S is the daughter of Q.
- T is the sister of S.
- R is the mother of Q.

The following statements are made based on the relationships given above.
(1) R is the grandmother of S.
(2) P is the uncle of S and T.
(3) R has only one son.
(4) Q has only one daughter.
Which one of the following options is correct?

(A) Both (1) and (2) are true.

(B) Both (1) and (3) are true.

(C) Only (3) is true.

(D) Only (4) is true.

Official answer: **A**

SOLUTION

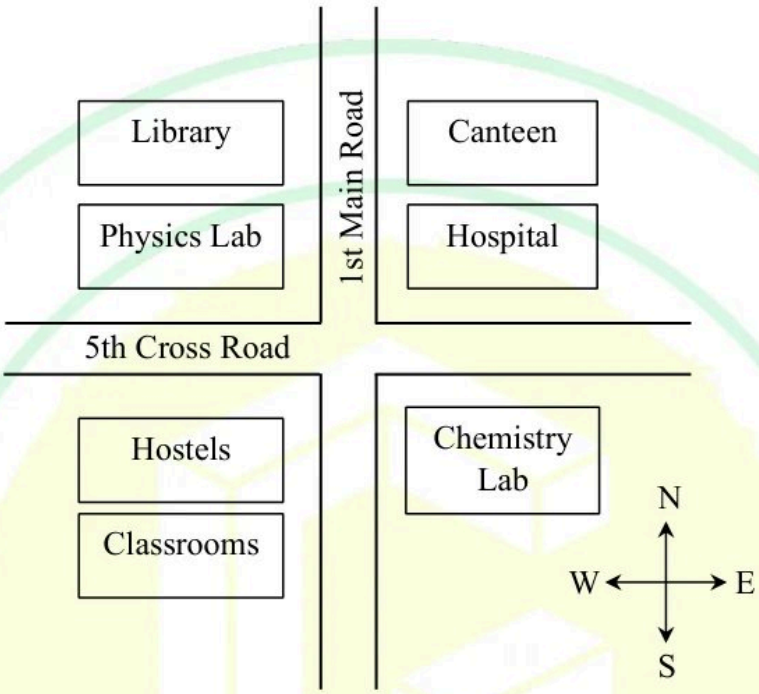
1. Read the given relationships with the usual convention that sisters and brothers are full siblings (same parents). P is the brother of Q and R is the mother of Q, so R is the mother of both P and Q.
2. S is the daughter of Q, and T is the sister of S, so T is also a daughter of Q. Hence Q has at least two daughters, S and T.
3. Statement (1): R is the mother of Q and Q is a parent of S, so R is the grandmother of S. True.
4. Statement (2): P is the brother of Q, and S and T are Q's children, so P is the uncle of both S and T. This holds whether Q is the mother or the father. True.
5. Statement (3): P is one son of R, but Q's gender is not given. If Q is a son, R has at least two sons. So "R has only one son" cannot be concluded. Not necessarily true.
6. Statement (4): Q has at least two daughters, S and T, so "Q has only one daughter" is false.
7. Only (1) and (2) are true, which is option (A).

KEY IDEA
Build the family tree from the given links only. A statement is true only if it follows in every case consistent with the facts. Where a fact such as Q's gender is not given, the statement that depends on it cannot be concluded.

WATCH OUT FOR
Assuming Q is a woman (so P would be R's only son and statement (3) would hold), or assuming Q is a man. The gender of Q is never stated, so statement (3) cannot be confirmed. Statement (2) holds either way.

Q.5 MCQ · 1 MARK

General Aptitude › Data interpretation

Q.5	According to the map shown in the figure, which one of the following statements is correct? Note: The figure shown is representative.
	
(A)	The library is located to the northwest of the canteen.
(B)	The hospital is located to the east of the chemistry lab.
(C)	The chemistry lab is to the southeast of physics lab.
(D)	The classrooms and canteen are next to each other.

Official answer: **C**

SOLUTION

1. Take north at the top of the map and east to the right, as the compass shows. 1st Main Road runs north to south and separates west from east. 5th Cross Road runs east to west and separates north from south.
2. The four blocks are as follows. North-west: Library above Physics Lab. North-east: Canteen above Hospital. South-west: Hostels above Classrooms. South-east: Chemistry Lab.

3. (A) The Library and the Canteen are at the same height on opposite sides of 1st Main Road. The Library is therefore directly west of the Canteen, not to its north-west. False.
4. (B) The Hospital lies north of 5th Cross Road and the Chemistry Lab lies south of it. Both are east of 1st Main Road and are in line vertically. The Hospital is directly north of the Chemistry Lab, not to its east. False.
5. (C) The Physics Lab is west of 1st Main Road and north of 5th Cross Road. The Chemistry Lab is east of 1st Main Road and south of 5th Cross Road. Going from the Physics Lab to the Chemistry Lab means moving east and south, so the Chemistry Lab is to the south-east of the Physics Lab. True.
6. (D) The Classrooms are in the south-west block and the Canteen is in the north-east block. They are on opposite sides of both roads, so they are not next to each other. False.
7. Only option (C) is correct.

KEY IDEA
Place each building in a block using the two roads as axes (1st Main Road splits west from east, 5th Cross Road splits north from south). Then read each direction from the compass. A compound direction such as south-east needs a shift in both axes.

WATCH OUT FOR
Using a compound direction for places that differ in only one axis. The library and canteen are level, so the library is west of the canteen, not north-west. The hospital and chemistry lab are in the same column, so the hospital is north of the lab, not east.

Q.6 MCQ · 2 MARKS

General Aptitude › Reading

"I put the brown paper in my pocket along with the chalks, and possibly other things. I suppose every one must have reflected how primeval and how poetical are the things that one carries in one's pocket: the pocket-knife, for instance the type of all human tools, the infant of the sword. Once I planned to write a book of poems entirely about the things in my pocket. But I found it would be too long: and the age of the great epics is past."

(From G.K. Chesterton's "A Piece of Chalk")

Based only on the information provided in the above passage, which one of the following statements is true?

(A) The author of the passage carries a mirror in his pocket to reflect upon things.

(B) The author of the passage had decided to write a poem on epics.

(C) The pocket-knife is described as the infant of the sword.

(D) Epics are described as too inconvenient to write.

Official answer: **C**

SOLUTION

1. A statement is true only if the passage says it directly. Do not bring in outside knowledge or loose inference.
2. Option (A): the passage says "every one must have reflected" on the things in one's pocket. Here "reflected" means thought about. The passage never mentions a mirror, so (A) is false.
3. Option (B): the author "planned to write a book of poems entirely about the things in my pocket". He did not decide to write a poem on epics, so (B) is false.
4. Option (C): the passage says "the pocket-knife, for instance the type of all human tools, the infant of the sword". The pocket-knife is therefore described as the infant of the sword, so (C) is true.
5. Option (D): the book of pocket-poems "would be too long", and "the age of the great epics is past". Epics are not described as too inconvenient to write, so (D) is false.
6. Only (C) is stated in the passage.

KEY IDEA

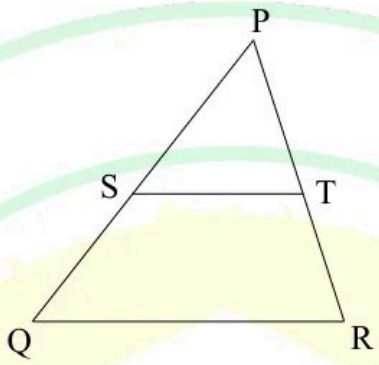
In "which statement is true based only on the passage" questions, pick the option the text states directly, and reject any option that rests on a pun, an assumption or outside knowledge.

WATCH OUT FOR

Reading "reflected" as a hint about a mirror in (A), or treating "the book would be too long" and "the age of the great epics is past" as a statement that epics are inconvenient to write (D). The passage says neither.

Q.7 MCQ · 2 MARKS

General Aptitude › Geometry

Q.7	In the diagram, the lines QR and ST are parallel to each other. The shortest distance between these two lines is half the shortest distance between the point P and line QR. What is the ratio of the area of the triangle PST to the area of the trapezium SQRT? Note: The figure shown is representative.
	
(A)	$\frac{1}{3}$
(B)	$\frac{1}{4}$
(C)	$\frac{2}{5}$
(D)	$\frac{1}{2}$

Official answer: **A**

SOLUTION

1. Let the shortest distance from P to line QR be h. The shortest distance between the parallel lines ST and QR is given as $\frac{h}{2}$.
2. P lies on the opposite side of ST from QR, so the distance from P to line ST is $h - \frac{h}{2} = \frac{h}{2}$.

3. Since $ST \parallel QR$, the corresponding angles at S and Q and at T and R are equal, and the angle at P is common. So $\triangle PST \sim \triangle PQR$ (AA similarity).
4. In similar triangles the ratio of corresponding sides equals the ratio of corresponding heights from P: $\frac{ST}{QR} = \frac{h/2}{h} = \frac{1}{2}$.
5. Areas of similar triangles are in the ratio of the squares of corresponding sides: $\frac{[PST]}{[PQR]} = \left(\frac{1}{2}\right)^2 = \frac{1}{4}$.
6. The trapezium is the rest of the big triangle: $[SQRT] = [PQR] - [PST] = \frac{3}{4}[PQR]$.
7. Required ratio: $\frac{[PST]}{[SQRT]} = \frac{1/4}{3/4} = \frac{1}{3}$.

KEY IDEA

A line parallel to the base cuts off a triangle similar to the whole. Its height from the apex is the ratio of similarity, and areas go as the square of that ratio. Compare the small triangle with the leftover trapezium, not with the whole triangle.

WATCH OUT FOR

Reporting $\frac{[PST]}{[PQR]} = \frac{1}{4}$ (option B) instead of dividing by the trapezium's area $\frac{3}{4}[PQR]$. Another slip is using the ratio of sides, $\frac{1}{2}$, as the area ratio (option D).

Q.8 MCQ · 2 MARKS

General Aptitude › Statistics & probability

A fair six-faced dice, with the faces labelled ‘1’, ‘2’, ‘3’, ‘4’, ‘5’, and ‘6’, is rolled thrice. What is the probability of rolling ‘6’ exactly once?

(A) $\frac{75}{216}$
(B) $\frac{1}{6}$
(C) $\frac{1}{18}$
(D) $\frac{25}{216}$

Official answer: **A**

SOLUTION

1. Each roll is independent. Let a roll showing '6' be a success, with $p = \frac{1}{6}$, and any other face a failure, with $q = \frac{5}{6}$.
2. Exactly one '6' in three rolls is a binomial event: $P(X = 1) = \binom{3}{1} p\, q^2$.
3. The '6' can appear on the first, second or third roll, so $\binom{3}{1} = 3$ arrangements.
4. $P(X = 1) = 3 \times \frac{1}{6} \times \left(\frac{5}{6}\right)^2 = 3 \times \frac{1}{6} \times \frac{25}{36} = \frac{75}{216}$.
5. Check by counting: favourable outcomes are $3 \times 1 \times 5 \times 5 = 75$, out of $6^3 = 216$ equally likely outcomes. The probability is $\frac{75}{216}$, which is option (A).

KEY IDEA

Repeated independent trials with two outcomes follow a binomial distribution. For exactly k successes in n trials, $P = \binom{n}{k} p^k q^{n-k}$.

WATCH OUT FOR

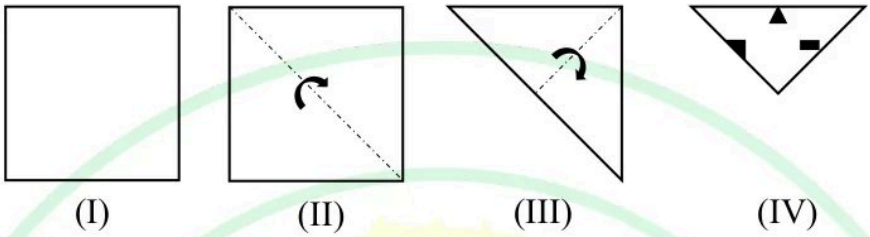
Leaving out the factor $\binom{3}{1} = 3$ for the position of the '6' gives $\frac{25}{216}$, which is option (D). Using only $\frac{1}{6}$ ignores the other two rolls.

Q.9 MCQ · 2 MARKS

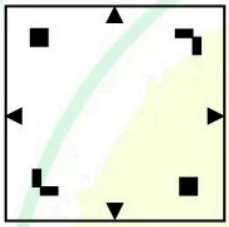
General Aptitude › Spatial

Q.9 A square paper, shown in figure (I), is folded along the dotted lines as shown in the figures (II) and (III). Then a few cuts are made as shown in figure (IV). Which one of the following patterns will be obtained when the paper is unfolded?

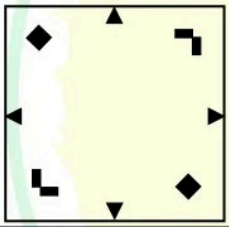
Note: The figures shown are representative.



(A)



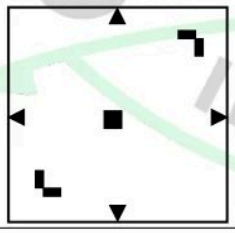
(B)



(C)



(D)



Official answer: **A**

SOLUTION

1. **Track the folds.** Call the square's corners TL, TR, BR, BL and its centre O. The first fold is along the diagonal TL-BR, which gives the triangle TL-TR-BR of figure (III). The second fold is along the line from TR to O, the midpoint of the long edge. Folding the part near BR back over the part near TL gives the triangle of figure (IV): long edge TL-TR (the top side of the square), right-angle vertex O pointing down. The paper now has 4 layers, and both creases lie along the diagonals of the square.

2. **Locate the cuts in figure (IV).** The left slanting edge, from TL to O, is the first crease, which is half of the diagonal TL-BR. The right slanting edge, from TR to O, is the second crease, which is half of the diagonal TR-BL. The triangle cut is on the long edge, the bar cut is on the right (TR-O) crease, and the square-like notch is on the left (TL-O) crease.
3. **Triangle at the middle of the long edge.** This cut is at the midpoint of the top side, pointing outward from the paper. The four layers of this edge are the four sides of the square, because the two diagonals are the mirror lines that carry one side onto the next. Unfolding therefore puts the same cut at the midpoint of every side, each pointing outward: up at the top, left at the left, right at the right, down at the bottom. This rules out C, whose top and bottom triangles point inward and which has no right triangle. It also rules out D, which has no left triangle and has a centre cut that nothing in figure (IV) produces.
4. **Bar on the right crease (diagonal TR-BL).** The bar lies horizontally with one end on the crease. Unfolding along the crease adds its mirror image across the diagonal TR-BL, which is a vertical bar meeting the first at the crease. Together they form an L-shaped cut with its corner on the diagonal, at about the middle of the half-diagonal from O to TR. The other fold repeats this across the diagonal TL-BR, so the same L appears at BL. L-shaped cuts therefore appear at the top-right and bottom-left corners, as in all four options.
5. **Square-like notch on the left crease (diagonal TL-BR).** The notch is half of an upright square. Its flat slanted side lies on the crease, so unfolding across the diagonal TL-BR completes the square, with its sides parallel to the sides of the paper. The other fold repeats it on the opposite end of that diagonal. Upright squares therefore appear at the top-left and bottom-right corners. This rules out B, whose cuts there are tilted diamonds.
6. **Conclusion.** Only A has outward triangles at all four midpoints, upright squares at top-left and bottom-right, and L-shaped cuts at top-right and bottom-left. The answer is (A).

KEY IDEA

Each fold line is a mirror line, so every cut is reflected across the creases when the paper is unfolded. Both creases here are the square's diagonals, so each cut is repeated in a symmetric pattern around the square. A cut on the open edge appears on every side. A cut on a crease joins with its mirror image to make a full shape.

WATCH OUT FOR

Mixing up which slanting edge of the folded triangle is which crease, or treating a half-cut on a crease as a full cut. A bar with one end on a crease joins its mirror image to make an L, not a square. Also, a cut on an outer edge must be repeated on all four sides, pointing the same way (outward) each time.

Q.10 MCQ · 2 MARKS

General Aptitude › Numerical

A shop has 4 distinct flavors of ice-cream. One can purchase any number of scoops of any flavor. The order in which the scoops are purchased is inconsequential. If one wants to purchase 3 scoops of ice-cream, in how many ways can one make that purchase?

(A) 4

(B) 20

(C) 24

(D) 48

Official answer: **B**

SOLUTION

1. Order does not matter and a flavor may be chosen more than once, so this is a selection with repetition: choose 3 scoops from 4 flavors.
2. Let $x_1, x_2, x_3, x_4 \geq 0$ be the number of scoops of each flavor. Then $x_1 + x_2 + x_3 + x_4 = 3$.

3. By stars and bars, the number of non-negative integer solutions of $x_1 + \cdots + x_k = n$ is $\binom{n+k-1}{k-1}$.
4. Here $n = 3$ and $k = 4$, so the count is $\binom{3+4-1}{4-1} = \binom{6}{3}$.
5. $\binom{6}{3} = \dfrac{6 \cdot 5 \cdot 4}{3 \cdot 2 \cdot 1} = 20$.
6. Check by cases. All three scoops of the same flavor: 4 ways. Two scoops of one flavor and one of a different flavor: $4 \times 3 = 12$ ways. Three different flavors: $\binom{4}{3} = 4$ ways. Total $4 + 12 + 4 = 20$, which is option (B).

KEY IDEA

When order does not matter and repeats are allowed, count the non-negative integer solutions of $x_1 + \cdots + x_k = n$. Stars and bars gives $\binom{n+k-1}{k-1}$.

WATCH OUT FOR

Using $4^3 = 64$ or $4 \times 3 \times 2 = 24$ counts ordered purchases. Using $\binom{4}{3} = 4$ forbids repeating a flavor. Neither fits a purchase where order is ignored and repeats are allowed.

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

Q.11 MCQ · 1 MARK

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

Suppose a program is running on a non-pipelined single processor computer system. The computer is connected to an external device that can interrupt the processor asynchronously. The processor needs to execute the interrupt service routine (ISR) to serve this interrupt. The following steps (not necessarily in order) are taken by the processor when the interrupt arrives:

(i) The processor saves the content of the program counter.
(ii) The program counter is loaded with the start address of the ISR.
(iii) The processor finishes the present instruction.

Which ONE of the following is the CORRECT sequence of steps?

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

Official answer: **A**

SOLUTION

1. Interrupts in a non-pipelined processor are recognized only at instruction boundaries, so the processor must first complete the instruction currently executing: step (iii) happens first.
2. Before the program counter can be reused, the return address it holds must be preserved so execution can resume later: the processor saves the program counter, step (i), immediately after finishing the instruction.
3. Only after the PC is safely saved can it be overwritten: it is loaded with the ISR's start address, step (ii), transferring control to the interrupt service routine.
4. This gives the unique valid order (iii) → (i) → (ii), which is option A.

KEY IDEA
Interrupt handling in a non-pipelined processor follows a fixed order dictated by hardware necessity: interrupts are checked and honored only at instruction boundaries, and the return address (PC) must be preserved before it is overwritten with the ISR's address.

WATCH OUT FOR
Assuming the PC is loaded with the ISR address before or without first saving it, or forgetting that a non-pipelined processor only recognizes interrupts at instruction boundaries (so the current instruction must finish first).

Q.12 MCQ · 1 MARK

Compiler Design › Lexical analysis

Which ONE of the following statements is FALSE regarding the symbol table?

(A) Symbol table is responsible for keeping track of the scope of variables.
(B) Symbol table can be implemented using a binary search tree.
(C) Symbol table is not required after the parsing phase.
(D) Symbol table is created during the lexical analysis phase.

Official answer: **C**

SOLUTION

1. The question asks for the FALSE statement, so check each option against the role of the symbol table.

2. (A) True: the symbol table records each name with its scope and attributes, so scope resolution depends on it.
3. (B) True: a symbol table is a dictionary, and a hash table, a balanced binary search tree or a linked list can all implement it.
4. (D) True: the lexical analyser enters identifiers into the table as it finds them, so the table starts to be built during lexical analysis.
5. (C) False: after parsing, semantic analysis needs the table for type checking and declaration checks, and code generation needs it for storage addresses and types. The table is used until the end of compilation.
6. So the false statement is (C).

KEY IDEA

The symbol table is built early (the lexer can enter identifiers) and is consulted through semantic analysis, optimisation and code generation. It is never discarded right after parsing.

WATCH OUT FOR

Treating the symbol table as a parsing-only structure and so picking (C) as true, or doubting (D) because some compilers fill the table in the parser instead of the lexer. Here (D) is still the accepted true statement and (C) is the only clearly false one.

Q.13 MCQ · 1 MARK

Compiler Design › Optimisation

Which ONE of the following techniques used in compiler code optimization uses live variable analysis?

(A) Run-time function call management

(B) Register assignment to variables

(C) Strength reduction

(D) Constant folding

Official answer: **B**

SOLUTION

1. A variable is live at a point if its current value may still be used later before it is redefined. Live variable analysis is a backward data-flow analysis that finds these points.
2. Two variables can share a register only if they are never live at the same time, so the live ranges decide which variables may be put in the same register.
3. Register assignment (allocation) builds an interference graph from the live ranges and colours it. It therefore uses live variable analysis.
4. Strength reduction replaces an expensive operation with a cheaper one, for example $2 \times i$ with $i + i$. It depends on the form of the expression, not on liveness.
5. Constant folding evaluates constant expressions at compile time. Any data-flow support it needs is constant propagation (reaching definitions), not liveness.
6. Run-time function call management concerns activation records and the stack. It is not a data-flow optimization.
7. Only option (B), register assignment to variables, uses live variable analysis.

KEY IDEA
Live variable analysis tells the compiler when a variable's value is still needed. Variables that are never live together can share a register, so register allocation depends on it.

WATCH OUT FOR
Picking constant folding or strength reduction because they are "optimizations that analyse expressions". They use the form of the expression or constant propagation, not liveness. The other classic use of liveness is dead-code elimination.

Q.14 MCQ · 1 MARK

Operating Systems › Memory management

Consider a demand paging memory management system with 32-bit logical address, 20-bit physical address, and page size of 2048 bytes. Assuming that the memory is byte addressable, what is the maximum number of entries in the page table?

(A) 2^{21}
(B) 2^{20}
(C) 2^{22}
(D) 2^{24}

Official answer: **A**

SOLUTION

1. A page table has one entry for each page of the logical (virtual) address space, so the number of entries is the number of pages, $\dfrac{\text{logical address space}}{\text{page size}}$.
2. The logical address is 32 bits and memory is byte addressable, so the logical address space is 2^{32} bytes.
3. The page size is $2048 = 2^{11}$ bytes, so the offset takes 11 bits.
4. The page number therefore takes $32 - 11 = 21$ bits.
5. Number of page table entries $= \dfrac{2^{32}}{2^{11}} = 2^{21}$.
6. The 20-bit physical address only fixes the number of frames, $\dfrac{2^{20}}{2^{11}} = 2^{9}$. It does not change the number of entries, because the table is indexed by page number, not by frame number. Demand paging only means some entries may be marked invalid, and the table still has one entry per page.
7. Hence the answer is option (A), 2^{21}.

KEY IDEA
The page table is indexed by virtual page number, so its size is $2^{\text{logical address bits} - \text{offset bits}}$. The physical address width does not affect it.

WATCH OUT FOR
Using the 20-bit physical address and computing $2^{20}/2^{11} = 2^{9}$ (the number of frames). Another slip is picking 2^{20} because the physical address has 20 bits.

Q.15 MCQ · 1 MARK

Database Management and Warehousing › Transactions

A schedule of three database transactions T_1, T_2, and T_3 is shown. $R_i(A)$ and $W_i(A)$ denote read and write of data item A by transaction T_i, $i = 1, 2, 3$. The transaction T_1 aborts at the end. Which other transaction(s) will be required to be rolled back?

$$R_1(X)\ W_1(Y)\ R_2(X)\ R_2(Y)\ R_3(Y)\ ABORT(T_1)$$

(A) Only T_2
(B) Only T_3
(C) Both T_2 and T_3

(D) Neither T_2 nor T_3

Official answer: **C**

SOLUTION

1. Write the schedule as one column per transaction, in time order:

T_1	T_2	T_3
$R_1(X)$		
$W_1(Y)$		
	$R_2(X)$	
	$R_2(Y)$	
		$R_3(Y)$
Abort		

2. When a transaction aborts, every write it made is undone. Any other transaction that has read a value written by the aborted transaction (a dirty read) has used data that never officially existed, so it must also be rolled back.
3. T_1 writes only Y. It never writes X, so $R_2(X)$ reads a value T_1 did not touch and causes no dependence.
4. $R_2(Y)$ comes after $W_1(Y)$ and before T_1 aborts, so T_2 read the uncommitted value of Y written by T_1. T_2 must be rolled back.
5. $R_3(Y)$ also comes after $W_1(Y)$ and before the abort, so T_3 also read the uncommitted value of Y. T_3 must be rolled back too.
6. Both rollbacks are cascaded from T_1's abort, because each of T_2 and T_3 read a value written by T_1 before T_1 finished. This schedule is therefore not cascadeless. (Neither T_2 nor T_3 has committed in the schedule shown, so they can still be rolled back, and no committed work is lost.)
7. Answer: both T_2 and T_3, option (C).

KEY IDEA

If a transaction aborts, every transaction that read data the aborted one wrote (a dirty read) must also be rolled back. This is a cascading rollback.

WATCH OUT FOR

Counting $R_2(X)$ as a dependence, or rolling back only the first transaction that read Y. A read of an item the aborted transaction never wrote is harmless, but every read of Y after $W_1(Y)$ is dirty.

Q.16 MCQ · 1 MARK

Computer Networks › Layering & switching

Q.16 Identify the ONE CORRECT matching between the OSI layers and their corresponding functionalities as shown.

OSI Layers	Functionalities
(a) Network layer	(I) Packet routing
(b) Transport layer	(II) Framing and error handling
(c) Datalink layer	(III) Host to host communication

(A) (a)-(I), (b)-(II), (c)-(III)

(B) (a)-(I), (b)-(III), (c)-(II)

(C) (a)-(II), (b)-(I), (c)-(III)

(D) (a)-(III), (b)-(II), (c)-(I)

Official answer: **B**

SOLUTION

1. The network layer is responsible for moving packets from the source to the destination across networks, which needs routing. So (a) Network layer matches (I) Packet routing.
2. The transport layer gives end-to-end delivery between the communicating hosts, with port numbers identifying the processes at each end. Of the three functions listed, this is closest to (III) Host to host communication. So (b) matches (III).
3. The data link layer groups bits into frames and detects or handles errors on a single link. So (c) Datalink layer matches (II) Framing and error handling.
4. The matching is (a)-(I), (b)-(III), (c)-(II), which is option B. Option A pairs (b) with (II) and (c) with (III), which is wrong. Option C pairs (a) with (II) and (b) with (I), which is wrong. Option D pairs (a) with (III) and (c) with (I), which is wrong.

KEY IDEA

Match each OSI layer to its core job. The network layer routes packets, the transport layer provides end-to-end (host to host) communication, and the data link layer does framing and error handling on one link.

WATCH OUT FOR

Pairing the transport layer with framing and error handling, or the network layer with host to host communication. Framing belongs to the data link layer, and end-to-end delivery belongs to the transport layer.

Q.17 MCQ · 1 MARK

Discrete Mathematics › Sets & relations

$g(\cdot)$ is a function from A to B, $f(\cdot)$ is a function from B to C, and their composition defined as $f(g(\cdot))$ is a mapping from A to C.

If $f(\cdot)$ and $f(g(\cdot))$ are onto (surjective) functions, which ONE of the following is TRUE about the function $g(\cdot)$?

(A) $g(\cdot)$ must be an onto (surjective) function.

(B) $g(\cdot)$ must be a one-to-one (injective) function.

(C) $g(\cdot)$ must be a bijective function, that is, both one-to-one and onto.

(D) $g(\cdot)$ is not required to be a one-to-one or onto function.

Official answer: **D**

SOLUTION

1. Given: $f : B \to C$ is onto and $f \circ g : A \to C$ is onto. We ask what is forced on $g : A \to B$.
2. The condition on f is already given, so the only other condition is that $f \circ g$ is onto. Whether g must be onto or one-to-one can be settled with small counterexamples that satisfy both given conditions.
3. Test whether g must be onto. Take $A = \{1\}$, $B = \{1, 2\}$, $C = \{1\}$, with $g(1) = 1$ and $f(1) = f(2) = 1$. Then f is onto C, and $(f \circ g)(1) = 1$ so $f \circ g$ is onto C. But $2 \in B$ is not in the range of g, so g is not onto. This rules out (A) and also (C), since a bijective function must be onto.
4. Test whether g must be one-to-one. Take $A = \{1, 2\}$, $B = \{1\}$, $C = \{1\}$, with $g(1) = g(2) = 1$ and $f(1) = 1$. Then f is onto C, and $f \circ g$ maps both elements to 1, so it is onto C. But $g(1) = g(2)$ with $1 \neq 2$, so g is not one-to-one. This rules out (B).
5. Both properties can fail while the hypotheses hold, so g is not required to be one-to-one or onto. Option (D) is true.

KEY IDEA

Knowing that f and $f \circ g$ are onto says nothing about injectivity or surjectivity of g. To refute a "must" claim, one small counterexample that meets all the hypotheses is enough.

WATCH OUT FOR

Thinking that because $f \circ g$ is onto, g must be onto. The true fact is that $f \circ g$ onto forces f to be onto, not g. Likewise, $f \circ g$ one-to-one would force g to be one-to-one, but nothing here gives that.

Q.18 MCQ · 1 MARK

Data Structures and Algorithms › Graph algorithms

Let G be any undirected graph with positive edge weights, and T be a minimum spanning tree of G. For any two vertices, u and v, let $d_1(u, v)$ and $d_2(u, v)$ be the shortest distances between u and v in G and T, respectively. Which ONE of the options is CORRECT for all possible G, T, u and v?

(A) $d_1(u, v) = d_2(u, v)$

(B) $d_1(u, v) \leq d_2(u, v)$

(C) $d_1(u, v) \geq d_2(u, v)$

(D) $d_1(u, v) \neq d_2(u, v)$

Official answer: **B**

SOLUTION

1. T is a spanning tree of G, so every edge of T is an edge of G with the same weight; T is a subgraph of G on the same vertex set.
2. Any path between u and v that lies entirely in T therefore also exists in G, using the same edges and hence the same total weight.

3. So the set of all u-v paths available in G is a superset of the set of all u-v paths available in T (every T-path is also a G-path, and G may additionally have paths that use non-tree edges).
4. $d_2(u, v)$ is the minimum weight over the (smaller) set of T-paths; $d_1(u, v)$ is the minimum weight over the (larger or equal) set of G-paths.
5. Minimizing a quantity over a superset of candidates can never give a larger value than minimizing over a subset of those candidates, since every candidate available to the subset is also available to the superset: $d_1(u, v) \leq d_2(u, v)$.
6. This holds for every choice of G, T, u, v (including cases where u, v are adjacent in T or where G itself is a tree, in which equality $d_1(u, v) = d_2(u, v)$ happens to hold), but equality is not guaranteed in general (e.g. a shorter path in G can bypass the tree route via non-tree edges), so (A) is not always true, and since $d_1 \leq d_2$ always, $d_1 > d_2$ (needed for (C) alone) and $d_1 \neq d_2$ (option D) can both fail whenever equality holds.
7. Hence the only relation guaranteed for all G, T, u, v is $d_1(u, v) \leq d_2(u, v)$: option (B).

KEY IDEA

A minimum spanning tree T only removes edges from G; it never adds a shorter route. So every path available inside T is also available inside G, meaning shortest-path distance can only stay the same or shrink when you move from T to G: $d_1(u, v) \leq d_2(u, v)$, always.

WATCH OUT FOR

Assuming a "minimum" spanning tree must preserve shortest-path distances (confusing MST with a shortest-path tree) and picking (A); or reasoning direction backwards and picking (C) by thinking "the tree is smaller so distances in it must be smaller."

Q.19 MCQ · 1 MARK

Theory of Computation › CFG & PDA

Consider the following context-free grammar G, where S, A, and B are the variables (non-terminals), a and b are the terminal symbols, S is the start variable, and the rules of G are described as:

$$S \to aaB \mid Abb$$

$$A \to a \mid aA$$

$$B \to b \mid bB$$

Which ONE of the languages $L(G)$ is accepted by G?

(A) $L(G) = \{a^2b^n \mid n \geq 1\} \cup \{a^nb^2 \mid n \geq 1\}$

(B) $L(G) = \{a^nb^{2n} \mid n \geq 1\} \cup \{a^{2n}b^n \mid n \geq 1\}$

(C) $L(G) = \{a^nb^n \mid n \geq 1\}$

(D) $L(G) = \{a^{2n}b^{2n} \mid n \geq 1\}$

Official answer: **A**

Q.20 MCQ · 1 MARK

Data Structures and Algorithms › Complexity

Consider the following recurrence relation:

$$T(n) = 2T(n-1) + n2^n \text{ for } n > 0, \quad T(0) = 1.$$

Which ONE of the following options is CORRECT?

(A) $T(n) = \Theta(n^2 2^n)$

(B) $T(n) = \Theta(n2^n)$

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

(D) $T(n) = \Theta(4^n)$

Official answer: **A**

SOLUTION

1. Recurrence: T(n) = 2T(n-1) + n·2ⁿ for n > 0, with base case T(0) = 1.
2. Define S(n) = T(n)/2ⁿ to normalize out the exponential growth, so T(n) = 2ⁿS(n) and T(n-1) = 2^{n-1}S(n-1).
3. Substituting into the recurrence: 2ⁿS(n) = 2·2^{n-1}S(n-1) + n·2ⁿ = 2ⁿS(n-1) + n·2ⁿ. Dividing both sides by 2ⁿ gives S(n) = S(n-1) + n, with S(0) = T(0)/2⁰ = 1.
4. Unrolling this simple additive recurrence: S(n) = S(0) + (1 + 2 + ... + n) = 1 + n(n+1)/2.
5. Since S(n) = 1 + n(n+1)/2 is a quadratic polynomial in n, S(n) = Θ(n²).
6. Therefore T(n) = 2ⁿ·S(n) = 2ⁿ·Θ(n²) = Θ(n²2ⁿ), which is option (A).

KEY IDEA

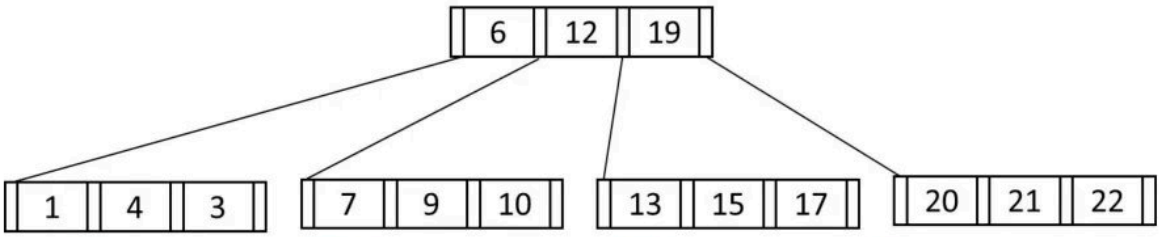
When a recurrence has the form T(n) = 2T(n-1) + f(n)·2^n, dividing through by 2^n turns the exponential recurrence into a simple additive one: define S(n) = T(n)/2^n, and the recurrence collapses to S(n) = S(n-1) + f(n), which can be solved by direct summation.

WATCH OUT FOR

Solving the recurrence directly by repeated substitution and losing track of the accumulating n·2^n terms (or mis-summing them) leads to an off-by-one-degree answer such as Θ(n2ⁿ) (option B) or assuming the extra n factor doesn't compound, missing that it must be summed over n terms to give the n² factor.

Q.21 MSQ · 1 MARK

Database Management and Warehousing › Indexing

Q.21	Consider the following B^+ tree with 5 nodes, in which a node can store at most 3 key values. The value 23 is now inserted in the B^+ tree. Which of the following options(s) is/are CORRECT?
	
(A)	None of the nodes will split.
(B)	At least one node will split and redistribute.
(C)	The total number of nodes will remain same.
(D)	The height of the tree will increase.

Official answer: **B, D**

SOLUTION

1. Each node holds at most 3 keys. The root $[6, 12, 19]$ and all four leaves $[1, 4, 3]$, $[7, 9, 10]$, $[13, 15, 17]$, $[20, 21, 22]$ are full.
2. Search for 23. Since $23 > 19$, it belongs in the rightmost leaf $[20, 21, 22]$, which is already full.
3. Insert 23 there. The leaf overflows to $[20, 21, 22, 23]$ and splits into $[20, 21]$ and $[22, 23]$. In a B^+ tree the first key of the new right leaf, 22, is copied up to the parent.
4. The parent is the root $[6, 12, 19]$, which is also full. Adding 22 gives $[6, 12, 19, 22]$ with five children, so the root overflows and splits too.
5. The internal split keeps $[6, 12]$ (children $[1, 4, 3]$, $[7, 9, 10]$, $[13, 15, 17]$) and $[22]$ (children $[20, 21]$, $[22, 23]$). The middle key 19 moves up (not copied) into a new root $[19]$.
6. Count the nodes. Before: 5. After: one new leaf, one new internal node and one new root, so $5 + 3 = 8$. The height goes from 2 levels to 3.
7. Check each option. (A) False, since the leaf and the root split. (B) True, since at least one node splits. (C) False, since the count rises from 5 to 8. (D) True, since a new root was created.

KEY IDEA

When a B^+ leaf overflows, it splits and copies a key up. If every ancestor up to the root is full, the splits cascade and a new root is made, so the height grows by one.

WATCH OUT FOR

Stopping after the leaf split and forgetting that the parent is also full. That leads to choosing (C), or to choosing (A) because only one node seems to split.

Q.22 MSQ · 1 MARK

Computer Networks › TCP

Consider the 3-way handshaking protocol for TCP connection establishment. Let the three packets exchanged during the connection establishment be denoted as P1, P2, and P3, in order. Which of the following option(s) is/are TRUE with respect to TCP header flags that are set in the packets?

(A) P3: SYN = 1, ACK = 1

(B) P2: SYN = 1, ACK = 1

(C) P2: SYN = 0, ACK = 1

(D) P1: SYN = 1

Official answer: **B, D**

SOLUTION

1. The three-way handshake uses three packets: P1 goes from client to server, P2 from server to client, and P3 from client to server.
2. P1 is the connection request. Only SYN is set, so $\mathrm{SYN} = 1$ and $\mathrm{ACK} = 0$. Option (D), "P1: SYN = 1", is true.
3. P2 is the server's reply. It acknowledges the client's SYN and carries the server's own SYN, so $\mathrm{SYN} = 1$ and $\mathrm{ACK} = 1$. Option (B) is true and option (C), "P2: SYN = 0, ACK = 1", is false.
4. P3 is the client's final acknowledgement of the server's SYN. It carries no SYN, so $\mathrm{SYN} = 0$ and $\mathrm{ACK} = 1$. Option (A), "P3: SYN = 1, ACK = 1", is false.
5.

Packet	SYN	ACK
P_1	1	0
P_2	1	1
P_3	0	1

6. The true statements are (B) and (D).

KEY IDEA

In the TCP three-way handshake the flags are SYN for the first packet, SYN+ACK for the second, and ACK alone for the third. Only the first packet has ACK = 0, and only the third has SYN = 0.

WATCH OUT FOR

Setting SYN again in the final packet P3 (option A). The client's last packet only acknowledges the server's SYN, so its SYN flag is 0. Another slip is reading P2 as a plain ACK (option C), when the server must also send its own SYN.

Q.23 MSQ · 1 MARK

Engineering Mathematics › Linear algebra

Consider the given system of linear equations for variables x and y, where k is a real-valued constant. Which of the following option(s) is/are CORRECT?

$$x + ky = 1$$
$$kx + y = -1$$

(A) There is exactly one value of k for which the above system of equations has no solution.

(B) There exist an infinite number of values of k for which the system of equations has no solution.

(C) There exists exactly one value of k for which the system of equations has exactly one solution.

(D) There exists exactly one value of k for which the system of equations has an infinite number of solutions.

Official answer: **A, D**

SOLUTION

1. Write the system as $\begin{bmatrix}1 & k\\ k & 1\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}1\\ -1\end{bmatrix}$.
2. The determinant of the coefficient matrix is $1-k^2$. A square system has exactly one solution when $\det \neq 0$, that is, when $k \neq \pm 1$.
3. So every real $k \neq \pm 1$ gives exactly one solution. That is infinitely many values of k, so option C (exactly one such value) is false.
4. Case $k=1$: the equations are $x+y=1$ and $x+y=-1$. They contradict each other, so there is no solution.
5. Case $k=-1$: the equations are $x-y=1$ and $-x+y=-1$. The second is the first multiplied by -1, so both describe the same line and there are infinitely many solutions.
6. Hence no solution occurs only for $k=1$ (A true, B false), and infinitely many solutions occur only for $k=-1$ (D true). The correct options are A and D.

KEY IDEA

For a square system $A\mathbf{x}=\mathbf{b}$, $\det A \neq 0$ gives a unique solution. Where $\det A = 0$, check each singular value separately for consistency. A consistent singular system has infinitely many solutions, and an inconsistent one has none.

WATCH OUT FOR

Stopping once $\det = 0$ gives $k=\pm 1$ and treating both values alike. The right-hand side $(1,-1)$ makes $k=1$ inconsistent but $k=-1$ consistent. Another slip is reading option C as true because the determinant condition looks like a single value.

Q.24 MSQ · 1 MARK

Digital Electronics › Boolean logic

Let X be a 3-variable Boolean function that produces output as '1' when at least two of the input variables are '1'. Which of the following statement(s) is/are CORRECT, where a, b, c, d, e are Boolean variables?

(A) $X(a,b,X(c,d,e)) = X(X(a,b,c),d,e)$

(B) $X(a,b,X(a,b,c)) = X(a,b,c)$

(C) $X(a,b,X(a,c,d)) = (X(a,b,a)\ \text{AND}\ X(c,d,c))$

(D) $X(a,b,c) = X(a,X(a,b,c),X(a,c,c))$

Official answer: **B, D**

SOLUTION

1. X is the majority function: $X(x,y,z)=xy+yz+zx$. Two facts follow. First, $X(x,y,x)=x$, because a repeated input wins. Second, if $x \neq y$ then $X(x,y,z)=z$, because the third input decides.
2. (A) Take $a=b=1$, $c=d=e=0$. Left side: $X(1,1,X(0,0,0))=X(1,1,0)=1$. Right side: $X(X(1,1,0),0,0)=X(1,0,0)=0$. The sides differ, so (A) is false.
3. (B) If $a=b$: the left side is $X(a,a,\cdot)=a$, and $X(a,b,c)=X(a,a,c)=a$, so they match. If $a \neq b$: $X(a,b,c)=c$, so the left side is $X(a,b,c)=c$. The right side is also $X(a,b,c)=c$. (B) is true.
4. (C) The right side is $X(a,b,a)\cdot X(c,d,c)=a\cdot c$. Take $a=b=1$, $c=d=0$. Left side: $X(1,1,X(1,0,0))=X(1,1,0)=1$. Right side: $1\cdot 0=0$. They differ, so (C) is false.
5. (D) Since $X(a,c,c)=c$, the right side is $X(a,X(a,b,c),c)$. If $a=c$: it equals a (repeated input), and $X(a,b,c)=X(a,b,a)=a$, so they match. If $a \neq c$: the third input decides, so the right side equals $X(a,b,c)$. (D) is true.
6. The correct options are (B) and (D).

KEY IDEA
X is the majority function. When two inputs are equal, that value is the output. When they differ, the third input decides. Split each identity into the cases on the repeated inputs and check them, or use a counter-example.

WATCH OUT FOR
Assuming majority is associative, as AND and OR are, and accepting (A). Another slip is reading (C) as a distributive law. The right side reduces to $a \cdot c$, which is not the same as the left side.

Q.25 MSQ · 1 MARK

Digital Electronics › Number systems

The number -6 can be represented as 1010 in 4-bit 2's complement representation. Which of the following is/are CORRECT 2's complement representation(s) of -6?

(A) 1000 1010 in 8-bits

(B) 1111 1010 in 8-bits

(C) 1000 0000 0000 1010 in 16-bits

(D) 1111 1111 1111 1010 in 16-bits

Official answer: **B, D**

SOLUTION

1. An n-bit 2's complement number has value $-b_{n-1}2^{n-1} + \sum_{i<n-1} b_i 2^i$. Check the 4-bit form: $1010 = -8 + 2 = -6$.
2. To widen a 2's complement number, copy the sign bit into the new high bits (sign extension). The value does not change.
3. The sign bit is 1, so the 8-bit form is $1111\,1010$ and the 16-bit form is $1111\,1111\,1111\,1010$.
4. Check (A): $1000\,1010 = -128 + 10 = -118 \neq -6$.
5. Check (B): $1111\,1010 = -128 + 122 = -6$. Correct.
6. Check (C): $1000\,0000\,0000\,1010 = -32768 + 10 = -32758 \neq -6$.
7. Check (D): $1111\,1111\,1111\,1010 = -32768 + 32762 = -6$. Correct.
8. So the correct options are (B) and (D).

KEY IDEA
Widening a 2's complement number is done by sign extension: copy the sign bit into every new high-order position. The value stays the same.

WATCH OUT FOR
Padding with zeros on the left, or keeping the sign bit as the leading bit and padding zeros after it. That gives $1000\,1010$ or $1000\,0000\,0000\,1010$, which are large negative numbers, not -6.

Q.26 MSQ · 1 MARK

Data Structures and Algorithms › Trees & heaps

Which of the following statement(s) is/are TRUE for any binary search tree (BST) having n distinct integers?

(A) The maximum length of a path from the root node to any other node is $(n-1)$.

(B) An inorder traversal will always produce a sorted sequence of elements.

(C) Finding an element takes $O(\log_2 n)$ time in the worst case.

(D) Every BST is also a Min-Heap.

Official answer: **A, B**

SOLUTION

1. **Option A (TRUE).** A path from the root down to any node is a simple path, so it visits each node at most once. The tree has only n nodes, so such a path has at most n nodes and at most $n-1$ edges. This limit is actually reached: insert $1, 2, \ldots, n$ in increasing order to get a right-skewed BST, and the path

from the root 1 to the node n has exactly $n-1$ edges. So for every BST on n distinct keys, the longest a root-to-node path can be is $n-1$. The statement gives this upper bound. It does not claim that every BST has height $n-1$.

2. **Option B (TRUE).** At every node, all keys in the left subtree are smaller than the node's key and all keys in the right subtree are larger. An inorder traversal visits the left subtree, then the node, then the right subtree. By induction on the subtree size, each subtree is output in increasing order, then the node's key, which is larger than everything already printed and smaller than everything still to come. So the whole sequence is sorted in increasing order, whatever the shape of the tree.
3. **Option C (FALSE).** A search follows one root-to-leaf path, so its worst-case cost is proportional to the height of the tree. In the skewed BST from option A the height is $n-1$, and searching for n (or for any key greater than n) takes n comparisons. That is $\Theta(n)$, not $O(\log_2 n)$. The $O(\log_2 n)$ bound holds only for a balanced BST, such as an AVL or red-black tree.
4. **Option D (FALSE).** A min-heap needs every parent $\leq$ its children. In a BST, a left child is always *smaller* than its parent. Take the BST with root 2, left child 1 and right child 3: the parent 2 is greater than its child 1, so the min-heap property fails. Any BST where some node has a left child breaks the min-heap property in the same way.
5. The statements that hold for every BST on n distinct integers are (A) and (B).

KEY IDEA

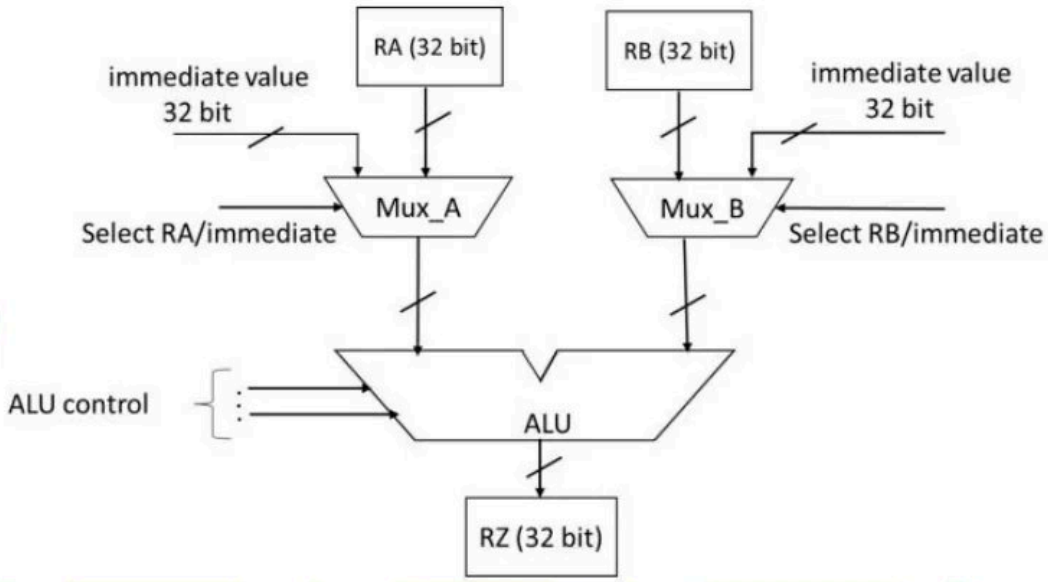
A root-to-node path cannot repeat a node, so in any n-node tree it has at most $n-1$ edges. A skewed BST reaches that limit. Inorder traversal of a BST (left, node, right) always gives the keys in sorted order. Search cost depends on the height, which can be as large as $n-1$, so an unbalanced BST has $\Theta(n)$ worst-case search. A left child is always smaller than its parent, so a BST with any left child cannot be a min-heap.

WATCH OUT FOR

The common mistake is to read (A) as "every BST has height $n-1$" and reject it because a balanced BST is only $O(\log n)$ deep. The statement gives the largest length a root-to-node path can have in any BST, and $n-1$ is that upper bound. It is reached by a skewed tree. A second mistake is to accept (C) by assuming the BST is balanced. A plain BST has no such guarantee.

Q.27 MSQ · 1 MARK

Computer Organization and Architecture › ALU & control unit

Q.27	A partial data path of a processor is given in the figure, where RA, RB, and RZ are 32-bit registers. Which option(s) is/are CORRECT related to arithmetic operations using the data path as shown?
	
(A)	The data path can implement arithmetic operations involving two registers.
(B)	The data path can implement arithmetic operations involving one register and one immediate value.
(C)	The data path can implement arithmetic operations involving two immediate values.
(D)	The data path can only implement arithmetic operations involving one register and one immediate value.

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

SOLUTION

1. Mux_A is a 2:1 multiplexer whose inputs are RA and a 32-bit immediate value; it is controlled by its own dedicated select line labeled "Select RA/immediate".
2. Mux_B is a 2:1 multiplexer whose inputs are RB and a 32-bit immediate value; it is controlled by its own dedicated select line labeled "Select RB/immediate".
3. Because these two select lines are physically separate signals in the figure (not a single shared control), they can be set independently, so all 2×2 = 4 combinations of (Mux_A output, Mux_B output) are realizable at the ALU's two 32-bit operand inputs.
4. Combination (RA, RB): both muxes select their register input → the ALU operates on two registers → option (A) is correct.

5. Combination (RA, immediate) or (immediate, RB): one mux selects its register, the other selects its immediate input → the ALU operates on one register and one immediate value → option (B) is correct.
6. Combination (immediate, immediate): both muxes select their immediate input → the ALU operates on two immediate values → option (C) is correct.
7. Since two-register operation (A) and two-immediate operation (C) are both achievable, the data path is not restricted to only "one register and one immediate value" operations, so option (D)'s exclusive claim is false.
8. Final answer: options (A), (B) and (C) are correct.

KEY IDEA

Mux_A and Mux_B are controlled by two separate, independent select lines ("Select RA/immediate" and "Select RB/immediate"), so all four combinations of their inputs (RA/RB, RA/immediate, immediate/RB, immediate/immediate) are physically realizable at the ALU's two operand ports.

WATCH OUT FOR

Assuming the two multiplexer select lines must be tied together (i.e., both muxes always pick "register" or both always pick "immediate" in lock-step), which would wrongly eliminate option (C) and make (D) look correct. The figure shows two independent select lines ("Select RA/immediate" and "Select RB/immediate"), so each mux can be controlled separately.

Q.28 MSQ · 1 MARK

Theory of Computation › Finite automata

A regular language L is accepted by a non-deterministic finite automaton (NFA) with n states. Which of the following statement(s) is/are FALSE?

(A) L may have an accepting NFA with $< n$ states.

(B) L may have an accepting DFA with $< n$ states.

(C) There exists a DFA with $\leq 2^n$ states that accepts L.

(D) Every DFA that accepts L has $> 2^n$ states.

Official answer: **D**

SOLUTION

1. The question asks which statements are FALSE. The NFA has n states, but nothing says n is the smallest number of states any NFA for L needs.
2. (A) Take $L = \Sigma^*$. It is accepted by an n-state NFA for any $n \geq 2$ (add extra states that do not change the language). A 1-state NFA also accepts it, and $1 < n$. So L may have an NFA with fewer than n states, and (A) is TRUE.
3. (B) In the same example, a 1-state DFA with a single accepting state looping on every symbol accepts Σ^*, and $1 < n$. So L may have a DFA with fewer than n states, and (B) is TRUE.
4. (C) The subset construction turns an n-state NFA into a DFA whose states are subsets of the NFA's states. There are at most 2^n subsets, so a DFA with $\leq 2^n$ states accepting L exists. (C) is TRUE.
5. (D) The claim is that every DFA for L has more than 2^n states. By (C), some DFA for L has at most 2^n states, so the claim fails. (D) is FALSE.
6. Only (D) is a false statement, so the answer is (D).

KEY IDEA

The subset construction gives a DFA with at most 2^n states for any n-state NFA, so no statement can say every DFA needs more than 2^n states. "May have" statements only need one example, such as Σ^*.

WATCH OUT FOR

Assuming the given NFA is minimal, which makes (A) and (B) look false. Another slip is reading 2^n as a lower bound on DFA size, when it is only an upper bound that some languages reach.

Q.29 NAT · 1 MARK

Operating Systems › Processes & threads

Suppose in a multiprogramming environment, the following C program segment is executed. A process goes into I/O queue whenever an I/O related operation is performed. Assume that there will always be a context switch whenever a process requests for an I/O, and also whenever the process returns from an I/O. The number of times the process will enter the ready queue during its lifetime (not counting the time the process enters the ready queue when it is run initially) is _____. (Answer in integer)

```
int main()
{
  int x=0,i=0;
  scanf("%d",&x);
  for(i=0; i<20; i++)
  {
    x = x+20;
    printf("%d\n",x);
  }
  return 0;
}
```

Official answer: **21**

SOLUTION

1. Each I/O request moves the process to the I/O queue. When that I/O completes, the process moves from the I/O queue back to the ready queue. So the process enters the ready queue once per completed I/O operation.
2. The call `scanf("%d",&x)` reads from the keyboard. That is 1 I/O operation.
3. The loop `for(i=0; i<20; i++)` runs for $i = 0, 1, \ldots, 19$, which is 20 iterations. Each iteration calls `printf`, which writes to the output device. That is 20 I/O operations. The statement `x = x+20` is pure computation and causes no I/O.
4. The declarations and `return 0` cause no I/O.
5. Total I/O operations $= 1 + 20 = 21$.
6. Each of the 21 I/O completions puts the process back in the ready queue once. The question does not count the first entry to the ready queue, when the process is run initially.
7. Number of ready-queue entries $= 21$.

KEY IDEA
Count the I/O operations. Every I/O request sends the process to the I/O queue, and every I/O completion brings it back to the ready queue, so each I/O call gives exactly one ready-queue entry.

WATCH OUT FOR
Counting only the 20 `printf` calls and forgetting the `scanf` (giving 20), or adding the initial entry to the ready queue (giving 22). The arithmetic statement `x = x+20` is not an I/O operation.

Q.30 NAT · 1 MARK

Discrete Mathematics › Counting

Let S be the set of all ternary strings defined over the alphabet $\{a, b, c\}$. Consider all strings in S that contain at least one occurrence of two consecutive symbols, that is, "aa", "bb" or "cc". The number of such strings of length 5 that are possible is _____. (Answer in integer)

Official answer: **195**

SOLUTION

1. Count the complement: strings of length 5 over $\{a, b, c\}$ in which no two adjacent symbols are equal. Then answer $=$ total $-$ complement.

2. Total strings of length 5 over 3 symbols: $3^5 = 243$.
3. A string has none of "aa", "bb", "cc" exactly when every symbol differs from the one before it. The first symbol has 3 choices. Each later symbol must differ from its predecessor, so it has 2 choices.
4. Complement count $= 3 \times 2^4 = 3 \times 16 = 48$. Check on length 2: the formula gives $3 \times 2 = 6$ strings with no repeat (ab, ac, ba, bc, ca, cb), and $9 - 6 = 3$ strings (aa, bb, cc) do contain a repeat, which is correct.
5. Strings with at least one of "aa", "bb" or "cc" $= 243 - 48 = 195$.

KEY IDEA

Count the complement. Strings with no two equal adjacent symbols number $3 \cdot 2^{n-1}$, because each new symbol has exactly 2 choices that differ from the previous one. Subtract this from 3^n.

WATCH OUT FOR

Counting strings containing "aa", "bb" or "cc" directly and adding the three counts. This double-counts strings that contain more than one repeat, such as "aabbc". Another slip is to allow 3 choices for every position of the complement instead of 2 for positions after the first.

Q.31 NAT · 1 MARK

Engineering Mathematics › Calculus

Consider the given function $f(x)$.

$$f(x) = \begin{cases} ax + b & \text{for } x < 1 \\ x^3 + x^2 + 1 & \text{for } x \geq 1 \end{cases}$$

If the function is differentiable everywhere, the value of b must be _____. (rounded off to one decimal place)

Official answer: **-2.1 to -1.9**

SOLUTION

1. Differentiable everywhere implies continuous at $x = 1$, so the value of the left piece as $x \to 1^-$ equals the value of the right piece at $x = 1$.
2. Right piece at $x = 1$: $1^3 + 1^2 + 1 = 3$. Left piece: $a(1) + b = a + b$. So $a + b = 3$.
3. Differentiability at $x = 1$ also needs equal one-sided derivatives. Left derivative: $\frac{d}{dx}(ax + b) = a$. Right derivative: $\frac{d}{dx}(x^3 + x^2 + 1) = 3x^2 + 2x$, which at $x = 1$ is $3 + 2 = 5$.
4. So $a = 5$.
5. Then $b = 3 - a = 3 - 5 = -2$.
6. Hence $b = -2.0$.

KEY IDEA

For a piecewise function to be differentiable at the join, both the function values (continuity) and the one-sided derivatives must match there. That gives two equations for the two unknowns a and b.

WATCH OUT FOR

Imposing only continuity ($a + b = 3$), which leaves a and b undetermined. Another slip is evaluating the right-hand derivative as $3x^2 + 2x + 1$ or forgetting that the constant 1 differentiates to 0.

Q.32 NAT · 1 MARK

Engineering Mathematics › Probability & statistics

A box contains 5 coins: 4 regular coins and 1 fake coin. When a regular coin is tossed, the probability $P(head) = 0.5$ and for a fake coin, $P(head) = 1$. You pick a coin at random and toss it twice, and get two

heads. The probability that the coin you have chosen is the fake coin is _____. (rounded off to two decimal places)

Official answer: **0.49 to 0.51**

SOLUTION

1. Let F be the event that the chosen coin is the fake coin and R the event that it is regular. Then $P(F) = \frac{1}{5}$ and $P(R) = \frac{4}{5}$. Let HH be the event of two heads in two tosses.
2. The tosses are independent, so $P(HH \mid F) = 1 \cdot 1 = 1$ and $P(HH \mid R) = 0.5 \times 0.5 = \frac{1}{4}$.
3. By the law of total probability, $P(HH) = \frac{1}{5} \cdot 1 + \frac{4}{5} \cdot \frac{1}{4} = \frac{1}{5} + \frac{1}{5} = \frac{2}{5}$.
4. By Bayes' theorem,
$$P(F \mid HH) = \frac{P(HH \mid F)\,P(F)}{P(HH)} = \frac{1/5}{2/5} = 0.5$$
5. Rounded to two decimal places, the answer is 0.50.

KEY IDEA
Two heads is evidence for the fake coin. Use Bayes' theorem, with the prior $P(F) = 1/5$ and the likelihoods $P(HH \mid F) = 1$ and $P(HH \mid R) = 1/4$.

WATCH OUT FOR
Taking the prior $1/5 = 0.20$ as the answer, or computing only $P(HH \mid R)$ and forgetting to weight it by $P(R) = 4/5$. Using $P(R) = 1/2$ instead of $4/5$ also gives a wrong result.

Q.33 NAT · 1 MARK

Data Structures and Algorithms › Sorting & hashing

The pseudocode of a function `fun()` is given below:

```
fun(int A[0,...,n-1]){
  for i=0 to n-2
    for j=0 to n-i-2
      if (A[j]>A[j+1])
        then swap A[j] and A[j+1]
}
```

Let $A[0, \ldots, 29]$ be an array storing 30 distinct integers in descending order. The number of swap operations that will be performed, if the function `fun()` is called with $A[0, \ldots, 29]$ as argument, is _____. (Answer in integer)

Official answer: **435**

SOLUTION

1. The pseudocode is bubble sort with an ascending-sort swap rule: outer loop i = 0 to n-2, inner loop j = 0 to n-i-2, and whenever `A[j] > A[j+1]` the two are swapped.
2. Total number of comparisons made, from the loop bounds alone (independent of array contents):
$$\sum_{i=0}^{n-2}(n-i-1) = (n-1) + (n-2) + \cdots + 1 = \frac{n(n-1)}{2}.$$
3. Claim: for A[0..n-1] strictly descending, EVERY comparison the algorithm makes evaluates to true, so every comparison is also a swap. Proof sketch: within one outer pass over a strictly descending sub-array, at each inner step the value currently sitting in A[j] is the running maximum carried forward by the previous swaps in this pass, and A[j+1] is the next not-yet-touched original element, which is always strictly smaller than that running maximum. So A[j] > A[j+1] holds at every single comparison, in every pass.
4. Check on a small case, n = 4, A = [4,3,2,1]: pass i=0 compares (4,3), (4,2), (4,1): all true, all swap, ending [3,2,1,4]; pass i=1 compares (3,2), (3,1): all true, all swap, ending [2,1,3,4]; pass i=2 compares

(2,1): true, swap, ending [1,2,3,4]. All 4·3/2 = 6 comparisons were swaps, confirming the general argument.

5. Since every comparison is a swap, number of swaps = number of comparisons = n(n-1)/2. With n = 30: swaps = 30×29/2 = 435.
6. This matches the fact that bubble sort's swap count always equals the number of inversions in the input, and a strictly descending array of 30 distinct elements has the maximum possible number of inversions, $\binom{30}{2} = 435$.

KEY IDEA

This pseudocode is bubble sort with the comparison `A[j] > A[j+1]` triggering a swap, the standard scheme that sorts an array into ascending order. When the input is already sorted in the exact opposite (fully descending) order, this is the worst case for bubble sort: every comparison the algorithm ever makes finds a violation and swaps, so the swap count equals the comparison count, which equals the total number of inversions in the array, $\binom{n}{2} = \frac{n(n-1)}{2}$.

WATCH OUT FOR

Confusing "number of comparisons" with "number of swaps" when the array is NOT already sorted the way the algorithm wants: students correctly compute n(n-1)/2 comparisons but then assume swaps must be fewer (e.g. by miscounting how many passes actually trigger a swap), forgetting that a strictly descending input is exactly the worst case for this ascending-sorting bubble sort, where every comparison is also a swap.

Q.34 NAT · 1 MARK

Programming in C › Pointers & arrays

```
#include <stdio.h>
void foo(int *p, int x){
   *p=x;
}
int main(){
   int *z;
   int a = 20, b = 25;
   z = &a;
   foo(z,b);
   printf("%d",a);
   return 0;
}
```

The output of the given C program is _____. (Answer in integer)

Official answer: **25**

SOLUTION

1. Declarations: `int *z;` and `int a = 20, b = 25;` set up an uninitialized pointer `z` and two ints, a = 20 and b = 25.
2. `z = &a;` makes z hold the address of a.
3. `foo(z, b)` is called: the parameter `p` receives the value of z (i.e., the address of a), and the parameter `x` receives the value of b (25) as a copy, not a reference.
4. Inside foo, `*p = x;` dereferences p (which points to a) and stores the value of x (25) into that memory location, i.e., into a.
5. Because p and z both hold the address of a, this write is visible in main: a is now 25 (its original value 20 is overwritten). b remains unchanged at 25 but plays no further role.
6. foo returns; `printf("%d", a);` prints the current value of a, which is 25.
7. Output: 25.

KEY IDEA

`foo` receives a pointer `p` and a value `x`. Since `p` is a copy of `z` (i.e., `p` also holds the address of `a`), the statement `*p = x` writes through the pointer directly into `a` in `main`, overwriting it with the value of `x` (a copy of `b`).

WATCH OUT FOR

Confusing pass-by-value with pass-by-reference: assuming that because `b` (value 25) is passed by value, the assignment inside `foo` cannot affect the caller's variables, missing that it is the pointer `z` (holding `&a`) that is passed, and dereferencing it writes directly into `a`.

Q.35 NAT · 1 MARK

Data Structures and Algorithms › Trees & heaps

The height of any rooted tree is defined as the maximum number of edges in the path from the root node to any leaf node.

Suppose a Min-Heap T stores 32 keys. The height of T is _____.
(Answer in integer)

Official answer: **5**

SOLUTION

1. A Min-Heap storing 32 keys is implemented as a complete binary tree (the standard array-based heap representation), meaning every level is fully filled except possibly the last, which is filled left to right.
2. For a complete binary tree of height h (height = maximum number of edges on any root-to-leaf path), the number of nodes n satisfies $2^h \le n \le 2^{h+1} - 1$: the minimum occurs when the last level has just 1 node, the maximum when the last level is completely full.
3. We need the h for which $2^h \le 32 \le 2^{h+1} - 1$.
4. Check $h = 5$: the range is $2^5 = 32$ to $2^6 - 1 = 63$. Since 32 lies in $[32, 63]$, $h = 5$ satisfies the bound.
5. Check $h = 4$: the range is $2^4 = 16$ to $2^5 - 1 = 31$. Since $32 \notin [16, 31]$, $h = 4$ is ruled out.
6. Equivalently, $h = \lfloor \log_2 32 \rfloor = \lfloor 5 \rfloor = 5$, confirming the same result: levels 0 through 4 together hold $2^5 - 1 = 31$ nodes, so the 32^{nd} node is the sole node on level 5, giving a root-to-leaf path of 5 edges.
7. So the height of the Min-Heap T storing 32 keys is 5.

KEY IDEA

A binary heap is stored as a complete binary tree. For n nodes, a complete binary tree of height h (height = max edges on a root-to-leaf path) satisfies $2^h \le n \le 2^{h+1} - 1$, equivalently $h = \lfloor \log_2 n \rfloor$.

WATCH OUT FOR

Confusing "height" (edges, 0 for a single node) with "number of levels" (which would give 6 for $n = 32$), or using the wrong bound direction ($2^{h+1} \le n$) and getting $h = 4$.

Q.36 MCQ · 2 MARKS

Computer Organization and Architecture › Memory & cache

Consider a memory system with 1M bytes of main memory and 16K bytes of cache memory. Assume that the processor generates 20-bit memory address, and the cache block size is 16 bytes. If the cache uses direct mapping, how many bits will be required to store all the *tag* values? [Assume memory is byte addressable, 1K=2^{10}, 1M=2^{20}.]

(A) 6×2^{10}

(B) 8×2^{10}

(C) 2^{12}

(D) 2^{14}

Official answer: **A**

SOLUTION

1. Main memory = 1M = 2^{20} bytes, so the processor address is 20 bits (given).
2. Cache size = 16K = 2^{14} bytes; block size = 16 = 2^4 bytes.
3. Number of cache lines = cache size / block size = $2^{14}/2^4 = 2^{10} = 1024$ lines.
4. Block offset bits = $\log_2(\text{block size}) = \log_2(2^4) = 4$ bits.
5. Direct mapping: index (line) bits = $\log_2(\text{number of lines}) = \log_2(2^{10}) = 10$ bits.
6. Tag bits per line = total address bits − index bits − offset bits = $20 - 10 - 4 = 6$ bits.
7. Total tag storage across the whole cache = tag bits per line × number of lines = 6×2^{10}.
8. This matches option (A).

KEY IDEA

In direct-mapped caches, the address splits into Tag | Line Index | Block Offset. The total tag storage for the whole cache is (tag bits per line) × (number of lines), a per-line quantity, not a fraction of the total address space.

WATCH OUT FOR

Computing only the tag width per line (6 bits) and stopping there, or matching it to an option like $2^{12}/2^{14}$ by confusing "bits per tag" with "total tag storage across the cache"; also miscounting the number of lines by forgetting to divide cache size by block size before comparing to the address bits.

Q.37 MCQ · 2 MARKS

Computer Organization and Architecture › Instructions & addressing

A processor has 64 general-purpose registers and 50 distinct instruction types. An instruction is encoded in 32-bits. What is the maximum number of bits that can be used to store the immediate operand for the given instruction?

```
ADD R1, #25    // R1 = R1 + 25
```

(A) 16

(B) 20

(C) 22

(D) 24

Official answer: **B**

SOLUTION

1. Opcode field: 50 instruction types need ceil(log2(50)) bits. Since 2^5 = 32 < 50 ≤ 2^6 = 64, the opcode requires 6 bits.
2. Register field width: addressing 64 general-purpose registers requires log2(64) = 6 bits per register operand.
3. Register operand count: in `ADD R1, #25` (i.e. R1 = R1 + 25), R1 is used as both the source and the destination, so only one register operand needs to be encoded: one 6-bit register field, not two.
4. Field budget: total instruction length is fixed at 32 bits, split into opcode + one register field + immediate field.
5. Bits used by opcode and register field: 6 + 6 = 12 bits.
6. Remaining bits available for the immediate operand: 32 − 12 = 20 bits.
7. This is the maximum size of the immediate field, matching option (B) 20.

KEY IDEA

Fixed-length instruction encoding: bits for opcode (enough to distinguish all instruction types) plus bits for each explicitly encoded register operand are subtracted from the total instruction width; whatever remains is available for the immediate field.

WATCH OUT FOR

Assuming the instruction needs two 6-bit register fields (source and destination) as in a generic three-address or two-address format, which would leave only 14 bits for the immediate and match no option. The given instruction `ADD R1, #25` uses R1 as both source and destination, so only one register field is encoded.

Q.38 MCQ · 2 MARKS

Operating Systems › CPU scheduling

A computer has two processors, M_1 and M_2. Four processes P_1, P_2, P_3, P_4 with CPU bursts of 20, 16, 25, and 10 milliseconds, respectively, arrive at the same time and these are the only processes in the system. The scheduler uses non-preemptive priority scheduling, with priorities decided as follows:
- M_1 uses priority of execution for the processes as, $P_1 > P_3 > P_2 > P_4$, i.e., P_1 and P_4 have highest and lowest priorities, respectively.
- M_2 uses priority of execution for the processes as, $P_2 > P_3 > P_4 > P_1$, i.e., P_2 and P_1 have highest and lowest priorities, respectively.

A process P_i is scheduled to a processor M_k, if the processor is free and no other process P_j is waiting with higher priority. At any given point of time, a process can be allocated to any one of the free processors without violating the execution priority rules. Ignore the context switch time. What will be the average waiting time of the processes in milliseconds?

(A) 9.00

(B) 8.75

(C) 6.50

(D) 7.50

Official answer: **A**

SOLUTION

1. All four processes arrive at $t = 0$ and both processors are free. Each free processor takes the waiting process that ranks highest in its own priority order, and no process is preempted once started.
2. At $t = 0$: M_1 ranks P_1 first, so P_1 runs on M_1 during $[0, 20]$. M_2 ranks P_2 first, so P_2 runs on M_2 during $[0, 16]$.
3. At $t = 16$: M_2 becomes free. P_3 and P_4 are waiting, and M_2 ranks $P_3 > P_4$, so P_3 runs on M_2 during $[16, 41]$. M_1 is still busy, so M_2 is the only free processor.
4. At $t = 20$: M_1 becomes free. Only P_4 is waiting, so P_4 runs on M_1 during $[20, 30]$.
5. Timeline: M_1: $P_1[0, 20]$, $P_4[20, 30]$. M_2: $P_2[0, 16]$, $P_3[16, 41]$.
6. All arrivals are at 0, so waiting time = start time:

Process	Start	Burst	Waiting
P_1	0	20	0
P_2	0	16	0
P_3	16	25	16
P_4	20	10	20

7. Average waiting time $= \dfrac{0 + 0 + 16 + 20}{4} = \dfrac{36}{4} = 9.00$ ms, which is option (A).

KEY IDEA
With non-preemptive scheduling on two processors, each processor that becomes free picks the waiting process that is highest in its own priority list. Simulate the events in time order.

WATCH OUT FOR
Using one global priority order, or letting P_4 run on M_2 at $t = 16$ before P_3. M_2 ranks P_3 above P_4, so P_3 goes first and P_4 waits until M_1 frees at $t = 20$.

Q.39 MCQ · 2 MARKS

Database Management and Warehousing › Relational algebra

Consider two relations describing *teams* and *players* in a sports league:

- $teams(tid, tname)$: tid, $tname$ are team-id and team-name, respectively
- $players(pid, pname, tid)$: pid, $pname$, and tid denote player-id, player-name and the team-id of the player, respectively

Which ONE of the following tuple relational calculus queries returns the name of the players who play for the team having $tname$ as 'MI'?

(A) $\{p.pname \mid p \in players \wedge \exists t\,(t \in teams \wedge p.tid = t.tid \wedge t.tname = \text{'MI'})\}$

(B) $\{p.pname \mid p \in teams \wedge \exists t\,(t \in players \wedge p.tid = t.tid \wedge t.tname = \text{'MI'})\}$

(C) $\{p.pname \mid p \in players \wedge \exists t\,(t \in teams \wedge t.tname = \text{'MI'})\}$

(D) $\{p.pname \mid p \in teams \wedge \exists t\,(t \in players \wedge t.tname = \text{'MI'})\}$

Official answer: **A**

SOLUTION

1. We need the names of players whose team is 'MI'. This needs a join of $players$ and $teams$ on tid, followed by a selection on $tname$.
2. The output is $p.pname$, and $pname$ is an attribute only of $players$. So p must range over $players$, that is $p \in players$. This rules out (B) and (D), where $p \in teams$ and $p.pname$ is undefined.
3. Between (A) and (C): in both, the existential variable t ranges over $teams$, but only (A) has the join condition $p.tid = t.tid$.
4. (C) has no join condition. If any team named 'MI' exists, it returns every player in the league, whatever team the player is on.
5. (A) gives $\pi_{\text{pname}}(players \bowtie_{players.tid=teams.tid} \sigma_{\text{tname}='MI'}(teams))$, which is exactly the required query. The answer is (A).

KEY IDEA
In tuple relational calculus, the free tuple variable must range over the relation that holds the output attribute. The link between relations must be written as an explicit join condition on the shared key, here $p.tid = t.tid$. The selection condition goes in the same existential clause.

WATCH OUT FOR
Choosing (C). It looks right because it ranges over $players$ and checks that a team named 'MI' exists. It never ties p to that team, so it returns every player whenever any team is named 'MI'.

Q.40 MCQ · 2 MARKS

Computer Networks › IPv4

Q.40 A packet with the destination IP address 145.36.109.70 arrives at a router whose routing table is shown. Which interface will the packet be forwarded to?

Subnet Address	Subnet Mask (in CIDR notation)	Interface
145.36.0.0	/16	E1
145.36.128.0	/17	E2
145.36.64.0	/18	E3
145.36.255.0	/24	E4
Default	--	E5

(A) E3

(B) E1

(C) E2

(D) E5

Official answer: **A**

SOLUTION

1. A router forwards by longest-prefix match: find every entry whose prefix matches the destination, then take the match with the longest mask. If nothing else matches, the default route applies.
2. All four non-default entries begin with 145.36, which is also the first two octets of the destination 145.36.109.70. Only the third octet separates them. The destination's third octet is $109 = 01101101_2$.
3.

Entry	Third-octet range	Contains 109?	Interface
$145.36.0.0/16$	0 to 255	yes	E1
$145.36.128.0/17$	128 to 255	no	E2
$145.36.64.0/18$	64 to 127	yes	E3
$145.36.255.0/24$	255 only	no	E4
Default	all	yes	E5

4. Why the /18 range is 64 to 127: the first 18 bits are the 16 bits of 145.36 plus the top 2 bits of the third octet. The entry's third octet is $64 = 01000000_2$, so its top two bits are 01. The remaining 6 bits are free, giving 01000000_2 to 01111111_2, which is 64 to 127. Since $109 = 01101101_2$ also starts with 01, it matches.
5. Likewise /17 needs the top bit of the third octet to be 1 (128 to 255), but 109 starts with 0, so it does not match. /24 needs the third octet to equal 255, so it does not match either.

6. The matching entries are /16 (E1), /18 (E3) and the default (/0, E5). The longest prefix is /18, so the packet is forwarded to E3, which is option (A).

KEY IDEA
Use longest-prefix match: list every entry whose prefix covers the destination, then choose the one with the longest mask. Here only the third octet decides, and /18 covers third octets 64 to 127.

WATCH OUT FOR
Picking the first matching row (/16, E1) instead of the most specific one, or forgetting that a /18 starting at 64 stops at 127 and so does cover 109.

Q.41 MCQ · 2 MARKS

Engineering Mathematics › Linear algebra

Let A be a 2×2 matrix as given.

$$A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

What are the eigenvalues of the matrix A^{13} ?

(A) $1,\ -1$

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

(C) $4\sqrt{2},\ -4\sqrt{2}$

(D) $64\sqrt{2},\ -64\sqrt{2}$

Official answer: **D**

SOLUTION

1. Characteristic polynomial of $A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$: $\det(A - \lambda I) = (1-\lambda)(-1-\lambda) - 1 = \lambda^2 - 1 - 1 = \lambda^2 - 2$.
2. So the eigenvalues of A are $\lambda = \sqrt{2}$ and $\lambda = -\sqrt{2}$. Check: trace $= 0 = \sqrt{2} + (-\sqrt{2})$ and determinant $= -2 = \sqrt{2} \cdot (-\sqrt{2})$.
3. If $Av = \lambda v$, then $A^k v = \lambda^k v$ by repeated multiplication. So λ^k is an eigenvalue of A^k. Since A is 2×2 with two distinct eigenvalues, A^k has exactly the two eigenvalues λ_1^k and λ_2^k.
4. $(\sqrt{2})^{13} = 2^{13/2} = 2^6 \cdot 2^{1/2} = 64\sqrt{2}$.
5. $(-\sqrt{2})^{13} = -(\sqrt{2})^{13} = -64\sqrt{2}$, because 13 is odd.
6. The eigenvalues of A^{13} are $64\sqrt{2}$ and $-64\sqrt{2}$, which is option (D).

KEY IDEA
The eigenvalues of A^k are the k-th powers of the eigenvalues of A. Find the eigenvalues of A from $\lambda^2 - \text{trace} \cdot \lambda + \det = \lambda^2 - 2$, then raise them to the 13th power.

WATCH OUT FOR
Multiplying the matrix out or taking $13 \cdot \sqrt{2}$ instead of $(\sqrt{2})^{13}$. Another slip is writing $2^{13/2}$ as 2^6 and dropping the leftover $\sqrt{2}$, which gives 64 instead of $64\sqrt{2}$. The negative eigenvalue keeps its sign because 13 is odd.

Q.42 MCQ · 2 MARKS

Digital Electronics › Boolean logic

Consider the following four variable Boolean function in sum-of-product form

$F(b_3, b_2, b_1, b_0) = \sum(0, 2, 4, 8, 10, 11, 12)$.

where the value of the function is computed by considering $b_3 b_2 b_1 b_0$ as a 4-bit binary number, where b_3 denotes the most significant bit and b_0 denotes the least significant bit. Note that there are no don't care terms. Which ONE of the following options is the CORRECT minimized Boolean expression for F?

(A) $\overline{b_1}\,\overline{b_0} + \overline{b_2}\,\overline{b_0} + b_1\overline{b_2}b_3$

(B) $\overline{b_1}\,\overline{b_0} + \overline{b_2}\,\overline{b_0}$

(C) $\overline{b_2}\,\overline{b_0} + b_1 b_2 b_3$

(D) $\overline{b_0}\,\overline{b_2} + \overline{b_3}$

Official answer: **A**

SOLUTION

1. Write the minterms in binary ($b_3 b_2 b_1 b_0$): $0 = 0000$, $2 = 0010$, $4 = 0100$, $8 = 1000$, $10 = 1010$, $11 = 1011$, $12 = 1100$.
2. Karnaugh map, rows $b_3 b_2$ and columns $b_1 b_0$ in Gray order (00, 01, 11, 10):

$b_3 b_2 \backslash b_1 b_0$	00	01	11	10
00	1	0	0	1
01	1	0	0	0
11	1	0	0	0
10	1	0	1	1

3. The quad $\{0, 2, 8, 10\}$ has $b_2 = 0$ and $b_0 = 0$ in every cell, so it gives the prime implicant $\overline{b_2}\,\overline{b_0}$.
4. The quad $\{0, 4, 8, 12\}$ has $b_1 = 0$ and $b_0 = 0$ in every cell, so it gives the prime implicant $\overline{b_1}\,\overline{b_0}$.
5. Minterm 11 (1011) is not covered yet. Its neighbours differ in one bit: 1010 (10), 1001 (9), 1111 (15), 0011 (3). Only 10 is a minterm, so the pair $\{10, 11\}$ is the only grouping for it. It gives $b_3 \overline{b_2} b_1$, which is essential.
6. Minterms 4 and 12 are covered only by the quad $\{0, 4, 8, 12\}$, and 2 only by $\{0, 2, 8, 10\}$. Both quads are therefore essential as well. Minimal cover: $F = \overline{b_1}\,\overline{b_0} + \overline{b_2}\,\overline{b_0} + b_1 \overline{b_2} b_3$. It covers $\{0, 4, 8, 12\} \cup \{0, 2, 8, 10\} \cup \{10, 11\}$, which is all 7 minterms and nothing else. This is option A.
7. Check the other options. B misses minterm 11, because $b_0 = 1$ makes both of its terms 0. C's term $b_1 b_2 b_3$ is true only for 1110 and 1111, which are not minterms, and C leaves 4, 12 and 11 uncovered. D contains $\overline{b_3}$, which is true for 0 to 7, so it is true at the non-minterms 1, 3, 5, 6, 7, and it misses 11 and 12.

KEY IDEA

Draw the K-map and take the essential prime implicants: the quads $\{0, 4, 8, 12\}$ and $\{0, 2, 8, 10\}$, plus the pair $\{10, 11\}$ for the one minterm (11) that no quad reaches.

WATCH OUT FOR

Stopping after the two quads and choosing option B, which leaves minterm 11 uncovered. Another slip is accepting a term that is true at a non-minterm, such as $\overline{b_3}$ in option D.

Q.43 MCQ · 2 MARKS

Data Structures and Algorithms › Graph algorithms

Let $G(V, E)$ be an undirected and unweighted graph with 100 vertices. Let $d(u, v)$ denote the number of edges in a shortest path between vertices u and v in V. Let the maximum value of $d(u, v)$, $u, v \in V$ such that $u \neq v$, be 30. Let T be any breadth-first-search tree of G. Which ONE of the given options is CORRECT for every such graph G?

(A) The height of T is exactly 15.

(B) The height of T is exactly 30.

(C) The height of T is at least 15.

(D) The height of T is at least 30.

Official answer: **C**

SOLUTION

1. Let $\mathrm{diam}(G) = \max_{u,v} d(u, v) = 30$, achieved by some pair of vertices (x, y).
2. For any root r, the breadth-first-search tree T rooted at r has height $h(T) = \max_v d(r, v) = \mathrm{ecc}(r)$, the eccentricity of r: BFS discovers vertices in strictly increasing order of distance from r, so the deepest tree level is exactly the farthest distance from r.

3. By the triangle inequality on shortest-path distances: $d(x, y) \le d(x, r) + d(r, y)$.
4. Since r is any vertex, both $d(x, r) \le \mathrm{ecc}(r) = h(T)$ and $d(r, y) \le \mathrm{ecc}(r) = h(T)$, so $d(x, y) \le 2\,h(T)$.
5. Substituting $d(x, y) = 30$: $30 \le 2\,h(T)$, hence $h(T) \ge 15$, and this holds for every root r, so for every BFS tree of G, and for every graph G satisfying the given diameter condition.
6. Checking the bound is not improvable to equality or to 30 for every case: choosing r to be a 'central' vertex (minimizing eccentricity) can make $h(T) = 15$ exactly, which already rules out option (D) ('at least 30') since $15 < 30$. Choosing r to be a diametral endpoint gives $h(T) = 30$, which rules out option (A) ('exactly 15') and option (B) ('exactly 30'), since different roots on the same graph give different heights (15 vs 30).
7. Only option (C), $h(T) \ge 15$, is a statement that holds for every such graph G and every BFS tree T of it.

KEY IDEA
See keyIdea field above.

WATCH OUT FOR
See commonMistake field above.

Q.44 MCQ · 2 MARKS

Theory of Computation › Regular languages

Consider the following two languages over the alphabet $\{a, b\}$:

$L_1 = \{\alpha\beta\alpha \mid \alpha \in \{a, b\}^+ \text{ AND } \beta \in \{a, b\}^+\}$

$L_2 = \{\alpha\beta\alpha \mid \alpha \in \{a\}^+ \text{ AND } \beta \in \{a, b\}^+\}$

Which ONE of the following statements is CORRECT?

(A) Both L_1 and L_2 are regular languages.

(B) L_1 is a regular language but L_2 is not a regular language.

(C) L_1 is not a regular language but L_2 is a regular language.

(D) Neither L_1 nor L_2 is a regular language.

Official answer: **C**

SOLUTION

1. L_2 **is regular.** Every string of L_2 has the form $a^i \beta a^i$ with $i \ge 1$ and $\beta \ne \varepsilon$, so it starts with a, ends with a and has length at least 3. Conversely, any string $a\,x\,a$ with $x \in \{a, b\}^+$ is in L_2: take $\alpha = a$ and $\beta = x$. Hence $L_2 = a(a + b)^+ a$, which is regular.
2. L_1 **is not regular.** A string w is in L_1 exactly when it has a nonempty border α (a string that is both a prefix and a suffix of w) with $2|\alpha| < |w|$, so that a nonempty β fits between the two copies. We use the Myhill-Nerode theorem on the strings $x_i = a^i b$ for $i \ge 1$.
3. Take $1 \le j < i$ and the suffix $z = b\,a^j b$. Then $x_j z = a^j b\,b\,a^j b = \alpha\beta\alpha$ with $\alpha = a^j b$ and $\beta = b$, so $x_j z \in L_1$.
4. Now consider $x_i z = a^i b\,b\,a^j b$, of length $i + j + 3$. It starts with a, so any border starts with a and ends with b. A suffix starting with a is either $a^k b$ with $1 \le k \le j$ (taken from the last block), or $a^m b\,b\,a^j b$ with $1 \le m \le i$ (taken from the first block).
5. If the border is $a^k b$, the prefix of the same length must be $a^i b$ (the word begins $a^i b$), which forces $k = i$. But $k \le j < i$, a contradiction. If the border is $a^m b\,b\,a^j b$, its prefix starts with $a^m b$, which forces $m = i$, and then the border is the whole word, which is not a proper border. So $x_i z$ has no valid border and $x_i z \notin L_1$.
6. Thus for every pair $j < i$ the suffix $b\,a^j b$ separates x_j from x_i. The strings $a^i b$ ($i \ge 1$) are pairwise distinguishable, so L_1 has infinitely many Myhill-Nerode classes and is not regular.
7. Therefore L_1 is not regular and L_2 is regular, which is option (C).

KEY IDEA

L_2 collapses to the regular language $a(a+b)^+a$ because the single letter a can always serve as α. In L_1, α can be any nonempty word, so the language demands a long matching border, and Myhill-Nerode on a^ib shows it is not regular.

WATCH OUT FOR

Seeing the same α at both ends and assuming both languages need unbounded memory, or assuming both are regular because the first and last letters match. In L_1 a border can start and end with different letters, for example $abbab$ with $\alpha = ab$, so L_1 is not just "first letter equals last letter". Also, the separating suffix must be chosen from the smaller index: $b\,a^ib$ fails for $j < i$ because a^jb is then also a border of x_jz_i.

Q.45 MCQ · 2 MARKS

Theory of Computation › Context-free languages

Consider the following two languages over the alphabet $\{a, b, c\}$, where m and n are natural numbers.

$$L_1 = \{a^m b^m c^{m+n} \mid m, n \geq 1\}$$

$$L_2 = \{a^m b^n c^{m+n} \mid m, n \geq 1\}$$

Which ONE of the following statements is CORRECT?

(A) Both L_1 and L_2 are context-free languages.

(B) L_1 is a context-free language but L_2 is not a context-free language.

(C) L_1 is not a context-free language but L_2 is a context-free language.

(D) Neither L_1 nor L_2 are context-free languages.

Official answer: **C**

Q.46 MSQ · 2 MARKS

Compiler Design › Parsing

Which of the following statement(s) is/are TRUE while computing First and Follow during top down parsing by a compiler?

(A) For a production $A \rightarrow \epsilon$, ϵ will be added to $\mathrm{First}(A)$.

(B) If there is any input right end marker, it will be added to $\mathrm{First}(S)$, where S is the start symbol.

(C) For a production $A \rightarrow \epsilon$, ϵ will be added to $\mathrm{Follow}(A)$.

(D) If there is any input right end marker, it will be added to $\mathrm{Follow}(S)$, where S is the start symbol.

Official answer: **A, D**

SOLUTION

1. $\mathrm{FIRST}(A)$ holds the terminals that can begin a string derived from A, plus ε if $A \Rightarrow^* \varepsilon$.
2. (A) With the production $A \rightarrow \varepsilon$, A derives the empty string, so $\varepsilon \in \mathrm{FIRST}(A)$. TRUE.
3. (B) The end marker \$ is not a grammar symbol, so it is never the first symbol of a string derived from S and never enters any FIRST set. FALSE.
4. (C) $\mathrm{FOLLOW}(A)$ holds the terminals that can come right after A, plus the end marker. ε is not a terminal and never appears in a FOLLOW set. FALSE.
5. (D) The first rule for FOLLOW is: put \$ in $\mathrm{FOLLOW}(S)$ for the start symbol S, because the input ends right after the whole string derived from S. TRUE.
6. Correct options: A and D.

KEY IDEA
ε belongs only in FIRST sets (when a symbol can derive the empty string); the end marker `$` belongs only in FOLLOW sets, starting with $\text{FOLLOW}(S)$ for the start symbol.

WATCH OUT FOR
Mixing up the two sets: adding ε to a FOLLOW set because of $A \to \varepsilon$, or adding the end marker `$` to $\text{FIRST}(S)$.

Q.47 MSQ · 2 MARKS

Database Management and Warehousing › Normal forms

Consider a relational schema `team(name, city, owner)`, with functional dependencies $\{\text{name} \to \text{city},\ \text{name} \to \text{owner}\}$.

The relation `team` is decomposed into two relations, `t1(name, city)` and `t2(name, owner)`. Which of the following statement(s) is/are TRUE?

(A) The relation team is NOT in BCNF.

(B) The relations t1 and t2 are in BCNF.

(C) The decomposition constitutes a lossless join.

(D) The relation team is NOT in 3NF.

Official answer: **B, C**

SOLUTION

1. Attribute closure: $\text{name}^+ = \{\text{name}, \text{city}, \text{owner}\}$, which is all of `team`. So `name` is a candidate key. It is the only one, because `city` and `owner` determine nothing.
2. BCNF test on `team`: both FDs, $\text{name} \to \text{city}$ and $\text{name} \to \text{owner}$, have a superkey on the left. So `team` is in BCNF, and (A) is false.
3. Every relation in BCNF is also in 3NF, so `team` is in 3NF and (D) is false.
4. Projected FDs: `t1(name, city)` holds only $\text{name} \to \text{city}$, and `name` is its key. `t2(name, owner)` holds only $\text{name} \to \text{owner}$, and `name` is its key. Every FD has a superkey on its left in both relations, so both are in BCNF and (B) is true.
5. Lossless-join test for a binary decomposition: $R_1 \cap R_2 = \{\text{name}\}$. Since name^+ contains all of R_1 (and of R_2), $R_1 \cap R_2 \to R_1$ holds. The decomposition is lossless, and (C) is true.
6. True statements: (B) and (C).

KEY IDEA
Find the key by attribute closure. Here name is the only key, so every FD has a superkey on its left and team is in BCNF. A binary decomposition is lossless when the common attributes functionally determine one of the two relations. Here name determines both.

WATCH OUT FOR
Thinking that team must be un-normalised because it was decomposed. Decomposing an already-BCNF relation is still allowed, and here it is lossless. Another slip is to claim a relation is not in 3NF without checking that the left side of each FD is a superkey.

Q.48 MSQ · 2 MARKS

Discrete Mathematics › Mathematical logic

Which of the following predicate logic formulae/formula is/are CORRECT representation(s) of the statement: "Everyone has exactly one mother"?

The meanings of the predicates used are:
- $mother(y, x)$: y is the mother of x
- $noteq(x, y)$: x and y are not equal

(A) $\forall x \exists y \exists z (mother(y, x) \wedge \neg mother(z, x))$

(B) $\forall x \exists y [mother(y, x) \wedge \forall z (noteq(z, y) \to \neg mother(z, x))]$

(C) $\forall x \forall y[mother(y,x) \rightarrow \exists z(mother(z,x) \wedge \neg noteq(z,y))]$

(D) $\forall x \exists y[mother(y,x) \wedge \neg \exists z(noteq(z,y) \wedge mother(z,x))]$

Official answer: **B, D**

SOLUTION

1. "Exactly one mother" for every x means two things: existence (some y is a mother of x) and uniqueness (nobody other than y is a mother of x). Target form: $\forall x \exists y\,[mother(y,x) \wedge \forall z\,(noteq(z,y) \rightarrow \neg mother(z,x))]$.
2. (A) $\forall x \exists y \exists z\,(mother(y,x) \wedge \neg mother(z,x))$ says only that x has some mother y and that some z is not a mother of x. The second part is true in any world with at least two people, even if x has ten mothers (take y as one mother and z as a non-mother). Uniqueness is not captured, so (A) is incorrect.
3. (B) $\forall x \exists y\,[mother(y,x) \wedge \forall z\,(noteq(z,y) \rightarrow \neg mother(z,x))]$ is exactly the target form: y is a mother of x, and every z different from y is not a mother of x. Correct.
4. (C) $\forall x \forall y\,[mother(y,x) \rightarrow \exists z\,(mother(z,x) \wedge \neg noteq(z,y))]$. Here $\neg noteq(z,y)$ means $z = y$, so the consequent is $\exists z\,(mother(z,x) \wedge z = y)$, which is $mother(y,x)$. The formula reduces to $\forall x \forall y\,[mother(y,x) \rightarrow mother(y,x)]$, a tautology. A person with no mother satisfies it vacuously, and so does a person with two mothers. It does not express "exactly one", so (C) is incorrect.
5. (D) $\forall x \exists y\,[mother(y,x) \wedge \neg \exists z\,(noteq(z,y) \wedge mother(z,x))]$. Use $\neg \exists z\, P \equiv \forall z \neg P$, then $\neg(a \wedge b) \equiv (a \rightarrow \neg b)$. This gives $\forall z\,(noteq(z,y) \rightarrow \neg mother(z,x))$, which is the same as the inner part of (B). So (D) is equivalent to (B). Correct.
6. Hence the correct options are B and D.

KEY IDEA

"Exactly one" means existence plus uniqueness: some y is the mother, and every z different from y is not. Writing uniqueness as $\forall z\,(z \neq y \rightarrow \neg P(z))$ or as $\neg \exists z\,(z \neq y \wedge P(z))$ gives equivalent formulas.

WATCH OUT FOR

Choosing (A) because it looks like "one mother and others are not mothers". Its $\neg mother(z,x)$ only says some z is not a mother, which does not limit the number of mothers. Another slip is reading $\neg noteq(z,y)$ in (C) as a real restriction, when it just means $z = y$ and makes the formula a tautology.

Q.49 MSQ · 2 MARKS

Discrete Mathematics › Groups

$A = \{0, 1, 2, 3, \dots\}$ is the set of non-negative integers. Let F be the set of functions from A to itself. For any two functions $f_1, f_2 \in F$, we define

$$(f_1 \odot f_2)(n) = f_1(n) + f_2(n)$$

for every number n in A. Which of the following is/are CORRECT about the mathematical structure $(F, \odot)$?

(A) $(F, \odot)$ is an Abelian group.

(B) $(F, \odot)$ is an Abelian monoid.

(C) $(F, \odot)$ is a non-Abelian group.

(D) $(F, \odot)$ is a non-Abelian monoid.

Official answer: **B**

SOLUTION

1. Closure: for $f_1, f_2 \in F$, the value $f_1(n) + f_2(n)$ is a sum of two non-negative integers, hence a non-negative integer, so $f_1 \odot f_2 \in F$.
2. Associativity: $((f_1 \odot f_2) \odot f_3)(n) = (f_1(n) + f_2(n)) + f_3(n) = f_1(n) + (f_2(n) + f_3(n)) = (f_1 \odot (f_2 \odot f_3))(n)$ for every n, because addition of integers is associative.

3. Identity: the zero function $e(n) = 0$ lies in F and $(f \odot e)(n) = f(n) + 0 = f(n) = (e \odot f)(n)$. So $(F, \odot)$ is a monoid.
4. Commutativity: $f_1(n) + f_2(n) = f_2(n) + f_1(n)$ for every n, so $f_1 \odot f_2 = f_2 \odot f_1$ and the monoid is Abelian. This rules out (C) and (D), which both call the structure non-Abelian.
5. Inverses: an inverse of f would be a function $g \in F$ with $f(n) + g(n) = 0$ for all n, i.e. $g(n) = -f(n)$. Since $g(n)$ must be non-negative, this is possible only when $f(n) = 0$ for every n.
6. Take $f(n) = 1$ for all n. Then $g(n) = -1 \notin A$, so f has no inverse in F and $(F, \odot)$ is not a group. This rules out (A).
7. So $(F, \odot)$ is an Abelian monoid that is not a group, and only (B) is correct.

KEY IDEA

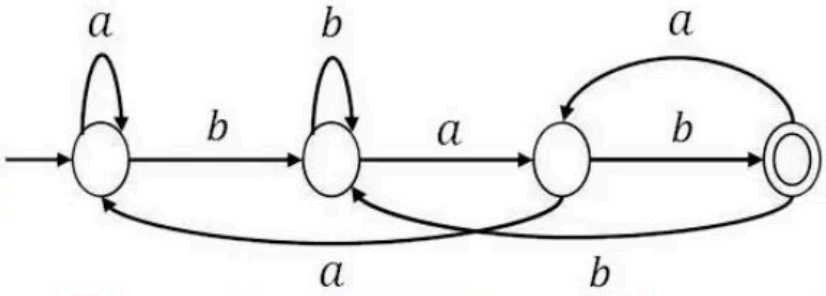
Pointwise addition of functions from the non-negative integers to themselves is closed, associative, commutative and has the zero function as identity. Inverses would need negative values, so it is a commutative monoid and not a group.

WATCH OUT FOR

Treating $(F, \odot)$ as an Abelian group because addition of integers has inverses. The inverse $-f(n)$ is not a non-negative integer, so it lies outside F for any non-zero f.

Q.50 MSQ · 2 MARKS

Theory of Computation › Finite automata

Q.50	Consider the following deterministic finite automaton (DFA) defined over the alphabet, $\Sigma = \{a, b\}$. Identify which of the following language(s) is/are accepted by the given DFA.
	
(A)	The set of all strings containing an even number of b's.
(B)	The set of all strings containing the pattern bab.
(C)	The set of all strings ending with the pattern bab.
(D)	The set of all strings not containing the pattern aba.

Official answer: **C**

SOLUTION

1. Read the DFA from the figure. States q_0 (start), q_1, q_2, q_3; q_3 is the only accepting state. Transitions: $\delta(q_0, a) = q_0$, $\delta(q_0, b) = q_1$, $\delta(q_1, a) = q_2$, $\delta(q_1, b) = q_1$, $\delta(q_2, a) = q_0$, $\delta(q_2, b) = q_3$, $\delta(q_3, a) = q_2$,

$\delta(q_3, b) = q_1$.

2. Give each state a meaning: the longest suffix of the input read so far that is a prefix of bab. So q_0 means no useful suffix (ε), q_1 means b, q_2 means ba, and q_3 means bab.
3. Check every transition against this meaning. From q_0: reading a gives suffix ε, so q_0; reading b gives b, so q_1. From q_1 (b): reading a gives ba, so q_2; reading b gives bb, longest useful suffix b, so q_1. From q_2 (ba): reading a gives baa, no useful suffix, so q_0; reading b gives bab, so q_3. From q_3 (bab): reading a gives $baba$, longest useful suffix ba, so q_2; reading b gives $babb$, longest useful suffix b, so q_1. All eight transitions match, so the DFA is in q_3 exactly when the input ends with bab. The language is $(a+b)^*bab$, and (C) is correct.
4. (A) is false. The string $bbab$ ends with bab, so it is accepted ($q_0 \to q_1 \to q_1 \to q_2 \to q_3$), but it has three b's, an odd number. Also bb has an even number of b's but ends in q_1 and is rejected.
5. (B) is false. The string $baba$ contains bab, but $q_0 \to q_1 \to q_2 \to q_3 \to q_2$ ends in the non-accepting state q_2, so it is rejected.
6. (D) is false. The string a does not contain aba, but it ends in q_0 and is rejected. Also $abab$ contains aba yet is accepted ($q_0 \to q_0 \to q_1 \to q_2 \to q_3$).

KEY IDEA

Label each state with the longest suffix of the input that is a prefix of the pattern bab, and check every transition. The DFA is then the standard pattern-matching automaton for strings ending in bab.

WATCH OUT FOR

Choosing (B) because the automaton is a bab-detector. Here the accepting state q_3 has outgoing transitions to q_2 and q_1, so it is not absorbing and the DFA remembers only the last few symbols. Accepting means the input ENDS with bab, not that it contains bab.

Q.51 NAT · 2 MARKS

Operating Systems › File systems & disks

A disk of size 512M bytes is divided into blocks of 64K bytes. A file is stored in the disk using linked allocation. In linked allocation, each data block reserves 4 bytes to store the pointer to the next data block. The link part of the last data block contains a NULL pointer (also of 4 bytes). Suppose a file of 1M bytes needs to be stored in the disk. Assume, 1K = 2^10 and 1M = 2^20. The amount of space in bytes that will be wasted due to internal fragmentation is _____. (Answer in integer)

Official answer: **65468**

SOLUTION

1. Block size $= 64\text{K} = 2^{16} = 65536$ bytes. Each block reserves 4 bytes for the pointer (the last block holds a 4-byte NULL pointer), so each block holds $65536 - 4 = 65532$ bytes of file data.
2. File size $= 1\text{M} = 2^{20} = 1048576$ bytes.
3. Blocks needed $= \left\lceil \dfrac{1048576}{65532} \right\rceil$. Since $16 \times 65532 = 1048512 < 1048576$, 16 blocks are not enough, so 17 blocks are needed.
4. The first 16 blocks hold 1048512 bytes of data, so the last block holds only $1048576 - 1048512 = 64$ bytes of data.
5. Internal fragmentation is the unused data area of the last block: $65532 - 64 = 65468$ bytes. The 4-byte pointer field in the last block holds the NULL pointer, so it is used and not wasted.
6. Wasted space $= 65468$ bytes.

KEY IDEA

Linked allocation loses 4 bytes of every block to the pointer, so a block holds 65532 data bytes. Internal fragmentation is the unused data area in the last block only.

WATCH OUT FOR

Using the full 65536 bytes as the data capacity (giving 17 blocks and 0 or wrong waste). Another slip is counting the last block's NULL pointer as wasted, or dividing the file by 65536 to get exactly 16 blocks.

Q.52 NAT · 2 MARKS

Compiler Design › Optimisation

Q.52 Refer to the given 3-address code sequence. This code sequence is split into basic blocks. The number of basic blocks is _____. (Answer in integer)

```
1001: i = 1
1002: j = 1
1003: t1 = 10*i
1004: t2 = t1+j
1005: t3 = 8*t2
1006: t4 = t3-88
1007: a[t4] = 0.0
1008: j = j+1
1009: if j <= 10 goto 1003
1010: i = i+1
1011: if i <= 10 goto 1002
1012: i = 1
1013: t5 = i-1
1014: t6 = 88*t5
1015: a[t6] = 1.0
1016: i = i+1
1017: if i <= 10 goto 1013
```

Official answer: **6**

SOLUTION

1. A basic block is a maximal straight-line sequence of statements, entered only at its first statement and left only at its last. To split the code, find the leaders (the first statement of each block).
2. Rule 1: the first statement of the code is a leader, so 1001 is a leader.
3. Rule 2: the target of any jump is a leader. The jumps are 1009 to 1003, 1011 to 1002 and 1017 to 1013, so 1003, 1002 and 1013 are leaders.
4. Rule 3: the statement right after a jump is a leader. The statement after 1009 is 1010, and the statement after 1011 is 1012. The jump at 1017 is the last statement, so nothing follows it and it starts no block.
5. Leaders: $\{1001, 1002, 1003, 1010, 1012, 1013\}$. These are six distinct statements.

6. Each block runs from one leader up to the statement before the next leader: B_1 = 1001 ; B_2 = 1002 ; B_3 = 1003 to 1009 ; B_4 = 1010 to 1011 ; B_5 = 1012 ; B_6 = 1013 to 1017 .
7. The number of basic blocks equals the number of leaders, so there are 6 basic blocks.

KEY IDEA

Find the leaders: the first statement, every jump target, and every statement right after a jump. Each leader starts one basic block, so the number of blocks equals the number of leaders.

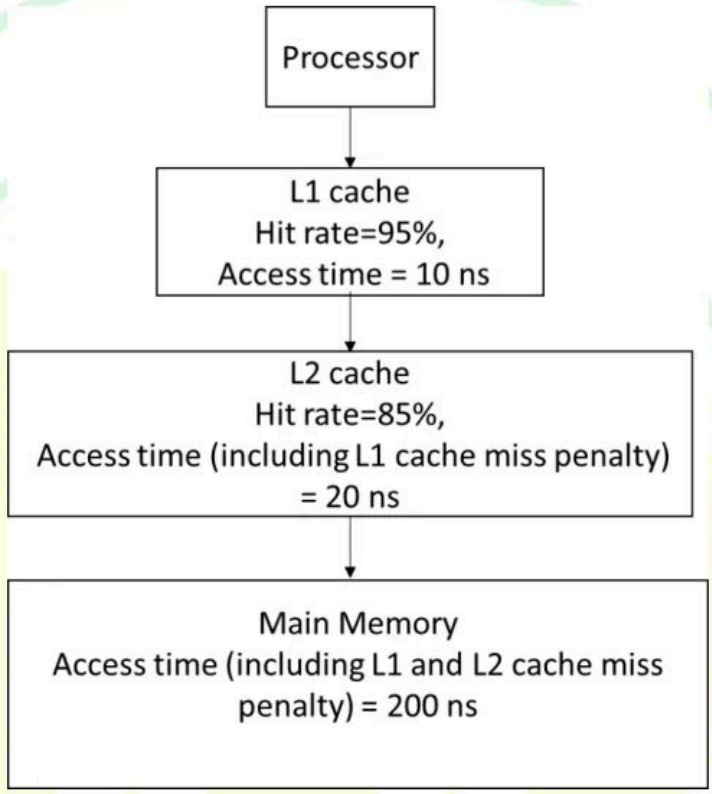
WATCH OUT FOR

Forgetting that a jump target in the middle of a loop nest, such as `1002`, starts its own block, which merges `1001` and `1002`. Another slip is counting only the jump targets and missing the statements after the jumps (`1010`, `1012`).

Q.53 NAT · 2 MARKS

Computer Organization and Architecture › Memory & cache

Q.53 A computer has a memory hierarchy consisting of two-level cache (L1 and L2) and a main memory. If the processor needs to access data from memory, it first looks into L1 cache. If the data is not found in L1 cache, it goes to L2 cache. If it fails to get the data from L2 cache, it goes to main memory, where the data is definitely available. *Hit rates* and *access times* of various memory units are shown in the figure. The *average memory access time* in nanoseconds (*ns*) is _____. (rounded off to two decimal places)



Official answer: **11.83 to 11.87**

SOLUTION

1. Identify the three mutually exclusive, exhaustive access outcomes from the figure: (i) L1 hit, (ii) L1 miss followed by L2 hit, (iii) L1 miss followed by L2 miss (data found in main memory).
2. P(L1 hit) = 0.95, with access time 10 ns (as stated for L1).
3. P(L1 miss and L2 hit) = 0.05 × 0.85 = 0.0425. Its access time is 20 ns, which the figure states already includes the L1 miss penalty, so this 20 ns is used directly without adding the L1 access time again.
4. P(L1 miss and L2 miss) = 0.05 × 0.15 = 0.0075. Its access time is 200 ns, which the figure states already includes both the L1 and L2 miss penalties, so this 200 ns is used directly.
5. Check: 0.95 + 0.0425 + 0.0075 = 1.0, confirming the three cases are exhaustive.

6. AMAT = 0.95 × 10 + 0.0425 × 20 + 0.0075 × 200 = 9.5 + 0.85 + 1.5 = 11.85 ns.

KEY IDEA
When a figure gives each level's access time as "including the miss penalty of the level(s) above," those numbers are already the total time incurred once that level is reached, so AMAT is simply the probability-weighted sum of the three mutually exclusive, exhaustive outcomes (L1 hit; L1 miss & L2 hit; L1 miss & L2 miss), using each stated time directly with no separate addition of the upstream access time.

WATCH OUT FOR
Adding a fresh L1-access-time term (10 ns) on top of the already-inclusive L2 and main-memory figures, e.g. computing 0.95×10 + 0.05×0.85×(10+20) + 0.05×0.15×(10+200), which double-counts the L1 access time, since the figure explicitly states the L2 and main-memory times already include the penalty of the level(s) above missing.

Q.54 NAT · 2 MARKS

Operating Systems › Virtual memory

In optimal page replacement algorithm, information about all future page references is available to the operating system (OS). A modification of the optimal page replacement algorithm is as follows:

The OS correctly predicts only up to next 4 page references (including the current page) at the time of allocating a frame to a page.

A process accesses the pages in the following order of page numbers: 1, 3, 2, 4, 2, 3, 1, 2, 4, 3, 1, 4.

If the system has three memory frames that are initially empty, the number of page faults that will occur during execution of the process is _____. (Answer in integer)

Official answer: **6**

SOLUTION

1. Number the references $1, 3, 2, 4, 2, 3, 1, 2, 4, 3, 1, 4$ as positions 1 to 12. When a fault occurs and all frames are full, look only at the next 4 references (the current one included). Evict the page whose next use in that window is farthest away, or which does not appear in the window at all.
2. Positions 1, 2 and 3 (pages $1, 3, 2$) fault into empty frames. That is 3 faults, and the frames hold $\{1, 3, 2\}$.
3. Position 4 (page 4) is a fault, making 4 faults. The window is $4, 2, 3, 1$, so the resident pages are next used at: page 2 at position 5, page 3 at position 6, page 1 at position 7. Page 1 is farthest, so evict it. The frames hold $\{3, 2, 4\}$.
4. Positions 5 and 6 (pages $2, 3$) are hits.
5. Position 7 (page 1) is a fault, making 5 faults. The window is $1, 2, 4, 3$, so the resident pages are next used at: page 2 at position 8, page 4 at position 9, page 3 at position 10. Page 3 is farthest, so evict it. The frames hold $\{2, 4, 1\}$.
6. Positions 8 and 9 (pages $2, 4$) are hits.
7. Position 10 (page 3) is a fault, making 6 faults. Only $3, 1, 4$ remain, so the window is $3, 1, 4$. Page 2 does not appear in the window, so evict it. The frames hold $\{4, 1, 3\}$.
8. Positions 11 and 12 (pages $1, 4$) are hits. The total is 6 page faults.

KEY IDEA
Simulate optimal replacement with a limited lookahead. On a fault with full frames, evict the page whose next use inside the next 4 references (current included) is farthest, and treat a page absent from that window as farthest of all.

WATCH OUT FOR
Using full future knowledge, as in ordinary optimal replacement, instead of only the 4-reference window. Another slip is forgetting that the window shrinks near the end of the string, and so miscounting which page to evict at position 10.

Q.55 NAT · 2 MARKS

Q.55 Consider the following database tables of a sports league.

```
player(pid,pname,age)          team(tid,tname,city,cid)
coach(cid,cname)               members(pid,tid)
```

An instance of the table and an SQL query are given.

player

pid	pname	age
1	Jasprit	31
2	Atharva	24
3	Ishan	26
4	Axar	30

coach

cid	cname
101	Ricky
102	Mark
103	Trevor

team

tid	tname	city	cid
10	MI	Mumbai	102
20	DC	Delhi	101
30	PK	Mohali	103

members

pid	tid
1	10
2	30
3	10
4	20

```
SELECT MIN(P.age)
FROM player P
WHERE P.pid IN (
    SELECT M.pid
    FROM team T, coach C, members M
    WHERE C.cname = 'Mark'
          AND T.cid = C.cid
          AND M.tid = T.tid
    )
```

The value returned by the given SQL query is _____ . (Answer in integer)

Official answer: **26**

SOLUTION

1. Run the inner subquery first. The condition C.cname = 'Mark' keeps only the coach row $(102, \text{Mark})$, so $C.cid = 102$.
2. Join with team on T.cid = C.cid . Only the team row $(10, \text{MI}, \text{Mumbai}, 102)$ matches, so $T.tid = 10$.
3. Join with members on M.tid = T.tid . The members rows with $tid = 10$ are $(1, 10)$ and $(3, 10)$.
4. The subquery therefore returns $M.pid \in \{1, 3\}$.
5. The outer query keeps the player rows with pid IN (1, 3) : Jasprit (age 31) and Ishan (age 26).
6. $$\text{MIN}(31, 26) = 26$$

KEY IDEA

Work out the IN subquery as a set of pids first (coach Mark, then his team, then the team's members). Then take the minimum age over only those players.

WATCH OUT FOR

Taking MIN over all players (24) or over Mark's team's whole roster by mistake. Another slip is to include the player from a different team, such as pid 2, who is on team 30 coached by Trevor. The subquery returns only pids 1 and 3.

Q.56 NAT · 2 MARKS

Engineering Mathematics › Probability & statistics

Suppose a 5-bit message is transmitted from a source to a destination through a noisy channel. The probability that a bit of the message gets flipped during transmission is 0.01. Flipping of each bit is independent of one another. The probability that the message is delivered error-free to the destination is _____ . (rounded off to three decimal places)

Official answer: **0.949 to 0.952**

SOLUTION

1. Each bit is delivered unflipped with probability $1 - 0.01 = 0.99$.
2. The message is error-free only if all 5 bits are unflipped. The flips are independent, so the probabilities multiply: $P = (0.99)^5$.
3. Compute: $0.99^2 = 0.9801$, $0.99^4 = 0.9801^2 = 0.96059601$, so $0.99^5 = 0.96059601 \times 0.99 = 0.9509900499$.
4. Check with the binomial form: $P(0 \text{ flips}) = \binom{5}{0}(0.01)^0(0.99)^5 = 0.99^5 \approx 0.95099$.
5. Rounded to three decimal places, $P \approx 0.951$.

KEY IDEA

With independent bit flips, "no error" means every bit survives, so multiply the per-bit survival probability 0.99 five times: 0.99^5.

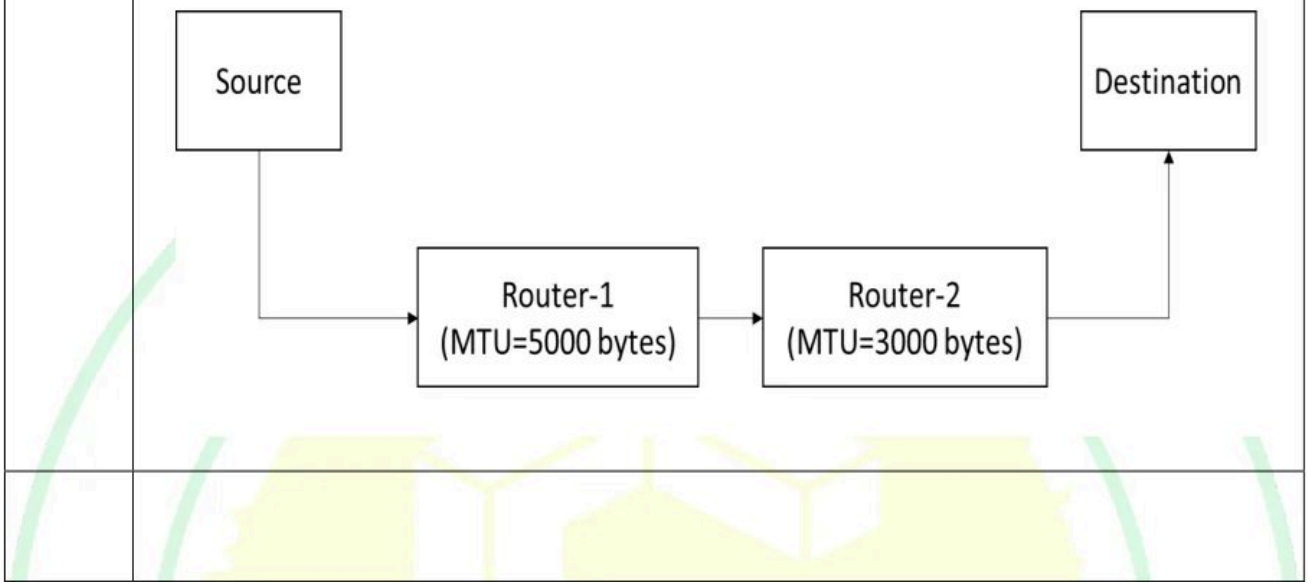
WATCH OUT FOR

Computing $1 - 5(0.01) = 0.95$ with a linear approximation, or reading the question as "exactly one flip". The linear form is only close, and the exact value $0.99^5 \approx 0.951$ is what the rounding asks for.

Q.57 NAT · 2 MARKS

Computer Networks › IPv4

Q.57 Suppose a message of size 15000 bytes is transmitted from a source to a destination using IPv4 protocol via two routers as shown in the figure. Each router has a defined maximum transmission unit (MTU) as shown in the figure, including IP header. The number of fragments that will be delivered to the destination is _____ . (Answer in integer)



Official answer: **7**

SOLUTION

1. Take the 15000 bytes as the IP payload, with a 20-byte IP header on every fragment. The source's own link MTU is not given, so the datagram reaches Router-1 whole.
2. Router-1 (MTU 5000, header included): the largest payload is $5000 - 20 = 4980$ bytes. Fragment offsets count 8-byte units, so every fragment except the last must carry a multiple of 8. The largest such size is $\lfloor 4980/8 \rfloor \times 8 = 4976$ bytes.
3. Router-1 output: $15000 = 3 \times 4976 + 72$, so the fragments carry 4976, 4976, 4976 and 72 bytes of payload. That is 4 fragments.
4. Router-2 (MTU 3000): the largest payload is $3000 - 20 = 2980$ bytes, rounded down to a multiple of 8, which is $\lfloor 2980/8 \rfloor \times 8 = 2976$ bytes.
5. Each 4976-byte fragment is split again: $4976 = 2976 + 2000$. This gives 2 fragments each, so the three of them give $3 \times 2 = 6$ fragments.
6. The 72-byte fragment already fits within 2980 bytes, so it stays as 1 fragment.
7. Fragments delivered to the destination: $6 + 1 = 7$. Router-2 fragments Router-1's fragments and does not reassemble them. Only the destination reassembles.

KEY IDEA

Fragment at each router with the largest payload that fits in the MTU after the 20-byte header and is a multiple of 8. Routers fragment but never reassemble, so Router-2 splits the fragments of Router-1 and only the destination reassembles.

WATCH OUT FOR

Fragmenting the original 15000 bytes directly at 2976 bytes (giving 6 fragments), or forgetting that the 72-byte fragment from Router-1 passes through Router-2 unchanged, or ignoring the multiple-of-8 rule (4980 and 2980 are not multiples of 8).

Q.58 NAT · 2 MARKS

Engineering Mathematics › Probability & statistics

Consider a probability distribution given by the density function $P(x)$.

$$P(x) = \begin{cases} Cx^2, & \text{for } 1 \le x \le 4 \\ 0, & \text{for } x < 1 \text{ or } x > 4 \end{cases}$$

The probability that x lies between 2 and 3, i.e., $P(2 \le x \le 3)$ is _____. (rounded off to three decimal places)

Official answer: **0.300 to 0.302**

SOLUTION

1. A density must integrate to 1 over its support: $\int_1^4 Cx^2\,dx = 1$.
2. $\int_1^4 x^2\,dx = \dfrac{4^3 - 1^3}{3} = \dfrac{63}{3} = 21$, so $21C = 1$ and $C = \dfrac{1}{21}$.
3. $P(2 \le x \le 3) = \int_2^3 \dfrac{x^2}{21}\,dx = \dfrac{1}{21} \cdot \dfrac{3^3 - 2^3}{3}$.
4. $= \dfrac{1}{21} \cdot \dfrac{19}{3} = \dfrac{19}{63}$.
5. $\dfrac{19}{63} \approx 0.30159$, which rounds to 0.302.

KEY IDEA

Fix the constant by normalising the density to total probability 1, then integrate the same density over the interval asked.

WATCH OUT FOR

Skipping the normalisation and integrating x^2 directly (giving $19/3$), or using $C = 1/3$ by integrating over the wrong limits.

Q.59 NAT · 2 MARKS

Digital Electronics › Sequential

Q.59 Consider a finite state machine (FSM) with one input X and one output f, represented by the given state transition table. The minimum number of states required to realize this FSM is _____. (Answer in integer)

Present state	Next state		Output f	
	$X = 0$	$X = 1$	$X = 0$	$X = 1$
A	F	B	0	0
B	D	C	0	0
C	F	E	0	0
D	G	A	1	0
E	D	C	0	0
F	F	B	1	1
G	G	H	0	1
H	G	A	1	0

Official answer: **5**

SOLUTION

1. Minimise by partition refinement. Two states are equivalent only if they give the same output for both inputs and move to equivalent states on both inputs.
2. Group the states by their output pair $(f|_{X=0}, f|_{X=1})$. This gives $P_1 = \{A, B, C, E\}$ with 00, $P_2 = \{D, H\}$ with 10, $P_3 = \{F\}$ with 11 and $P_4 = \{G\}$ with 01.
3. Refine P_1 by the classes of the next states. $A \to (F, B)$ lies in (P_3, P_1). $C \to (F, E)$ lies in (P_3, P_1). $B \to (D, C)$ lies in (P_2, P_1). $E \to (D, C)$ lies in (P_2, P_1). So P_1 splits into $\{A, C\}$ and $\{B, E\}$.
4. $D \to (G, A)$ and $H \to (G, A)$ have identical next states, so $\{D, H\}$ stays together.
5. Re-check with the new classes. $A \to (F, B)$ and $C \to (F, E)$ both go to $(\{F\}, \{B, E\})$. $B \to (D, C)$ and $E \to (D, C)$ have identical next states, going to $(\{D, H\}, \{A, C\})$. D and H still agree. No class splits further, so the partition is stable.
6. The final classes are $\{A, C\}$, $\{B, E\}$, $\{D, H\}$, $\{F\}$, $\{G\}$. The minimum number of states is 5.

KEY IDEA

Minimise a completely specified FSM by partition refinement. Start from the output classes, then split any class whose members go to different classes on some input, until no class splits.

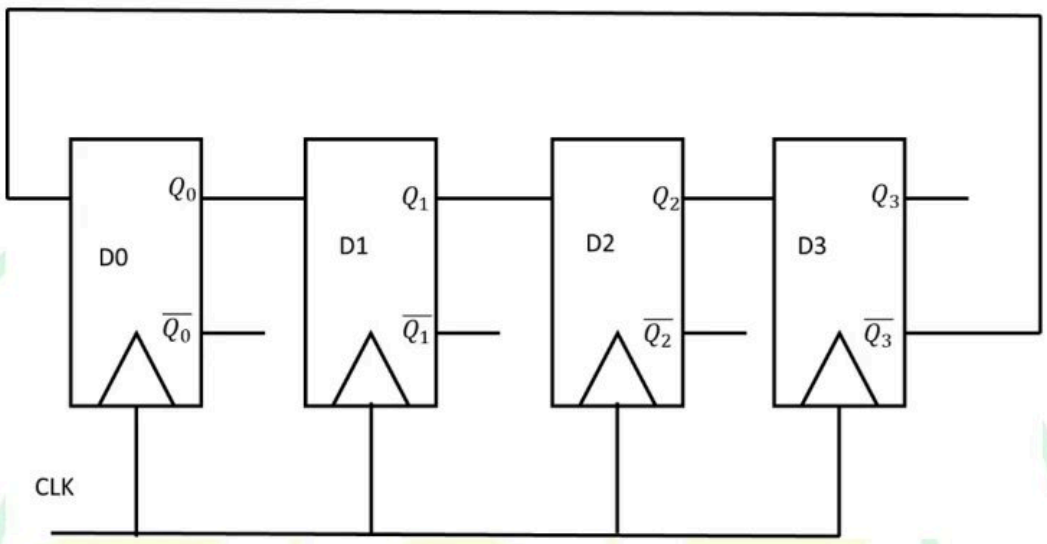
WATCH OUT FOR

Merging states just because their outputs match, such as all of A, B, C, E, without checking that their next states fall in equivalent classes. Another slip is stopping after one refinement pass.

Q.60 NAT · 2 MARKS

Digital Electronics › Sequential

Q.60 Consider the given sequential circuit designed using D-Flip-flops. The circuit is initialized with some value (initial state). The number of distinct states the circuit will go through before returning back to the initial state is _____ . (Answer in integer)



Official answer: **7 to 8**

SOLUTION

1. Read the connections from the figure: $D_1 = Q_0$, $D_2 = Q_1$, $D_3 = Q_2$. The feedback wire leaves the $\overline{Q_3}$ output of the last flip-flop and enters D_0, so $D_0 = \overline{Q_3}$. The Q_3 output is open. This is a shift register fed with the complement of its last output, which is a twisted-ring (Johnson) counter.
2. On each clock edge the state $Q_0Q_1Q_2Q_3$ shifts right by one place, and the new Q_0 is $\overline{Q_3}$.
3. Start from 0000 and trace clock by clock: $0000 \to 1000 \to 1100 \to 1110 \to 1111 \to 0111 \to 0011 \to 0001 \to 0000$.
4. The sequence returns to 0000 after 8 clock pulses, and the 8 states in it are all different.
5. An n-flip-flop Johnson counter has $2n$ states in its cycle, so $2 \times 4 = 8$.
6. The other $16 - 8 = 8$ states form a second cycle: $0100 \to 1010 \to 1101 \to 0110 \to 1011 \to 0101 \to 0010 \to 1001 \to 0100$. This also has 8 states, so whichever initial state is chosen, the circuit passes through 8 distinct states before returning to it.

KEY IDEA

Feeding the complement of the last flip-flop's output back to the first input of a shift register makes a Johnson counter. With n flip-flops its cycle has $2n$ states, and the other $2^n - 2n$ states form further cycles.

WATCH OUT FOR

Treating the circuit as a ring counter ($D_0 = Q_3$, 4 states) or as a full binary counter ($2^4 = 16$ states). The complemented feedback makes the cycle $2n = 8$ states long.

Q.61 NAT · 2 MARKS

Programming in C › Recursion

```
#include <stdio.h>
int foo(int S[],int size){
   if(size == 0) return 0;
   if(size == 1) return 1;
   if(S[0] != S[1]) return 1+foo(S+1,size-1);
   return foo(S+1,size-1);
}
int main(){
   int A[]={0,1,2,2,2,0,0,1,1};
   printf("%d",foo(A,9));
   return 0;
}
```

The value printed by the given C program is _____ . (Answer in integer)

Official answer: **5**

SOLUTION

1. Array A[0..8] = {0,1,2,2,2,0,0,1,1}, so foo is called as foo(A,9).
2. The recursion compares consecutive elements: while the current pair of adjacent elements (S[0] and S[1]) differ, it counts 1 and recurses on the rest of the array; while they are equal, it recurses without counting (skips one duplicate).
3. foo(A,9): A[0]=0 , A[1]=1 , different → 1 + foo(A+1,8)
4. foo(A+1,8) (window A[1..8] = 1,2,2,2,0,0,1,1): A[1]=1 , A[2]=2 , different → 1 + foo(A+2,7)
5. foo(A+2,7) (window A[2..8] = 2,2,2,0,0,1,1): A[2]=2 , A[3]=2 , equal → foo(A+3,6)
6. foo(A+3,6) (window A[3..8] = 2,2,0,0,1,1): A[3]=2 , A[4]=2 , equal → foo(A+4,5)
7. foo(A+4,5) (window A[4..8] = 2,0,0,1,1): A[4]=2 , A[5]=0 , different → 1 + foo(A+5,4)
8. foo(A+5,4) (window A[5..8] = 0,0,1,1): A[5]=0 , A[6]=0 , equal → foo(A+6,3)
9. foo(A+6,3) (window A[6..8] = 0,1,1): A[6]=0 , A[7]=1 , different → 1 + foo(A+7,2)
10. foo(A+7,2) (window A[7..8] = 1,1): A[7]=1 , A[8]=1 , equal → foo(A+8,1)
11. foo(A+8,1): base case size==1 → returns 1
12. Unwinding: foo(A+7,2) = 1; foo(A+6,3) = 1+1 = 2; foo(A+5,4) = 2; foo(A+4,5) = 1+2 = 3; foo(A+3,6) = 3; foo(A+2,7) = 3; foo(A+1,8) = 1+3 = 4; foo(A,9) = 1+4 = 5
13. The program prints 5.

KEY IDEA

The function walks the array with a "current index / remaining size" recursion, adding 1 whenever the current element differs from the next one and skipping the count (just recursing on the smaller window) whenever it matches, effectively counting "boundaries" between unequal adjacent elements while chaining through a run of equal ones, until only one element (or zero) is left.

WATCH OUT FOR

Losing track of which two elements are being compared at each recursive call (since the array pointer shifts by one each time but the comparison is always between the first two elements of the *current* window, i.e. positions `i` and `i+1` of the original array), or forgetting that a "match" case still advances the pointer by one without adding to the count, leading to either double-counting or missing a boundary.

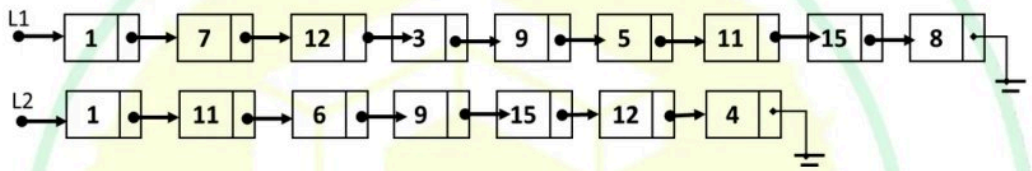
Q.62 NAT · 2 MARKS

Q.62 Let **LIST** be a datatype for an implementation of linked list defined as follows:

```
typedef struct list {
    int data;
    struct list *next;
} LIST;
```

Suppose a program has created two linked lists, L1 and L2, whose contents are given in the figure below (code for creating L1 and L2 is not provided here). L1 contains 9 nodes, and L2 contains 7 nodes.

Consider the following C program segment that modifies the list L1. The number of nodes that will be there in L1 after the execution of the code segment is _____ . (Answer in integer)




```
int find (int query, LIST *list) {
    while (list != NULL){
        if(list->data == query) return 1;
        list = list->next;
    }
    return 0;
}
int main () {
    … … …
    ptr1=L1; ptr2=L2;
    while (ptr1->next != NULL){
        query = ptr1->next->data;
        if (find (query, L2))
            ptr1->next = ptr1->next->next;
        else ptr1 = ptr1->next;
    }
    … … …
    return 0;
}
```

Official answer: **5**

Q.63 NAT · 2 MARKS

Programming in C › C basics

Consider the following C program:

```
#include <stdio.h>
int gate (int n) {
   int d, t, newnum, turn;
   newnum = turn = 0; t=1;
   while (n>=t) t *= 10;
   t /=10;
   while (t>0) {
      d = n/t;
      n = n%t;
      t /= 10;
      if (turn) newnum = 10*newnum + d;
      turn = (turn + 1) % 2;
   }
   return newnum;
}
int main () {
   printf ("%d", gate(14362));
   return 0;

}
```

The value printed by the given C program is _____ . (Answer in integer)

Official answer: **46**

SOLUTION

1. First while loop: t starts at 1 and is repeatedly multiplied by 10 while n (=14362) >= t, so t goes 1 → 10 → 100 → 1000 → 10000 → 100000, stopping when t=100000 first exceeds n.
2. The statement t /= 10 brings t back to 10000, the place-value of n's leading (most significant) digit.
3. The second while loop (t>0) runs 5 times, each time extracting the top remaining digit via d=n/t and n=n%t, then shrinking t by /=10; turn starts at 0 (false) and toggles 0↔1 after every iteration.
4. Iteration 1 (t=10000): d=1, n becomes 4362, t becomes 1000. turn was 0, so newnum stays 0. turn becomes 1.
5. Iteration 2 (t=1000): d=4, n becomes 362, t becomes 100. turn was 1, so newnum = 10*0+4 = 4. turn becomes 0.
6. Iteration 3 (t=100): d=3, n becomes 62, t becomes 10. turn was 0, so newnum stays 4. turn becomes 1.
7. Iteration 4 (t=10): d=6, n becomes 2, t becomes 1. turn was 1, so newnum = 10*4+6 = 46. turn becomes 0.
8. Iteration 5 (t=1): d=2, n becomes 0, t becomes 0. turn was 0, so newnum stays 46. turn becomes 1.
9. Loop exits since t=0; the function returns newnum=46, which main() prints.

KEY IDEA

The function alternately drops and keeps digits of n, most-significant first: turn starts false (0) so the leading digit is dropped, then toggles true/false each iteration. For 14362 (digits 1,4,3,6,2 from MSD), digit 1 is dropped, 4 is kept, 3 is dropped, 6 is kept, 2 is dropped, leaving newnum built from 4 then 6 giving 46.

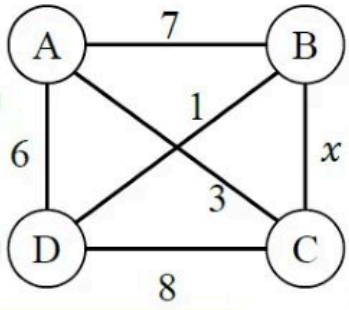
WATCH OUT FOR

Forgetting that turn starts at 0 (false) and toggles AFTER each digit is examined, so the loop skips appending the very first (most significant) digit, leading to including digit 1 and wrongly getting 146 instead of 46. Another slip is misjudging where the first while loop stops: it multiplies t until t first EXCEEDS n (t=100000), and only the following t/=10 brings it to the leading digit's place value (10000), not to 1000.

Q.64 NAT · 2 MARKS

Data Structures and Algorithms › Graph algorithms

Q.64 The maximum value of x such that the edge between the nodes B and C is included in every minimum spanning tree of the given graph is _____ . (answer in integer)



Official answer: **5**

SOLUTION

1. Read the graph from the figure. It is K_4 on $\{A, B, C, D\}$ with $AB = 7$, $AD = 6$, $DC = 8$, $BC = x$, $BD = 1$ and $AC = 3$.
2. The correct test for an edge being in EVERY minimum spanning tree: an edge e is in every MST if and only if e is not a maximum-weight edge (strict or tied) on any cycle. Equivalently, e must be the strictly lightest edge crossing some cut. A tie for the maximum is not enough. If e ties another edge for the heaviest weight on a cycle, you can swap e for that edge and get another MST of the same weight that does not contain e.
3. List every cycle through BC and the largest of its other edges:
   - $B-C-A-B$ (edges $CA = 3$, $AB = 7$): largest is 7.
   - $B-C-D-B$ (edges $CD = 8$, $DB = 1$): largest is 8.
   - $B-C-A-D-B$ (edges $CA = 3$, $AD = 6$, $DB = 1$): largest is 6.
   - $B-C-D-A-B$ (edges $CD = 8$, $DA = 6$, $AB = 7$): largest is 8.
4. For BC not to be a maximum on any of these cycles, x must be strictly less than every one of these values:
$$x < \min(7,\ 8,\ 6,\ 8) = 6.$$
5. The same condition from the cut side. Take the cut $\{B, D\}$ versus $\{A, C\}$. The edges crossing it are $AB = 7$, $AD = 6$, $DC = 8$ and $BC = x$. If $x < 6$, then BC is the unique lightest edge crossing this cut, so every MST must contain it (cut property).
6. Check $x = 6$. Kruskal's algorithm takes $BD = 1$, then $AC = 3$, leaving the two components $\{B, D\}$ and $\{A, C\}$. The next lightest edges are $AD = 6$ and $BC = 6$, and either one joins the two components. So $\{BD, AC, AD\}$ and $\{BD, AC, BC\}$ are both MSTs of weight 10. The first one does not contain BC, so $x = 6$ fails.
7. For any $x > 6$, BC is the strict unique maximum on cycle $B-C-A-D-B$, so no MST contains it.
8. Check $x = 5$. Kruskal's algorithm takes $BD = 1$, $AC = 3$, then $BC = 5$, which joins $\{B, D\}$ and $\{A, C\}$. All remaining edges $(6, 7, 8)$ would close a cycle. The MST $\{BD, AC, BC\}$ of weight 9 is unique and contains BC.

9. So x must be an integer with $x < 6$, and the maximum value is $x = 5$.

KEY IDEA
An edge is in every MST exactly when it is the strictly lightest edge crossing some cut. Here that cut is {B, D} versus {A, C}. The other edges crossing it weigh 6, 7 and 8, so B–C must weigh strictly less than 6. The largest integer that works is 5.

WATCH OUT FOR
Treating a tie as good enough and answering 6. When x = 6, B–C ties with A–D = 6 across the cut {B, D} | {A, C}. Kruskal's algorithm can then pick A–D instead, which gives an MST without B–C. "Included in SOME MST" allows ties, but "included in EVERY MST" needs a strict inequality.

Q.65 NAT · 2 MARKS

Data Structures and Algorithms › Sorting & hashing

In a double hashing scheme, $h_1(k) = k \bmod 11$ and $h_2(k) = 1 + (k \bmod 7)$ are the auxiliary hash functions. The size m of the hash table is 11. The hash function for the i-th probe in the open address table is $[h_1(k) + i\,h_2(k)] \bmod m$. The following keys are inserted in the given order: 63, 50, 25, 79, 67, 24.

The slot at which key 24 gets stored is _____. (Answer in integer)

Official answer: **10 to 11**

SOLUTION

1. Compute $h_1(k) = k \bmod 11$ for the first five keys, which are inserted with no collisions since all slots differ: $h_1(63) = 8$, $h_1(50) = 6$, $h_1(25) = 3$, $h_1(79) = 2$, $h_1(67) = 1$.
2. Table after these insertions: slot 1 = 67, slot 2 = 79, slot 3 = 25, slot 6 = 50, slot 8 = 63; all other slots (0, 4, 5, 7, 9, 10) are empty.
3. For key 24: $h_1(24) = 24 \bmod 11 = 2$.
4. $h_2(24) = 1 + (24 \bmod 7) = 1 + 3 = 4$.
5. Probe $i = 0$: $[2 + 0 \cdot 4] \bmod 11 = 2$: occupied by 79, so probing continues.
6. Probe $i = 1$: $[2 + 1 \cdot 4] \bmod 11 = 6$: occupied by 50, so probing continues.
7. Probe $i = 2$: $[2 + 2 \cdot 4] \bmod 11 = 10 \bmod 11 = 10$: this slot is empty, so key 24 is stored at slot 10.

KEY IDEA
In double hashing, the probe sequence for a key k is $[h_1(k) + i\,h_2(k)] \bmod m$ for $i = 0, 1, 2, \ldots$; since $h_2(k)$ never depends on i, each key traces a fixed "skip" sequence through the table until it lands on an empty slot.

WATCH OUT FOR
Forgetting that once a slot is found occupied, the probe sequence must be recomputed with the *next* value of i (not restarted or incremented incorrectly), and mixing up which key already occupies a slot, leading students to place 24 in slot 6 or slot 2 by mistake.

FROM IAE ACADEMY

Preparing for GATE CS?

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

FREE COUNSELLING

Chat on WhatsApp

Call +91 84204 36524

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

Scan to chat on WhatsApp

THE GATE CS COURSE

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

Scan: course page

See the course

FREE ON IAEACADEMY.IN

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

Scan: more papers

More solved papers

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

Send to a friend on WhatsApp

IAE Academy free study material. The questions are those of the GATE CS 2025 Set 1 question paper, organised by IIT Roorkee; the answers are from its official answer key. Prepared and published by IAE Academy, Kolkata. Free to read and free to share, unaltered and with attribution. Not affiliated with, endorsed by or produced by IIT Roorkee or the GATE committee.

IAE ACADEMY INSTITUTE FOR ENGINEERS

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

227A Rash Behari Ave, Ballygunge, Kolkata 700019

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

Get directions