GATE 2026 Data Science & Artificial Intelligence Question Paper with Solutions

collegedunia

General Instructions

(i) This booklet contains 65 questions, each provided with a complete, step-by-step solution.

(ii) It comprises 35 single-correct multiple-choice questions, 12 multiple-correct questions and 18 numerical / integer-type questions.

(iii) Attempt each question on your own before reviewing the given solution.

(iv) For multiple-correct questions, all correct options must be selected to earn full marks.

(v) For numerical questions, report the answer rounded exactly as asked.

1.

Verbosity : Brevity :: Insolence : _____

Choose the word that best fills the blank.

(A) Innocence

(B) Respect

(C) Solace

(D) Wealth

Correct Answer: (B) Respect

Solution:

This question tests a word analogy built on an antonym pair. Look at the first pair, Verbosity and Brevity. Verbosity means using far more words than needed, while brevity means being short and to the point. The two words sit at opposite ends of the same idea, so the link joining them is 'is the opposite of'.

1. **Innocence**: this means being free of guilt or wrongdoing. It sits on a scale of guilt, not manners, so it does not answer to insolence.
2. **Respect**: insolence means rude, disrespectful behaviour. Respect is exactly the opposite of that, esteem and regard for another person. This keeps the same antonym link as the first pair.
3. **Solace**: this word means comfort given in a time of grief. It has nothing to do with rudeness or manners.
4. **Wealth**: this refers to money and riches, a different idea altogether from how a person behaves towards others.

Since Brevity is the opposite of Verbosity, the blank must hold the opposite of Insolence. Respect is that word, so option (B) completes the analogy correctly.

Let's summarize:

- The given pair, Verbosity and Brevity, are antonyms.
- Insolence means rudeness, and its antonym is Respect.
- Innocence, Solace and Wealth belong to unrelated ideas (guilt, comfort, money), so they do not fit the antonym pattern.

The word that best fills the blank is Respect.

Quick Tip: Find how the first two words are related (they are opposites), then look for the opposite of the second word in the blank pair.

2.

The product of the digits of a three-digit number is 70. The sum of the digits of this three-digit number is _____

(A) 12

(B) 14

(C) 16

(D) 18

Correct Answer: (B) 14

Solution:

Step 1: Set up the digit constraint.

Let the three digits of the number be a, b and c, each a whole number from 0 to 9 (and a, the leading digit, at least 1 since it is a three digit number). We are told $a \times b \times c = 70$.

Step 2: Break 70 into three digits.

Write 70 using its prime factors: $70 = 2 \times 5 \times 7$. All three factors, 2, 5 and 7, are already single digits, so $\{2, 5, 7\}$ is one valid set of digits with product 70.

Step 3: Check no other digit set gives 70.

For a different set of digits to multiply to 70, two of the prime factors would have to be combined into a single digit. Combining any two of 2, 5, 7 gives 10, 14 or 35, and each of these is bigger than 9, so it cannot be one digit. Bringing a 1 into the mix does not help either, since then the remaining two digits would need to multiply to 70, and no pair of digits from 1 to 9 does that (the biggest such product, 9×9, is only 81, and checking every divisor pair of 70 shows none of them are both single digits). So 2, 5 and 7 are the only three digits whose product is 70.

Step 4: Add the digits.

Whatever order these digits appear in the number (257, 275, 527, 572, 725 or 752), the digit sum stays the same: $2 + 5 + 7 = 14$.

Final Answer:

The sum of the digits is 14.

$\boxed{14}$

Quick Tip: Break 70 into three factors that are each a single digit, then add them.

3.

The four pieces of a puzzle are shown in the figure below.

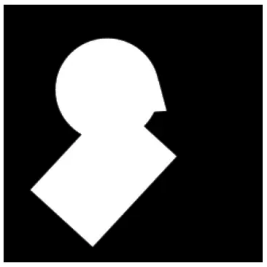
Which one of the figures labelled as P, Q, R, and S can be constructed by using each of the four pieces only once without overlaps?

(A)

P

Figure P

(B)



Q

Figure Q

(C)

R

Figure R

(D)

S

Figure S

Correct Answer: (A)

P

Figure P

Solution:

This question checks visual assembly, fitting four irregular black pieces together, edge to edge with no gaps and no overlaps, to recreate one of the four candidate silhouettes. The four given pieces are a rounded piece with a smooth curved bite cut into one edge, a dome shaped piece with a matching curved notch, a piece with a jagged zig-zag edge, and a piece shaped like twin triangular peaks (a small peak beside a taller one).

Each candidate silhouette (P, Q, R, S) shows a bird-like outline: a round head, a pointed beak on the right, and an angular body below, joined by a jagged little step where the head meets the body. The four pieces must combine, without stretching past what their true edges allow, to trace out that exact outline.

1. **Figure P**: the beak here is one clean, straight-edged spike. This is exactly what you get when the taller peak of the twin-peaked piece becomes the beak, while its smaller peak tucks neatly inside the curved bite of the rounded piece, hidden from the outline. The jagged piece supplies the small step at the neck, and the dome piece rounds off the shoulder. Every edge of every piece lines up with a matching edge on another piece, so the four pieces close up perfectly with nothing left over and nothing overlapping.
2. **Figure Q**: there is no beak at all here, just a small stepped notch where a beak should be. Since one of the four pieces is a long pointed spike, that spike has to appear somewhere on the outer boundary of the assembled shape. Q's outline has nowhere for that spike to go, so it cannot be built from these four pieces.

3. **Figure R**: the beak is present, but there is extra jagged detail right where the beak meets the head, more notches than the pieces can actually produce at that joint. Forcing the pieces together this way leaves a mismatched edge, so R cannot be assembled without a gap or overlap.
4. **Figure S**: the notch near the beak is shaped like a star or crown with an extra upward spike, again more detail than the boundary of the four pieces can supply at that single joint. This extra spike does not correspond to any edge on the four pieces, so S is not achievable either.

Only Figure P uses every edge of the four pieces exactly once, in a way that closes up cleanly with no leftover gaps or overlaps.

Let's summarize:

- Match curved edges to curved edges and straight edges to straight edges across the four pieces before checking any candidate figure.
- A candidate that shows a spike or notch the pieces cannot produce (Q's missing beak, R and S's extra jagged detail) can be ruled out immediately.
- Figure P is the only outline that the four pieces trace out exactly.

The correct figure is P.

Quick Tip: Count the distinct edge types on the four pieces (one curve pair, one jagged edge, one spike) and check which candidate figure shows exactly that many of each, no extra bumps or missing spikes.

4.

Consider two distinct positive real numbers m, n, with $m > n$.

Let $x = n^{\log_{10}(m)}$ and $y = m^{\log_{10}(n)}$. The relation between x and y is _____.

(A) $x > y$

(B) $x < y$

(C) $x = y$

(D) $x = \log_{10}(y)$

Correct Answer: (C) $x = y$

Solution:

Step 1: Understanding the Question.

We are given two distinct positive real numbers m and n, with $m > n$, and two quantities built from them, $x = n^{\log_{10}(m)}$ and $y = m^{\log_{10}(n)}$. We need to find how x and y compare. Since m and n are both positive, x and y are also both positive, so it is safe to take $\log_{10}$ of both sides to compare them.

Step 2: Take the logarithm of x.

Using the log power rule $\log_{10}(a^k) = k \cdot \log_{10}(a)$, with $a = n$ and $k = \log_{10}(m)$:

$$\log_{10}(x) = \log_{10}\left(n^{\log_{10}(m)}\right) = \log_{10}(m) \cdot \log_{10}(n)$$

Step 3: Take the logarithm of y.
The same power rule applies here, with $a = m$ and $k = \log_{10}(n)$:

$$\log_{10}(y) = \log_{10}\left(m^{\log_{10}(n)}\right) = \log_{10}(n) \cdot \log_{10}(m)$$

Step 4: Compare the two results.
$\log_{10}(x)$ and $\log_{10}(y)$ are both equal to the same product, $\log_{10}(m) \cdot \log_{10}(n)$, just written in a different order. Multiplication does not care about order, so:

$$\log_{10}(x) = \log_{10}(y)$$

Since $\log_{10}$ is a one to one function on positive numbers (equal logs mean equal arguments), this gives $x = y$. Notice this holds true no matter what specific positive values m and n take, and the condition $m > n$ turns out not to matter at all for this equality.

Step 5: Why the other options are wrong.
Options (A) and (B) claim a strict inequality between x and y, but we have shown their logarithms are exactly equal for any valid m, n, so neither can be strictly greater. Option (D) claims $x = \log_{10}(y)$, but since $x = y$ here, this would require $y = \log_{10}(y)$, which is false for a generic positive y (for example $y = 10$ gives $\log_{10}(10) = 1$, not 10).

Final Answer:
x and y are always equal.

$$\boxed{x = y}$$

Quick Tip: Take $\log_{10}$ of both x and y and use the rule $\log_{10}(a^k) = k \log_{10}(a)$.

5.

'If his latest movie had been a commercial success, the actor would have made enough money to sponsor his next movie.'

Based only on the above sentence, which one of the following statements is true?

(A) The actor will certainly sponsor his next movie.

(B) His latest movie was a commercial success.

(C) The actor made enough money from his latest movie.

(D) His latest movie was not commercially successful.

Correct Answer: (D) His latest movie was not commercially successful.

Solution:

This sentence is a third conditional, built with 'if + past perfect' in the if-clause and 'would have + past participle' in the main clause ('If his latest movie had been..., the actor would have made...'). This structure talks about a hypothetical past that did not actually happen, and it always carries the built-in meaning that the stated condition was false in real life.

1. **The actor will certainly sponsor his next movie**: the sentence talks about what would have happened in the past under a condition that did not hold, it makes no promise about the future at all. This cannot be concluded.
2. **His latest movie was a commercial success**: a third conditional's if-clause states the opposite of what really happened. Saying 'if it had been a success' tells us it was not a success, so this statement contradicts the sentence.
3. **The actor made enough money from his latest movie**: the sentence says he 'would have made enough money', meaning this did not actually happen. So he did not make that money.
4. **His latest movie was not commercially successful**: this matches the real-world implication of the third conditional exactly. Because the condition ('had been a success') is presented as unreal, the actual state of affairs is the opposite, the movie was not a commercial success.

Since a third conditional always signals that its if-clause condition was not true, the fact that we are told what 'would have' happened if the movie 'had been' a success tells us, in reality, it was not one.

Let's summarize:

- 'If + past perfect, would have + past participle' describes an unreal past condition.
- The truth of the real world is the opposite of what the if-clause states.
- Here, that means the movie was not a commercial success, matching option (D).

The only statement that can be concluded is that his latest movie was not commercially successful.

Quick Tip: 'If + had been, would have' describes an imagined past that did not happen, so the real situation is the opposite of the if-clause.

6.

'My friend and I parted ___ the door ___ the cabin that I had rented ___ the night.'

Choose the option with the correct sequence of words to fill the blanks.

(A) at; of; for

(B) for; at; of

(C) of; for; in

(D) in; of; for

Correct Answer: (A) at; of; for

Solution:

Step 1: Understanding the Question.
The sentence has three blanks, each needing a preposition: 'My friend and I parted ___ the door ___ the cabin that I had rented ___ the night.' We need to pick the set of three prepositions that fits naturally at each blank.

Step 2: Key Formula or Approach.
Fill each blank one at a time using the standard preposition that goes with the phrase around it: a location point uses 'at', a possessive or 'belonging to' relationship uses 'of', and a stretch of time being rented for uses 'for'.

Step 3: Detailed Explanation.
First blank, 'parted ___ the door': we say someone parts (says goodbye and separates) 'at' a specific point or place, so 'parted at the door' is the natural phrase.
Second blank, 'the door ___ the cabin': this describes the door belonging to the cabin, and 'belonging to' is expressed with 'of', giving 'the door of the cabin'.
Third blank, 'rented ___ the night': renting something is for a length of time, and 'for' is the preposition used to state a duration or purpose, giving 'rented for the night'.
Putting the three together: 'My friend and I parted at the door of the cabin that I had rented for the night.'
This reads naturally and each preposition matches its usual collocation.

Step 4: Why the other options are wrong.
Option (B), 'for; at; of', would give 'parted for the door', which does not make sense since 'for' does not mark a location where someone parts.
Option (C), 'of; for; in', would give 'parted of the door', which is not a standard collocation, 'parted of' is not correct usage.
Option (D), 'in; of; for', would give 'parted in the door', which is unnatural since 'in the door' describes being inside the door frame, not the point of separation.

Step 5: Final Answer.
The correct sequence is 'at; of; for', matching option (A).

Quick Tip: Check each blank against the phrase it sits in: parted 'at' a place, the door 'of' something, rented 'for' a duration.

7.

Five integers are picked from 0 to 20, with possible repetitions, such that their mean is 12, median is 18, and they have a single mode of 20.

Ignoring permutations, the number of ways to pick these five integers is _____

(A) 0

(B) 1

(C) 2

(D) 3

Correct Answer: (B) 1

Solution:

Step 1: Set up the sorted list and use the mean.
Arrange the five chosen integers in non-decreasing order as $a \le b \le c \le d \le e$, with every value between 0 and 20. The mean is 12, so the sum of all five numbers is $5 \times 12 = 60$.

Step 2: Use the median to fix the middle value.
The median of five sorted numbers is the middle (third) value, so $c = 18$. Since the list is sorted, this also forces $a \le b \le 18$ and $18 \le d \le e \le 20$.

Step 3: Use the mode condition to pin down d and e.
We need 20 to be the single (unique) mode, meaning 20 appears strictly more often than every other value in the list. Since $a, b \le c = 18$, the value 20 can only appear among d and e, so 20 can appear at most twice. If 20 appeared only once, no value in the list could stand out as a repeated, unique mode (if every value were distinct there would be no single mode at all). So for 20 to be a genuine, unique mode, it must appear exactly twice, which forces $d = e = 20$.

Step 4: Solve for a and b.
With $c = 18$, $d = e = 20$, the sum condition gives $a + b + 18 + 20 + 20 = 60$, so $a + b = 2$. Since $a \le b \le 18$ and both are non-negative integers, the only possibilities are $(a, b) = (0, 2)$ or $(a, b) = (1, 1)$.

Step 5: Apply the single-mode condition to rule out ties.
If $(a, b) = (1, 1)$, the value 1 would also appear exactly twice, tying with 20's frequency of two, so 20 would no longer be the unique mode. This case is invalid.
If $(a, b) = (0, 2)$, the values 0, 2 and 18 each appear once, and only 20 appears twice, so 20 is genuinely the single mode. This case is valid, giving the multiset $\{0, 2, 18, 20, 20\}$.

Step 6: Count the valid selections.
Only one multiset of five integers satisfies every condition (mean 12, median 18, single mode 20). Since the question asks to ignore permutations, that is, count unordered selections and not orderings, the answer is exactly one way.

Final Answer:
There is exactly 1 way to pick the five integers: $\{0, 2, 18, 20, 20\}$.

$\boxed{1}$

Quick Tip: The median fixes the middle number at 18, so 20 can only sit in the fourth and fifth positions, meaning it can appear at most twice.

8.

Rishi and Swathi are students of Class 5. Pavan and Tanvi are students of Class 4. Rishi and Pavan are boys. Swathi and Tanvi are girls. The four students played a total of three games of chess. The games were played one after another. A player who lost a game did not participate in any more games. It was observed that:

1. the first game was the only game where two students of the same class played against each other,
2. the students of Class 5 won more games than the students of Class 4, and
3. the boys won two games and the girls won one game.

The student who did not lose any game is _____.

(A) Pavan

(B) Rishi

(C) Swathi

(D) Tanvi

Correct Answer: (D) Tanvi

Solution:

Step 1: Understand the tournament structure.
Four students play three games, one after another, and a player who loses a game drops out for good. With four distinct players and three games, the natural structure is: game 1 is between two players, its winner plays a third (new) player in game 2, and the winner of game 2 plays the fourth (last) player in game 3. This way, every player who ever loses is eliminated immediately, and all four players get to play at least once, while whoever keeps winning plays more than once.

Step 2: Use condition (i) to fix game 1.
Condition (i) says only the first game pairs two students of the same class. Class 5 has Rishi and Swathi, Class 4 has Pavan and Tanvi. So game 1 is either 'Rishi vs Swathi' or 'Pavan vs Tanvi', and games 2 and 3 must each be a cross-class matchup.

Step 3: Use the win totals from (ii) and (iii).
There are 3 games in total, so total wins split between Class 5 and Class 4 must add to 3. Condition (ii) says Class 5 won more games than Class 4, so the only way to split 3 wins with Class 5 strictly ahead is Class 5 = 2 wins, Class 4 = 1 win. Condition (iii) says boys won 2 games and girls won 1 game overall.

Step 4: Test game 1 = Rishi vs Swathi.
Say Rishi wins game 1 (the other case is checked in Step 6). The loser, Swathi, is eliminated with 0 wins forever. Rishi (the winner) then plays game 2 against one of the Class 4 students. If Rishi lost game 2, the winner (a Class 4 student) would then face the other Class 4 student in game 3, since only Rishi and one Class 4 student remain unused, but that would make game 3 a same-class match, contradicting condition (i). So Rishi must also win game 2. Rishi then plays game 3 against the last remaining student, the second Class 4 student. If Rishi won all three games, all 3 wins would belong to one student, giving either boys 3 or girls 3, which does not match 'boys 2, girls 1' from condition (iii). So Rishi must lose game 3, meaning the Class 4 student in game 3 wins it.

Step 5: Assign the classes and genders to match the win counts.
So far: Rishi wins games 1 and 2 (2 wins), and the Class 4 student in game 3 wins game 3 (1 win). This matches Class 5 = 2, Class 4 = 1 from Step 3. For boys to total 2 and girls to total 1, Rishi (a boy) contributing 2 wins already satisfies the boys' total on his own, so the girls' 1 win must come from the winner of game 3. That means the Class 4 student who wins game 3 is a girl, so it is Tanvi, not Pavan. This also fixes Pavan as the Class 4 student who lost to Rishi in game 2.

Step 6: Check if Swathi could have won game 1 instead.
If Swathi had won game 1 instead of Rishi, she (a girl) would go on to win games 1 and 2 by the same reasoning as Step 4, giving girls 2 wins, which contradicts 'girls won only 1 game'. So Swathi cannot be the game 1 winner, confirming Rishi wins game 1.

Step 7: Lay out the full sequence and find who never lost.
Game 1: Rishi beats Swathi (Swathi eliminated, played 1 game, lost it).
Game 2: Rishi beats Pavan (Pavan eliminated, played 1 game, lost it).
Game 3: Tanvi beats Rishi (Rishi played 3 games total, won 2 and lost 1).
Tanvi only ever played game 3, and she won it, so she never lost a single game.

Final Answer:
The student who did not lose any game is Tanvi.

Tanvi

Quick Tip: With four players and three knockout-style games, the win totals split as Class 5 = 2 wins, Class 4 = 1 win; work out who can hold those wins without creating a second same-class game.

9. $P, Q, R, S, X,$ and Y are distinct single-digit whole numbers taking values from 0 to 9.
PQ is a two-digit number with Q in the units place and P in the tens place. Similarly, RS is a two-digit number.
It is known that PQ and RS are consecutive numbers and $(PQ)^2 + (RS)^2 = XYP$, with XYP being a three-digit number.
The value of Y is _____.

(A) 4
(B) 5
(C) 6
(D) 7

Correct Answer: (C) 6

Solution:

Step 1: Set up what the numbers can look like.
Write $PQ = 10P + Q$ and $RS = 10R + S$. Both are two digit numbers, so each lies between 10 and 99. We are

told PQ and RS are consecutive, so they differ by exactly 1, and their squares add up to a three digit number XYP, whose own units digit is the same P that is the tens digit of PQ.

Step 2: Bound how big the two numbers can be.
Since XYP has only three digits, $(PQ)^2 + (RS)^2 \leq 999$. If we let the smaller number be a and the larger be $a + 1$, then $a^2 + (a+1)^2 \leq 999$, which simplifies to $2a^2 + 2a + 1 \leq 999$, so $a^2 + a \leq 499$. Checking values, $a = 21$ gives $21^2 + 21 = 462$ (fits), while $a = 22$ gives $22^2 + 22 = 506$ (too big). So a can only run from 10 up to 21.

Step 3: Use the distinctness condition to narrow it down.
If both a and $a + 1$ sit in the same tens decade (like 12 and 13, both starting with tens digit 1), then P (tens digit of a) and R (tens digit of $a + 1$) become equal, which breaks the rule that all six letters must be distinct digits. The only pair in our range 10 to 21 where the tens digit actually changes between a and $a + 1$ is $a = 19$ (tens digit 1) and $a + 1 = 20$ (tens digit 2). Every other consecutive pair in this range keeps the same tens digit on both numbers, so this is the only candidate worth testing.

Step 4: Check the candidate $PQ = 19$, $RS = 20$.
Here $P = 1, Q = 9, R = 2, S = 0$. Compute $19^2 + 20^2 = 361 + 400 = 761$. This is indeed a three digit number, and its units digit is 1, which matches $P = 1$ exactly, as required. So $XYP = 761$ gives $X = 7$, $Y = 6$, $P = 1$.

Step 5: Confirm all six digits are distinct.
We now have $P = 1, Q = 9, R = 2, S = 0, X = 7, Y = 6$. These six values, 1, 9, 2, 0, 7, 6, are all different from each other, so every condition in the question is satisfied. Reversing the order, that is trying $PQ = 20, RS = 19$, fails because then P would be 2, but the units digit of 761 is 1, so that assignment is rejected.

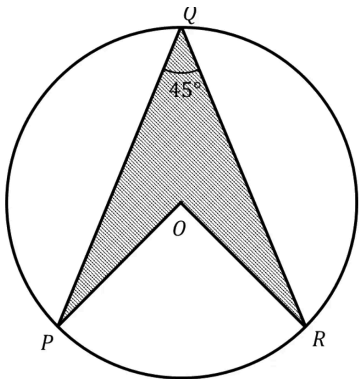
Final Answer:
The value of Y is 6, which is option (C).

$$\boxed{Y = 6}$$

Quick Tip: Bound the two numbers using $(PQ)^2 + (RS)^2 < 1000$, then notice only one consecutive pair in that range has different tens digits.

10. In the given figure, P, Q, and R are three points on a circle of radius 10 cm with O as its center, $\overline{PQ} = \overline{RQ}$, and $\angle PQR = 45°$. The figure is representative.

The area of the shaded region $PQRO$ is _____ cm^2.



(A) 50

(B) $25\sqrt{2}$

(C) $50\sqrt{2}$

(D) 100

Correct Answer: (C) $50\sqrt{2}$

Solution:

Step 1: Understand the shape of the shaded region.

The shaded region $PQRO$ is a quadrilateral traced by going P to Q, Q to R, R to O, then O back to P. Since O is the center, $OP = OQ = OR = 10$ cm, each being a radius. Drawing the diagonal OQ splits this quadrilateral into two triangles, $\triangle OPQ$ and $\triangle OQR$.

Step 2: Show the diagonal OQ bisects the given angle.

We are told $PQ = RQ$, and also $OP = OR$ (both radii). So triangles OPQ and ORQ share the side OQ and have $OP = OR$, $QP = QR$, which by the SSS rule makes them congruent. This congruence means OQ splits the 45 degree angle $\angle PQR$ into two equal halves, so $\angle OQP = \angle OQR = 22.5°$.

Step 3: Find the angle at the center for one triangle.

In $\triangle OPQ$, the sides OP and OQ are equal (both 10 cm radii), so this triangle is isosceles, and the angles opposite these equal sides are equal too. That means $\angle OPQ = \angle OQP = 22.5°$. Since the three angles of a triangle add to 180 degrees, $\angle POQ = 180° - 22.5° - 22.5° = 135°$.

Step 4: Compute the area of $\triangle OPQ$.

For a triangle with two known sides and the included angle, area $= \frac{1}{2} \times \text{side}_1 \times \text{side}_2 \times \sin(\text{included angle})$. Here both sides are 10 cm and the included angle at O is 135 degrees:

$$\text{Area}(\triangle OPQ) = \frac{1}{2}(10)(10)\sin(135°) = 50 \times \frac{\sqrt{2}}{2} = 25\sqrt{2}\ \text{cm}^2$$

Step 5: Add the second triangle by symmetry.

Because $\triangle OPQ$ and $\triangle OQR$ are congruent (shown in Step 2), $\triangle OQR$ also has area $25\sqrt{2}$ cm^2. Adding both pieces gives the full shaded quadrilateral:

$$\text{Area}(PQRO) = 25\sqrt{2} + 25\sqrt{2} = 50\sqrt{2}\ \text{cm}^2$$

Final Answer:
The area of the shaded region is $50\sqrt{2}$ cm^2, option (C).

$$\boxed{50\sqrt{2}\ \text{cm}^2}$$

Quick Tip: Split the quadrilateral along OQ into two congruent triangles, each with two 10 cm sides and a 135 degree included angle.

11. For a classification problem, Principal Component Analysis (PCA) has been used to reduce the dimensionality of a feature space from 100 to 10.
Which of the following options is true about the angle θ between the first and the tenth principal components?

(A) $\theta = 0°$

(B) $\theta = 90°$

(C) $90° < \theta \leq 180°$

(D) $0 < \theta < 90°$

Correct Answer: (B) $\theta = 90°$

Solution:

Step 1: Understanding the Question.
We need to work out the geometric relationship between two different principal components produced by PCA, specifically the 1st and the 10th, when a 100 dimensional feature space is compressed down to 10 dimensions.

Step 2: Key Formula or Approach.
PCA works by finding the eigenvectors of the data's covariance matrix. A covariance matrix is always real and symmetric. A core theorem of linear algebra states that eigenvectors of a real symmetric matrix, when they correspond to distinct eigenvalues, are always mutually orthogonal, meaning the angle between any two of them is exactly 90 degrees.

Step 3: Detailed Explanation.
The principal components are simply these eigenvectors, ranked from the one with the largest eigenvalue (capturing the most variance, this is the 1st principal component) down to smaller eigenvalues. The 10th principal component is just another eigenvector of the same covariance matrix, associated with a different eigenvalue than the 1st. It does not matter that we reduced 100 dimensions down to 10, keeping only 10

components does not change the fact that each kept component is still orthogonal to every other one, since orthogonality is a property of the eigenvectors themselves, not of how many we choose to retain. So the 1st and 10th principal components must be perpendicular to each other.

Step 4: Why the other options fail.
Option (A), $\theta = 0°$, would mean the two components point in the exact same direction, but PCA components are built to be linearly independent and orthogonal, never parallel. Options (C) and (D) suggest some in-between acute or obtuse angle, but that would only happen for arbitrary vectors, not for eigenvectors of a symmetric matrix, which are guaranteed to be exactly perpendicular, not just roughly so.

Step 5: Final Answer.
The angle between the first and tenth principal components is exactly 90 degrees, option (B).

$$\boxed{\theta = 90°}$$

Quick Tip: Principal components are eigenvectors of a symmetric covariance matrix, and such eigenvectors are always mutually orthogonal.

12. Consider that you are training a classifier for a 10-class classification problem. Each input is represented as a 512-dimensional vector. There are 1000 samples, out of which the first 100 will be used for testing.
Let Leave-One-Out-Cross-Validation (LOOCV) be used for selection of the classifier model before testing.
Which of the following options is the correct number of validation splits that will be generated?

(A) 10

(B) 512

(C) 900

(D) 1000

Correct Answer: (C) 900

Solution:

Step 1: Understanding the Question.
We are told 100 out of 1000 samples are set aside purely for final testing, and model selection happens before that testing stage, using LOOCV on whatever samples remain. We need the number of validation splits LOOCV creates.

Step 2: Key Formula or Approach.
Leave-One-Out-Cross-Validation is a special case of k-fold cross validation where k equals the number of samples in the pool being used. In each split, exactly one sample is held out as the validation point, the model trains on all the rest, and this repeats once for every sample in that pool. So the number of splits generated by LOOCV always equals the size of the dataset it is run on, not the total dataset size, only the portion actually available to it.

Step 3: Detailed Explanation.
The question specifies model selection (which uses LOOCV) happens "before testing", meaning the 100 samples reserved for testing are excluded entirely from the LOOCV process, they are never touched during model selection. That leaves $1000 - 100 = 900$ samples as the pool LOOCV works on. Since LOOCV produces one split per sample in its working pool, it produces exactly 900 validation splits, one for each of these 900 samples taking a turn as the held-out validation point.

Step 4: Why the other options fail.
Option (A), 10, is the number of classes in the problem, which has nothing to do with how many times LOOCV partitions the data. Option (B), 512, is the input feature dimensionality, also unrelated to the sample count LOOCV cycles through. Option (D), 1000, would be correct only if LOOCV ran over the entire dataset including the 100 test samples, but the question explicitly reserves those for testing only, so they are excluded from the LOOCV pool.

Step 5: Final Answer.
The correct number of validation splits is 900, option (C).

$$\boxed{900}$$

Quick Tip: LOOCV always generates one split per sample in its working pool; here that pool excludes the 100 test samples, leaving 900.

13. Which of the following algorithms is NOT an example of uninformed search?

(A) Breadth First Search

(B) Depth First Search

(C) A* Search

(D) Depth-limited Search

Correct Answer: (C) A* Search

Solution:

Step 1: Understanding the Concept.
Search algorithms in AI are grouped into two families. Uninformed (also called blind) search algorithms explore the state space using only the problem definition itself, the list of possible moves and a way to check if a state is the goal, with no extra guidance about which direction looks more promising. Informed search algorithms, by contrast, use a heuristic function that estimates how close a state is to the goal, and use that estimate to guide the search more efficiently.

Step 2: Key Formula or Approach.
To classify each option, check whether it relies on a heuristic function $h(n)$ that estimates remaining distance or cost to the goal. If it does, it is informed search. If it explores purely by the order of the search tree or graph structure, it is uninformed search.

Step 3: Detailed Explanation.
Breadth First Search expands nodes level by level, nearest first, using no heuristic at all, this is uninformed search. Depth First Search dives down one branch as far as possible before backtracking, again with no heuristic, this is also uninformed search. Depth-limited Search is simply Depth First Search with a maximum depth cutoff imposed, it still uses no heuristic guidance, so it remains uninformed search. A* Search is different: it computes $f(n) = g(n) + h(n)$, where $g(n)$ is the actual cost so far and $h(n)$ is a heuristic estimate of the cost remaining to the goal. Because it actively uses this heuristic $h(n)$ to decide which node to expand next, A* Search is classified as informed search, not uninformed search.

Step 4: Final Answer.
The algorithm that is NOT an example of uninformed search is A* Search, option (C).

A* Search

Quick Tip: Uninformed search uses no heuristic; A* uses $f(n) = g(n) + h(n)$, so it belongs to informed search.

14. Which of the following statements is NOT true? (The names of the predicates are intuitive.)

(A) $\forall x \forall y\ Classmate(x, y) \Rightarrow Classmate(y, x)$

(B) $\forall x\ Likes(x, Icecream) \Rightarrow \neg\exists x\ \neg Likes(x, Icecream)$

(C) "Each king is a person" is equivalent to $\forall x\ IsKing(x) \wedge IsPerson(x)$

(D) "All humans are mortal" is equivalent to $\forall x\ IsHuman(x) \Rightarrow IsMortal(x)$

Correct Answer: (C) "Each king is a person" is equivalent to $\forall x\ IsKing(x) \wedge IsPerson(x)$

Solution:

Step 1: Understanding the Question.
We need to check four statements written in first-order logic and find the one that is logically incorrect, meaning either the formula is false in general or it wrongly claims to be equivalent to the given English sentence.

Step 2: Key Formula or Approach.
The standard rule for translating an English sentence of the form "every A is a B" or "each A is a B" into first-order logic is to use an implication: $\forall x\ A(x) \Rightarrow B(x)$, meaning "for every x, if x is an A, then x is a B". Using a conjunction $\wedge$ instead of an implication changes the meaning completely, since $\forall x\ A(x) \wedge B(x)$ claims every single object in the whole domain is both an A and a B, which is a much stronger and usually false statement.

Step 3: Detailed Explanation.
Check option (A): the classmate relation is naturally symmetric, if x is a classmate of y then y is a classmate of x, so $\forall x \forall y\ Classmate(x, y) \Rightarrow Classmate(y, x)$ correctly captures this and is a true statement. Check option (B): let A denote the sentence $\forall x\ Likes(x, Icecream)$. The expression $\neg\exists x\ \neg Likes(x, Icecream)$ is the textbook logical equivalent of $\forall x\ Likes(x, Icecream)$, meaning it also equals A. So the whole statement reads as $A \Rightarrow A$,

which is always true no matter what, a tautology, so option (B) is a true statement. Check option (C): the English sentence "Each king is a person" should translate as $\forall x\ IsKing(x) \Rightarrow IsPerson(x)$, using an implication. But option (C) instead uses $\forall x\ IsKing(x) \wedge IsPerson(x)$, a conjunction, which wrongly asserts that every object in the domain, kings and non-kings alike, is both a king and a person. This is not what the English sentence says, so this stated equivalence is false, making option (C) NOT true. Check option (D): "All humans are mortal" correctly translates as $\forall x\ IsHuman(x) \Rightarrow IsMortal(x)$, the standard implication form, so this is a true statement.

Step 4: Final Answer.
The statement that is NOT true is option (C), because it wrongly uses a conjunction where an implication is needed.

Option (C)

Quick Tip: "Every A is a B" translates using implication ($A(x) \Rightarrow B(x)$), never a conjunction ($A(x) \wedge B(x)$).

15. Consider that the quick sort algorithm is used to sort an array of n distinct randomly ordered elements. In every call, the pivot is chosen as the first element of the current subarray.
Let $T(n)$ denote the expected time to sort the array. Assume that the time to partition is linear in the size of the current subarray.
Which of the following recurrence relations correctly represents $T(n)$ in this scenario?

(A) $T(n) = T(1) + T(n-1) + O(n)$

(B) $T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{3n}{4}\right) + O(n)$

(C) $T(n) = 2T\left(\frac{n}{2}\right) + O(n)$

(D) $T(n) = \frac{1}{n}\sum_{k=0}^{n-1}[T(k) + T(n-k-1)] + O(n)$

Correct Answer: (D) $T(n) = \frac{1}{n}\sum_{k=0}^{n-1}[T(k) + T(n-k-1)] + O(n)$

Solution:

Step 1: Understand what makes this the expected time, not the worst case.
The array is randomly ordered, and the pivot is always taken as the first element of whatever subarray we are currently sorting. Because the input order is random, we cannot assume the pivot always lands in a fixed spot like the smallest or the middle element, instead, the pivot could equally likely end up being the smallest, the largest, or anything in between, once we finish partitioning.

Step 2: Work out the distribution of the pivot's final rank.
After partitioning around the pivot, suppose k elements end up smaller than the pivot (forming the left subarray of size k) and the remaining $n-1-k$ elements end up larger (forming the right subarray of size $n-1-k$). Since the array is randomly ordered, the first element is equally likely to be any of the n elements in

terms of rank, so k is equally likely to be any value from 0 up to $n-1$, each with probability $\frac{1}{n}$.

Step 3: Build the recurrence by averaging over all possible splits.
Given a particular value of k, the cost is $T(k) + T(n-1-k)$ for sorting the two resulting parts, plus $O(n)$ for the partition step itself (given as linear in the current subarray size). Since every split value k from 0 to $n-1$ is equally likely, the expected time is the average of $T(k) + T(n-1-k)$ over all these n equally likely outcomes, plus the partition cost:

$$T(n) = \frac{1}{n}\sum_{k=0}^{n-1}[T(k) + T(n-k-1)] + O(n)$$

This matches option (D) exactly.

Step 4: Why the other options fail.
Option (A), $T(n) = T(1) + T(n-1) + O(n)$, describes the worst case, where the pivot always ends up smallest or largest, giving one part of size 1 every single time, this happens only for already sorted or specially adversarial inputs, not for a randomly ordered array. Options (B) and (C) both assume a single fixed split ratio, one quarter-three quarters, or half-half, happens on every single call. But with a random array and a first-element pivot, the actual split ratio varies from call to call, we cannot pin it to one fixed ratio, we must average across every possible ratio, which is exactly what option (D) does.

Step 5: Final Answer.
The correct recurrence for the expected time is option (D).

$$\boxed{T(n) = \frac{1}{n}\sum_{k=0}^{n-1}[T(k) + T(n-k-1)] + O(n)}$$

Quick Tip: Random input means the pivot's rank after partitioning is uniformly distributed over 0 to n-1, so average $T(k) + T(n-1-k)$ over all k.

16. Consider the given Python program.

```
def append_to_lst(val, lst=[]):
    lst.append(val)
    return lst

print(append_to_lst(1))
print(append_to_lst(2))
print(append_to_lst(3, []))
```

Which of the following is the correct output of this program?

(A) [1]
[2]
[3]

(B) [1]
[1, 2]
[3]

(C) [1]
[2]
[1, 2, 3]

(D) [1]
[1, 2]
[1, 3]

Correct Answer: (B) [1]
[1, 2]
[3]

Solution:

Step 1: Understand the mutable default argument trap.
In Python, a default argument value like $lst = []$ is created only once, at the moment the function is defined, not fresh every time the function is called. If a call does not explicitly pass its own value for lst, Python reuses that exact same list object from before, including any changes made to it by earlier calls. Only a call that explicitly supplies its own list, like $append_to_lst(3, [])$, gets a brand new separate list instead of the shared one.

Step 2: Trace the first call.
$append_to_lst(1)$ does not pass a value for lst, so Python uses the shared default list, which starts as an empty list $[]$. Inside the function, 1 is appended, turning the shared list into $[1]$, and this $[1]$ is returned and printed. The shared default list itself is now permanently $[1]$ until it is mutated again.

Step 3: Trace the second call.
$append_to_lst(2)$ again does not pass a value for lst, so it uses that same shared list, which is currently $[1]$ from the previous call, not a fresh empty list. Appending 2 turns it into $[1, 2]$, and this $[1, 2]$ is what gets returned and printed.

Step 4: Trace the third call.
$append_to_lst(3, [])$ explicitly passes a brand new, separate empty list as lst. This new list has nothing to do with the shared default list from the earlier two calls. Appending 3 to this fresh list gives $[3]$, which is printed. The shared default list is untouched by this call and remains $[1, 2]$ internally, but that does not affect this printed output.

Step 5: Final Answer.
Putting the three printed lines together: $[1]$, then $[1, 2]$, then $[3]$. This matches option (B).

[1] [1, 2] [3]

Quick Tip: A mutable default argument like $lst = []$ is created once at function definition and reused across calls that do not pass their own list.

17. Let $R(A, B, C, D, E)$ be a relational schema with functional dependency set $F = \{A \to BC,\ CD \to E,\ E \to A\}$. Which of the following statements is correct?

(A) AD, ED and CD are the only candidate keys of R.

(B) AD and ED are the only candidate keys of R.

(C) A, E and CD are the only candidate keys of R.

(D) A and CD are the only candidate keys of R.

Correct Answer: (A) AD, ED and CD are the only candidate keys of R.

Solution:

Step 1: Recall what a candidate key is.
A candidate key is a minimal set of attributes whose closure under the given functional dependencies (FDs) covers every attribute of the relation. Minimal means no proper subset of it also covers every attribute.

Step 2: Test single attributes first.
We are given $F = \{A \to BC,\ CD \to E,\ E \to A\}$ on $R(A, B, C, D, E)$.
A^+: start with $\{A\}$. Since $A \to BC$, add B, C, giving $\{A, B, C\}$. No other FD's left side is now fully present, so $A^+ = \{A, B, C\} \neq R$. A alone is not a key.
E^+: start with $\{E\}$. Since $E \to A$, add A, giving $\{A, E\}$. Now $A \to BC$ applies, add B, C, giving $\{A, B, C, E\}$. D is still missing and nothing produces it, so $E^+ \neq R$. E alone is not a key.
$C^+ = \{C\}$ and $D^+ = \{D\}$: neither C nor D alone starts any FD, so their closures never grow. Neither is a key.

Step 3: Test the pair CD.
CD^+: start with $\{C, D\}$. Since $CD \to E$, add E, giving $\{C, D, E\}$. Since $E \to A$, add A, giving $\{A, C, D, E\}$. Since $A \to BC$, add B (C is already there), giving $\{A, B, C, D, E\} = R$. So CD is a superkey, and since neither C nor D alone works, CD is minimal, hence a candidate key.

Step 4: Test the pair AD.
AD^+: start with $\{A, D\}$. Since $A \to BC$, add B, C, giving $\{A, B, C, D\}$. Now both C and D are present, so $CD \to E$ applies, add E, giving $\{A, B, C, D, E\} = R$. So AD is a superkey. A alone reaches only $\{A, B, C\}$ and D alone reaches only $\{D\}$, so AD is minimal, hence a candidate key.

Step 5: Test the pair ED.
ED^+: start with $\{D, E\}$. Since $E \to A$, add A, giving $\{A, D, E\}$. Since $A \to BC$, add B, C, giving $\{A, B, C, D, E\} = R$. So ED is a superkey, and since E alone and D alone both fail, ED is minimal, hence a candidate key.

Step 6: Rule out the other options.
Option (B) leaves out CD, which we just proved is also a valid candidate key, so it is incomplete.
Option (C) claims A and E are keys on their own, but their closures never reach D, so neither is really a key.
Option (D) keeps only A and CD, but A fails as shown, and it also misses ED entirely.

Final Answer:
The complete, minimal set of candidate keys of R is AD, ED and CD.

Option (A)

Quick Tip: Find which attribute never appears on the right side of any FD, that attribute must be in every candidate key. Then pair it with each other attribute and check whether the closure covers all of R.

18. Consider that the visualization of a 3-dimensional data cube is showing Sales Quantity for each combination of the attributes Product Type, Month and Country.
From this, if we want to further visualize the Sales Quantity for each combination of Product Type, Month and State, which of the following OLAP operations should be performed?

(A) Slicing
(B) Dicing
(C) Roll-up
(D) Drill-down

Correct Answer: (D) Drill-down

Solution:

Step 1: Recall the four OLAP operations.
Slicing picks one fixed value for a single dimension and looks at the resulting lower-dimension cross section, for example fixing Country = India.
Dicing picks a sub-cube by fixing a range or a set of values across two or more dimensions at once.
Roll-up moves up a dimension's concept hierarchy, combining detailed values into a coarser summary, for example City data combined into Country data.
Drill-down does the opposite of roll-up: it moves down a concept hierarchy, breaking a summarized value into its more detailed components.

Step 2: Identify the hierarchy involved here.
In a geography dimension, the usual hierarchy is Country at the top, then State below it, then City below that. Country and State sit on the same dimension (geography), with State being a finer level of detail than Country.

Step 3: Match the change in the question to an operation.
We start with Sales Quantity grouped by Product Type, Month and Country (a coarser, more summarized geography level). We want to move to Sales Quantity grouped by Product Type, Month and State (a more

detailed geography level, since one Country contains many States).
No dimension is being removed (Slicing) and no dimension is being restricted to a sub-cube of specific values (Dicing), we are just moving to a lower, more detailed level of the same geography hierarchy.

Step 4: Why the other options are wrong.
(A) Slicing would mean fixing one dimension to a single value and dropping it, which is not what is happening, all three dimensions remain, only the level of detail of one of them changes.
(B) Dicing would mean selecting a sub-cube using multiple dimension restrictions, but here we are not restricting to specific values, we are changing the granularity.
(C) Roll-up would apply if we were going from State to Country (coarser), which is the reverse of what is asked here.

Final Answer:
Moving from Country (coarse) to State (detailed) within the same hierarchy is Drill-down.

Option (D): Drill-down

Quick Tip: Country and State belong to the same geography hierarchy, with State being more detailed than Country. Moving to a more detailed level of an existing hierarchy is Drill-down.

19. Let M be a randomly chosen non-empty subset of $S = \{1, 2, 3, \ldots, 2026\}$.
Which of the following is the probability that the product of all the elements of M is even?

(A) $\frac{2^{1013}(2^{1013} - 1)}{2^{2026}}$
(B) $\frac{2^{1013}}{2^{2026}}$
(C) $\frac{2^{1013}(2^{1013} - 1)}{2^{2026} - 1}$
(D) $\frac{1}{2^{2026} - 1}$

Correct Answer: (C) $\frac{2^{1013}(2^{1013} - 1)}{2^{2026} - 1}$

Solution:

Step 1: Count the total number of non-empty subsets.
The set $S = \{1, 2, \ldots, 2026\}$ has 2026 elements, so it has 2^{2026} subsets in total, including the empty set.
Removing the empty set (since M is required to be non-empty) leaves $2^{2026} - 1$ equally likely choices for M.
This is the sample space size.

Step 2: Split the elements into odd and even.
Among $1, 2, \ldots, 2026$, exactly half are odd and half are even, since 2026 is even. So there are 1013 odd numbers $(1, 3, 5, \ldots, 2025)$ and 1013 even numbers $(2, 4, \ldots, 2026)$.

Step 3: Use the fact that a product is even unless every factor is odd.

The product of the elements of M is odd only when M contains no even number at all, that is, when M is a non-empty subset made entirely of odd numbers. If M contains even one even number, the whole product becomes even. So it is easier to first count the "product is odd" case and subtract from the total.

Step 4: Count subsets whose product is odd.
A subset with product odd must be a non-empty subset of the 1013 odd numbers only, chosen from any subset of these 1013 odd elements (with even numbers excluded entirely). There are 2^{1013} subsets of the odd numbers, and removing the empty one leaves $2^{1013} - 1$ non-empty all-odd subsets.

Step 5: Compute the probability that the product is even.

$$P(\text{even}) = 1 - P(\text{odd}) = 1 - \frac{2^{1013} - 1}{2^{2026} - 1}$$

Combine over the common denominator:

$$P(\text{even}) = \frac{(2^{2026} - 1) - (2^{1013} - 1)}{2^{2026} - 1} = \frac{2^{2026} - 2^{1013}}{2^{2026} - 1}$$

Factor 2^{1013} out of the numerator, since $2^{2026} = 2^{1013} \cdot 2^{1013}$:

$$P(\text{even}) = \frac{2^{1013}(2^{1013} - 1)}{2^{2026} - 1}$$

Step 6: Why the other options are wrong.
Option (A) and (B) use 2^{2026} as the denominator, but the sample space excludes the empty set, so the correct denominator must be $2^{2026} - 1$, not 2^{2026}.
Option (D) is actually the probability of the complement event (product odd, with the empty-set case wrongly folded in), not the event asked for.

Final Answer:

$$\boxed{P(\text{product is even}) = \frac{2^{1013}(2^{1013} - 1)}{2^{2026} - 1}}$$

Quick Tip: The product is odd only when every chosen element is odd. Count non-empty all-odd subsets, subtract that fraction from 1, keeping the denominator as $2^{2026} - 1$ since M must be non-empty.

20. Suppose that a computer program provides a non-negative and integer-valued random solution to the equation $n_1 + n_2 + n_3 + n_4 = 20$.
Which of the following is the probability that all of n_1, n_2, n_3, n_4 in the provided solution are positive?

(A) $\binom{19}{3}/\binom{23}{3}$

(B) $\binom{20}{4}/\binom{24}{4}$

(C) $\binom{20}{3}/\binom{23}{3}$

(D) $\binom{19}{4}/\binom{24}{4}$

Correct Answer: (A) $\binom{19}{3}/\binom{23}{3}$

Solution:

Step 1: Recall the stars and bars formula.

The number of non-negative integer solutions to $x_1 + x_2 + \cdots + x_r = n$ is $\binom{n+r-1}{r-1}$. This counts the ways to place n identical items into r labeled bins.

Step 2: Count the total sample space.

Here $n = 20$ and $r = 4$ (the variables n_1, n_2, n_3, n_4). The total number of non-negative integer solutions is:

$$\binom{20+4-1}{4-1} = \binom{23}{3}$$

Every one of these solutions is treated as equally likely, since the program picks a "random" solution.

Step 3: Count the solutions where all four variables are positive.

If every $n_i \geq 1$, substitute $m_i = n_i - 1 \geq 0$. Then:

$$(m_1 + 1) + (m_2 + 1) + (m_3 + 1) + (m_4 + 1) = 20 \implies m_1 + m_2 + m_3 + m_4 = 16$$

Now this is a non-negative integer equation with $n = 16$ and $r = 4$, so the number of solutions is:

$$\binom{16+4-1}{4-1} = \binom{19}{3}$$

Step 4: Form the probability.

$$P(\text{all positive}) = \frac{\binom{19}{3}}{\binom{23}{3}}$$

Step 5: Why the other options are wrong.

Option (B) and (D) use $\binom{n}{4}$ type terms, but the "choose" part of stars and bars for 4 variables should use $r - 1 = 3$ as the lower index, not 4, since we are placing 3 dividers among the items, not 4.

Option (C) uses $\binom{20}{3}$ in the numerator, but that is the count for the total non-negative case restated with a wrong shift, it does not correctly represent the "all positive" restriction on the sum equal to 20.

Final Answer:

$$\boxed{P(\text{all positive}) = \frac{\binom{19}{3}}{\binom{23}{3}}}$$

Quick Tip: Total non-negative solutions use $\binom{23}{3}$ (stars and bars with 20 items, 4 bins). For all positive, substitute $m_i = n_i - 1$ and solve $\sum m_i = 16$, giving $\binom{19}{3}$.

21. Let $M = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$ be a 2×2 matrix, where $\theta = \frac{2\pi}{5}$, and $I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.
Which of the following options is equal to M^{2026}?

(A) M^2

(B) M

(C) M^{-1}

(D) I_2

Correct Answer: (B) M

Solution:

Step 1: Recognize what M represents.
The matrix $M = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$ is the standard 2D rotation matrix, it rotates any vector counter-clockwise by angle θ about the origin.

Step 2: Use the key property of rotation matrices.
Applying the rotation matrix k times in a row rotates a vector by $k\theta$ in total, so M^k is itself the rotation matrix for angle $k\theta$:

$$M^k = \begin{pmatrix} \cos(k\theta) & -\sin(k\theta) \\ \sin(k\theta) & \cos(k\theta) \end{pmatrix}$$

Also, since angles that differ by a full 2π (a complete revolution) give the exact same rotation, M^k only depends on $k\theta$ modulo 2π.

Step 3: Compute 2026θ modulo 2π.
Here $\theta = \frac{2\pi}{5}$, so:

$$2026\,\theta = 2026 \times \frac{2\pi}{5} = \frac{2026}{5} \times 2\pi$$

Write $2026 = 5 \times 405 + 1$, so $\frac{2026}{5} = 405 + \frac{1}{5}$. Then:

$$2026\,\theta = \left(405 + \frac{1}{5}\right) \times 2\pi = 405 \times 2\pi + \frac{2\pi}{5}$$

The term $405 \times 2\pi$ is exactly 405 full revolutions, which brings us back to the same direction, so it contributes nothing extra to the rotation. What remains is just $\frac{2\pi}{5} = \theta$.

Step 4: Conclude the value of M^{2026}.
Since 2026θ and θ represent the same rotation angle (modulo 2π), we get:

$$M^{2026} = M^1 = M$$

Step 5: Why the other options are wrong.
(A) M^2 would only be correct if $2026 \equiv 2 \pmod 5$, but we found $2026 \equiv 1 \pmod 5$, so this is wrong.
(C) M^{-1} is the rotation by $-\theta$ (or equivalently 4θ, since $M^5 = I_2$ means $M^{-1} = M^4$), which does not match $k \equiv 1 \pmod 5$.
(D) $I_2 = M^0$ (or M^5) would require $2026 \equiv 0 \pmod 5$, but 2026 leaves remainder 1, not 0, when divided by 5.

Final Answer:

$$\boxed{M^{2026} = M}$$

Quick Tip: M is a rotation matrix by θ, so M^k is rotation by $k\theta$, which only depends on $k\theta \bmod 2\pi$, i.e., on $k \bmod 5$ here. Find $2026 \bmod 5$.

22. Consider a set $S_1 = \{x = (x_1, x_2, x_3)^T \in \mathbb{R}^3 \mid x^T x \leq 16\}$. Let S_2 be another set which is a subspace of $\mathbb{R}^3$ with dimension two.
Which of the following gives the area of $S_1 \cap S_2$?

(A) 16π

(B) 4π

(C) $4\pi^2$

(D) $16\pi^2$

Correct Answer: (A) 16π

Solution:

Step 1: Identify what S_1 is.
$x^T x = x_1^2 + x_2^2 + x_3^2$, which is just the squared distance of the point x from the origin. So $S_1 = \{x \in \mathbb{R}^3 \mid x_1^2 + x_2^2 + x_3^2 \leq 16\}$ is exactly the solid ball of radius $\sqrt{16} = 4$ centered at the origin.

Step 2: Identify what S_2 is.

A subspace of $\mathbb{R}^3$ always contains the origin (since it must be closed under scalar multiplication, and multiplying by 0 gives the zero vector). A subspace of dimension two is therefore a flat plane passing through the origin.

Step 3: Understand the intersection geometrically.
S_1 is a solid ball centered at the origin with radius 4, and S_2 is a plane that passes exactly through that same center (the origin). When a plane cuts through the exact center of a ball, the cross-section formed is a flat disk whose radius equals the ball's own radius, because every point of the plane that is within distance 4 of the center (which is also on the plane) satisfies the ball's inequality.

Step 4: State the shape and compute its area.
So $S_1 \cap S_2$ is a disk of radius 4 lying in the plane S_2. The area of a disk of radius r is πr^2:

$$\text{Area} = \pi(4)^2 = 16\pi$$

Step 5: Why the other options are wrong.
(B) 4π would be the area of a disk of radius 2, which does not match a ball of radius 4.
(C) and (D) involve π^2, which does not arise in a plain 2D area formula, these look like errors from squaring π or confusing area with some other quantity like surface area of a sphere in three dimensions.

Final Answer:

$$\boxed{\text{Area}(S_1 \cap S_2) = 16\pi}$$

Quick Tip: S_1 is a solid ball of radius 4 centered at the origin, S_2 is a plane through the origin. Their intersection is a disk of radius 4, so use area $= \pi r^2$.

23. In the following table, the Task column lists a few tasks related to machine learning. The Algorithm column lists a few algorithms.
Each entry "t" from the Task column is to be matched with an appropriate entry "a" from the Algorithm column such that the task "t" can be solved using the algorithm "a". Denote such a match as t:a.

Task	Algorithm
T1 - Clustering	A1 - Markov Chain Monte Carlo
T2 - Classification	A2 - K-Medoid
T3 - Sampling	A3 - Linear Discriminant Analysis
T4 - Feature Extraction	A4 - Naive Bayes

Which of the following options is/are the correct matching(s)?

(A) T1:A4, T2:A3, T3:A1, T4:A2

(B) T1:A2, T2:A4, T3:A1, T4:A3

(C) T1:A3, T2:A4, T3:A1, T4:A2

(D) T1:A4, T2:A2, T3:A1, T4:A3

Correct Answer: (B) T1:A2, T2:A4, T3:A1, T4:A3

Solution:

Step 1: Recall what each algorithm actually does.

K-Medoid is a partitioning algorithm that groups similar data points into clusters around representative "medoid" points, it is a clustering method.

Naive Bayes is a probabilistic classifier that assigns a class label to a data point using Bayes' theorem with a "naive" independence assumption between features, it is a classification method.

Markov Chain Monte Carlo (MCMC) is a family of algorithms used to draw samples from a complicated probability distribution when direct sampling is hard, it is a sampling method.

Linear Discriminant Analysis (LDA) projects data onto a lower-dimensional space chosen to best separate classes, it is primarily used as a feature extraction/dimensionality reduction method (though it can also assist classification).

Step 2: Match each task to the algorithm designed for it.

T1 - Clustering needs an algorithm that groups unlabeled data by similarity, this is K-Medoid (A2). So T1:A2.

T2 - Classification needs an algorithm that assigns class labels, this is Naive Bayes (A4). So T2:A4.

T3 - Sampling needs an algorithm for drawing samples from a distribution, this is MCMC (A1). So T3:A1.

T4 - Feature Extraction needs an algorithm for reducing/transforming features, this is LDA (A3). So T4:A3.

Step 3: Compare with each option.

Option (A), T1:A4 wrongly pairs Clustering with Naive Bayes (a classifier, not a clustering method), so (A) is wrong.

Option (B), T1:A2, T2:A4, T3:A1, T4:A3, matches exactly what we derived in Step 2.

Option (C), T1:A3 wrongly pairs Clustering with LDA (a feature extraction tool, not a clustering method), so (C) is wrong.

Option (D), T2:A2 wrongly pairs Classification with K-Medoid (a clustering method, not a classifier), so (D) is wrong.

Final Answer:

Only option (B) gives the correct matching for all four tasks.

Option (B): T1:A2, T2:A4, T3:A1, T4:A3

Quick Tip: K-Medoid clusters, Naive Bayes classifies, MCMC samples, and LDA extracts/reduces features. Match each task to the algorithm built for that exact purpose.

24. Sentence X is said to entail Sentence Y if whenever X is $TRUE$, Y also must hold $TRUE$.
Which of the following statements is/are correct if X entails Y?

(A) $X \Rightarrow Y$

(B) $X \wedge \neg Y$ is $FALSE$

(C) if X then Y

(D) if Y then X

Correct Answer: (A) $X \Rightarrow Y$; (B) $X \wedge \neg Y$ is $FALSE$; (C) if X then Y

Solution:

Step 1: Restate the definition of entailment in logic form.
X entails Y means: whenever X is $TRUE$, Y must also be $TRUE$. There is no case where X is $TRUE$ and Y is $FALSE$ at the same time.

Step 2: Check option (A), $X \Rightarrow Y$.
The logical implication $X \Rightarrow Y$ is defined to be $FALSE$ only in the one case where X is $TRUE$ and Y is $FALSE$, and $TRUE$ in every other case. This is exactly the same condition as "whenever X is $TRUE$, Y is $TRUE$". So $X \Rightarrow Y$ correctly captures entailment. Option (A) is correct.

Step 3: Check option (B), $X \wedge \neg Y$ is $FALSE$.
$X \wedge \neg Y$ is $TRUE$ exactly when X is $TRUE$ and Y is $FALSE$ simultaneously, this is the one forbidden case under entailment. Saying $X \wedge \neg Y$ is $FALSE$ is just another way of saying this forbidden case never happens, which is the definition of entailment itself. Option (B) is correct, and in fact it is logically identical to $X \Rightarrow Y$.

Step 4: Check option (C), "if X then Y".
In plain English, "if X then Y" is the natural-language phrasing of the material implication $X \Rightarrow Y$, it means the same thing: whenever X holds, Y must hold too. Option (C) is correct, it is just a restatement of (A) in words.

Step 5: Check option (D), "if Y then X".
This is the converse of the implication, $Y \Rightarrow X$. Entailment only guarantees that X being $TRUE$ forces Y to be $TRUE$, it says nothing about what happens if Y is $TRUE$, X could very well be $FALSE$ in that case. For example, "it is raining" entails "the ground is wet", but "the ground is wet" does not entail "it is raining" (the ground could be wet for other reasons). So option (D) does not follow from X entailing Y, it is incorrect.

Final Answer:
Options (A), (B) and (C) are all correct restatements of "X entails Y", while (D) is the converse and is not implied.

Options (A), (B), (C)

Quick Tip: Entailment rules out only $X = TRUE, Y = FALSE$. Check each statement against this and note that "if Y then X" is the converse, which is not guaranteed.

25. You are given the following Pre-order and In-order traversals of a Binary Tree ***T*** with nodes E, F, G, P, Q, R, S.

Pre-order: P Q S E R F G
In-order: S Q E P F R G

Which of the following statements is/are true about the Binary Tree ***T***?

(A) Node P is the root of T

(B) The Post-order traversal of T is: S E Q F G R P

(C) Node Q has only one child

(D) The left subtree of node R contains the node G

Correct Answer: (A) Node P is the root of T ; (B) The Post-order traversal of T is: S E Q F G R P

Solution:

Step 1: Find the root using the pre-order list.
In a pre-order traversal, the first node visited is always the root. The pre-order list is P Q S E R F G, so the root of the tree is P.

Step 2: Split the in-order list around the root.
In the in-order list S Q E P F R G, everything left of P belongs to the left subtree and everything right of P belongs to the right subtree. This gives left subtree in-order = S Q E and right subtree in-order = F R G.

Step 3: Split the pre-order list using the subtree sizes.
The left subtree has 3 nodes, so the next 3 entries after P in the pre-order list belong to it: Q S E. The remaining entries R F G belong to the right subtree.

Step 4: Rebuild the left subtree (nodes S, Q, E).
Left subtree pre-order = Q S E, in-order = S Q E. The first entry of this pre-order, Q, is the root of this subtree. In its in-order list S Q E, S lies to the left of Q and E lies to the right of Q. Both sides have only 1 node, so Q has a left child S and a right child E.

Step 5: Rebuild the right subtree (nodes R, F, G).
Right subtree pre-order = R F G, in-order = F R G. The first entry, R, is the root. In its in-order list F R G, F lies to the left of R and G lies to the right of R, so R has a left child F and a right child G.

Step 6: Assemble the full tree.
P is the root.
P's left child is Q, whose left child is S and right child is E.
P's right child is R, whose left child is F and right child is G.

Step 7: Check each option.

(A) Node P is the root of T: from Step 1, P is indeed the root, so this is TRUE.
(B) Post-order visits the left subtree, then the right subtree, then the root. The left subtree (rooted at Q) gives S, E, Q. The right subtree (rooted at R) gives F, G, R. Followed by the root P, the full post-order is S E Q F G R P, which matches the option exactly, so this is TRUE.
(C) Node Q has children S and E, that is two children, not one, so this statement is FALSE.
(D) The left subtree of R is just the single node F. G is R's right child, not part of R's left subtree, so this statement is FALSE.

Final Answer:
Only statements (A) and (B) are true.

(A) and (B)

Quick Tip: Use the first element of the pre-order list as the root, then split the in-order list around it to find the left and right subtrees.

26. Consider two relations r and s defined on the relational schemas $R(A, B)$ and $S(E, C)$, respectively. A is the primary key of R and E is a foreign key of S referencing A in R.

Which of the following operations will NEVER violate the foreign key constraint?

(A) Inserting records into relation r
(B) Deleting records from relation s
(C) Deleting records from relation r
(D) Inserting records into relation s

Correct Answer: (A) Inserting records into relation r ; (B) Deleting records from relation s

Solution:

Step 1: Recall what the foreign key constraint requires.
E is a foreign key in S that references A, the primary key of R. This means every value of E that appears in relation s must also appear as a value of A in relation r. The constraint is checked on the value stored in S, against the values available in R.

Step 2: Check option (A), inserting into r.
Inserting a new record into r only adds new possible values of A. It does not remove or change any value of A that existing rows in s might be referencing, so it can never break the constraint. This operation is always safe.

Step 3: Check option (B), deleting from s.
Deleting a record from s only removes a row, it does not add a new E value that would need to match an A value in r. Since removing rows from the referencing relation cannot create a broken reference, this operation is always safe.

Step 4: Check option (C), deleting from r.
If a deleted row from r had an A value that some row in s references through E, that reference would now point to nothing, breaking the constraint. So this operation can violate the constraint, depending on the data. It is not always safe.

Step 5: Check option (D), inserting into s.
A new row inserted into s might have an E value that does not exist as any A value in r. That would break the constraint immediately. So this operation can also violate the constraint. It is not always safe.

Final Answer:
Only inserting into r (A) and deleting from s (B) can never violate the foreign key constraint.

$$\boxed{\text{(A) and (B)}}$$

Quick Tip: Think about which side of the reference, the referenced primary key or the referencing foreign key, each operation changes.

27. Let $f(x) = x^3 - 3x^2 + 2$ be a function defined on $(-1, 3]$.

Which of the following statements is/are correct?

(A) $f(x)$ has exactly two roots in $[-0.9, 0]$.

(B) $f(x)$ has a minimum at 2 only.

(C) $f(x)$ has maximum at 0 only.

(D) $f(x)$ has a root at 1.

Correct Answer: (B) $f(x)$ has a minimum at 2 only. ; (D) $f(x)$ has a root at 1.

Solution:

Step 1: Find the critical points using the first derivative.

$$f(x) = x^3 - 3x^2 + 2, \quad f'(x) = 3x^2 - 6x = 3x(x - 2)$$

Setting $f'(x) = 0$ gives $x = 0$ and $x = 2$. Both lie inside the domain $(-1, 3]$.

Step 2: Classify the critical points using the second derivative.

$$f''(x) = 6x - 6$$

At $x = 0$: $f''(0) = -6 < 0$, so $x = 0$ is a local maximum, with $f(0) = 2$.
At $x = 2$: $f''(2) = 6 > 0$, so $x = 2$ is a local minimum, with $f(2) = 8 - 12 + 2 = -2$.

Step 3: Find the roots of f(x).
Testing $x = 1$: $f(1) = 1 - 3 + 2 = 0$, so $x = 1$ is a root, confirming option (D).
Dividing out $(x - 1)$: $x^3 - 3x^2 + 2 = (x - 1)(x^2 - 2x - 2)$.
Solving $x^2 - 2x - 2 = 0$ with the quadratic formula gives $x = 1 \pm \sqrt{3}$, which is about 2.732 and -0.732.
So the three roots of $f(x)$ are approximately -0.732, 1, and 2.732.

Step 4: Check option (A), roots in [-0.9, 0].
Of the three roots, only -0.732 lies inside $[-0.9, 0]$. The roots at 1 and 2.732 fall outside this interval. So there is exactly one root there, not two, making (A) FALSE.

Step 5: Check option (B), minimum at 2 only.
On the domain $(-1, 3]$, the point $x = -1$ is excluded, so the value $f(-1) = -1 - 3 + 2 = -2$ is only an approached limit, never actually attained. The local minimum value -2 is genuinely reached only at $x = 2$. So (B) is TRUE.

Step 6: Check option (C), maximum at 0 only.
$f(0) = 2$ is the local maximum. But check the right endpoint: $f(3) = 27 - 27 + 2 = 2$, and $x = 3$ is included in the domain. So the value 2 is attained at both $x = 0$ and $x = 3$, meaning the maximum is not unique to $x = 0$. So (C) is FALSE.

Step 7: Check option (D), root at 1.
Already shown in Step 3, $f(1) = 0$, so (D) is TRUE.

Final Answer:
Statements (B) and (D) are correct.

(B) and (D)

Quick Tip: Find the critical points with the first derivative, classify them with the second derivative, and do not forget to check the domain endpoints.

28. Suppose a random variable Z follows $Normal(\mu = 0, \sigma^2 = 1)$ distribution with probability density function $g(z)$ and cumulative distribution function $G(z)$. Another random variable Y follows t_1 distribution with probability density function $h(y)$ and cumulative distribution function $H(y)$. Let c be the positive real number for which $g(c) = h(c)$.

Which of the following statements is/are correct?

(A) $G(0) = H(0)$

(B) $G(c) < H(c)$

(C) $G(-c) < H(-c)$

(D) $g(0) = h(0)$

Correct Answer: (A) $G(0) = H(0)$; (C) $G(-c) < H(-c)$

Solution:

Step 1: Note the shape of both densities.

Z follows a standard normal distribution and Y follows a t_1 distribution (the same as a standard Cauchy distribution). Both densities $g(z)$ and $h(y)$ are symmetric about 0, bell-shaped, and strictly decreasing as you move away from 0 in either direction.

Step 2: Compare the peak heights at 0.

$$g(0) = \frac{1}{\sqrt{2\pi}} \approx 0.399, \qquad h(0) = \frac{1}{\pi(1+0^2)} = \frac{1}{\pi} \approx 0.318$$

So $g(0) > h(0)$, meaning statement (D), $g(0) = h(0)$, is FALSE. The normal curve is taller at the centre than the Cauchy curve.

Step 3: Use symmetry to compare the CDFs at 0.

Both densities are symmetric about 0, so exactly half the probability mass lies on each side of 0 for both distributions. This gives $G(0) = 0.5$ and $H(0) = 0.5$, so $G(0) = H(0)$. Statement (A) is TRUE.

Step 4: Understand where c comes from.

The normal density falls off very fast (like $e^{-z^2/2}$) while the Cauchy density falls off slowly (like $1/y^2$), so the Cauchy has much heavier tails. Since $g(0) > h(0)$ but the Cauchy eventually overtakes the normal far from 0, the two curves must cross at some positive point, and this crossing point is c. For $0 < y < c$, $g(y) > h(y)$ (normal is higher, closer to the centre); for $y > c$, $h(y) > g(y)$ (Cauchy's heavy tail is higher).

Step 5: Check option (B), G(c) < H(c).

$$G(c) - G(0) = \int_0^c g(y)\,dy, \qquad H(c) - H(0) = \int_0^c h(y)\,dy$$

Since $g(y) > h(y)$ throughout $(0, c)$, the left integral is larger than the right one. Because $G(0) = H(0) = 0.5$, this means $G(c) > H(c)$, not less than. So statement (B) is FALSE.

Step 6: Check option (C), G(-c) < H(-c).

By symmetry of both distributions about 0, $G(-c) = 1 - G(c)$ and $H(-c) = 1 - H(c)$. Since $G(c) > H(c)$ from Step 5, subtracting both sides from 1 flips the inequality: $1 - G(c) < 1 - H(c)$, which is $G(-c) < H(-c)$. So statement (C) is TRUE.

Final Answer:
Statements (A) and (C) are correct.

(A) and (C)

Quick Tip: Remember both densities are symmetric about 0, so their CDFs must equal 0.5 there regardless of the shape of the tails.

29. Consider that for a supervised learning task, the objective function being minimized is $f_w(x) = wx$, where $x \in \mathbb{R}$ is the input and $w \in \mathbb{R}$ is the parameter. Stochastic Gradient Descent with a learning rate of 0.10 is used for parameter updates.

Suppose that at the end of iteration i, the value of w becomes 10.00.

Let $x = 10.00$ be the input for iteration $(i + 1)$.

The value of w at the end of iteration $(i + 1)$ is _____. (Rounded off to two decimal places)

Correct Answer: 9.00

Solution:

Step 1: Write down the SGD update rule.

Stochastic Gradient Descent updates the parameter using $w_{new} = w_{old} - \eta \cdot \frac{\partial f_w(x)}{\partial w}$, where η is the learning rate.

Step 2: Compute the gradient of the objective function.

The objective function is $f_w(x) = wx$. Taking the partial derivative with respect to w:

$$\frac{\partial f_w(x)}{\partial w} = x$$

Step 3: Plug in the known values.

At the end of iteration i, $w = 10.00$. The input for iteration $(i + 1)$ is $x = 10.00$, so the gradient at this step is 10.00. The learning rate is $\eta = 0.10$.

Step 4: Apply the update rule.

$$w_{new} = w_{old} - \eta \cdot x = 10.00 - 0.10 \times 10.00$$

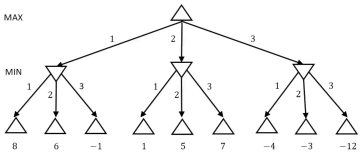
$$w_{new} = 10.00 - 1.00 = 9.00$$

Final Answer:
The value of w at the end of iteration $(i+1)$ is 9.00.

$$\boxed{9.00}$$

Quick Tip: The gradient of wx with respect to w is just x, so the update rule reduces to w minus the learning rate times x.

30. Consider the game tree for a two-player turn-taking minimax game as shown in the figure. The value of a terminal node represents the utility of the game state if the game ends there. The numbers written next to the edges denote the strategies.



There are two players MAX and MIN. At any particular state of the game, MAX prefers to move to a state of maximum value. On the other hand, MIN prefers to move to a state of minimum value.

Suppose MAX starts the game at the root and has three strategies: 1, 2 and 3. Next, MIN plays and also has three strategies: 1, 2 and 3. The game ends there. Both players always take optimal strategies throughout the game.

At the root, the best strategy for MAX is _____. (Answer in integer)

Correct Answer: 2

Solution:

Step 1: Understand the tree structure.
MAX moves first at the root with three choices, labelled strategies 1, 2 and 3, each leading to a MIN node. Each MIN node then has three choices, also labelled 1, 2 and 3, each leading to a terminal (leaf) node with a fixed utility value.

Step 2: Evaluate the MIN node reached by MAX's strategy 1.
Its children have values 8, 6, -1 (for MIN's strategies 1, 2, 3). MIN always picks the move with the minimum value, so this node's value is $\min(8, 6, -1) = -1$.

Step 3: Evaluate the MIN node reached by MAX's strategy 2.

Its children have values 1, 5, 7. MIN picks the minimum: $\min(1, 5, 7) = 1$.

Step 4: Evaluate the MIN node reached by MAX's strategy 3.
Its children have values -4, -3, -12. MIN picks the minimum: $\min(-4, -3, -12) = -12$.

Step 5: Back up the values to the root and let MAX choose.
The root now sees three backed-up values: -1 (from strategy 1), 1 (from strategy 2), and -12 (from strategy 3). MAX always picks the move with the maximum value: $\max(-1, 1, -12) = 1$, which is reached through strategy 2.

Final Answer:
The best strategy for MAX at the root is strategy 2, which guarantees a game value of 1.

2

Quick Tip: Evaluate each MIN node first by taking the minimum of its children, then let MAX pick the largest of those results.

31. Let A be a sorted array containing 1000 distinct integers. You perform a recursive binary search on A to find an element y. Suppose each comparison checks whether the middle element computed during the current recursive step is equal to, less than, or greater than y.

The maximum number of comparisons that may have to be performed if y is not an element of A is _____. (Answer in integer)

Correct Answer: 10

Solution:

Step 1: Recall how a three-way comparison binary search works.
At each recursive step, the search looks at the middle element of the current sub-array and checks whether it is equal to, less than, or greater than the target y. If equal, the search stops (found). If less, the search continues on the right half. If greater, it continues on the left half. Since y is not present in A, the "equal" case never triggers, and the search keeps halving the search space until nothing is left to search.

Step 2: Model the search size after each comparison.
Starting with $n = 1000$ elements, each comparison narrows the search down to roughly half the elements (rounding down), because one element is examined (and ruled out as unequal) and the rest is split into two halves of which only one is kept. This continues until the sub-array size becomes 0, meaning y is confirmed absent.

Step 3: Count how many times 1000 can be halved.
The number of comparisons needed in the worst case to search n elements this way is:

$$\text{comparisons} = \lfloor \log_2 n \rfloor + 1$$

Substituting $n = 1000$:

$$\log_2 1000 \approx 9.966$$

$$\lfloor 9.966 \rfloor + 1 = 9 + 1 = 10$$

Step 4: Sanity check with a direct doubling argument.
$2^9 = 512$ and $2^{10} = 1024$. Since $512 < 1000 \leq 1024$, it takes 10 halving steps (comparisons) in the worst case to shrink a search space of 1000 elements down to nothing, confirming the formula.

Final Answer:
The maximum number of comparisons needed is 10.

$$\boxed{10}$$

Quick Tip: Each comparison roughly halves the search space, so count how many times 1000 can be halved before nothing is left.

32. In a relational database, a B+ Tree Index is to be constructed for a relation on a key field. In a B+ Tree, a Node Pointer points to a sub-tree and a Data Record Pointer points to a block of database records.

Let, Node size = 4096 bytes, Node Pointer size = 10 bytes, Search Key Field size = 11 bytes and Data Record Pointer size = 12 bytes.

The maximum number of Node Pointers that can be present in a non-leaf node of the B+ Tree is _____. (Answer in integer)

Correct Answer: 195

Solution:

Step 1: Recall the structure of a non-leaf B+ Tree node.
A non-leaf node with n Node Pointers holds n pointers and $(n-1)$ search key values arranged between them (the keys are used to decide which pointer to follow). It does not store Data Record Pointers, those only appear in leaf nodes.

Step 2: Write the space constraint.
Total space used by n node pointers and $(n-1)$ keys must fit inside the node size:

$$n \times (\text{Node Pointer size}) + (n-1) \times (\text{Search Key size}) \leq \text{Node size}$$

$$n \times 10 + (n-1) \times 11 \leq 4096$$

Step 3: Simplify the inequality.

$$10n + 11n - 11 \leq 4096$$

$$21n \leq 4107$$

$$n \leq 195.57$$

Step 4: Take the largest integer satisfying the constraint.
Since n must be a whole number, $n = 195$.

Step 5: Verify by substitution.
For $n = 195$: $195 \times 10 + 194 \times 11 = 1950 + 2134 = 4084$ bytes, which fits within 4096 bytes.
For $n = 196$: $196 \times 10 + 195 \times 11 = 1960 + 2145 = 4105$ bytes, which exceeds 4096 bytes, so 196 pointers do not fit.

Final Answer:
The maximum number of Node Pointers in a non-leaf node is 195.

$$\boxed{195}$$

Quick Tip: A non-leaf node with n pointers holds n-1 keys between them; set up the space inequality and solve for the largest integer n.

33. The number of bijections $f(\cdot)$ from the set $S = \{1, 2, 3, 4\}$ to itself such that $f(f(n)) = n$, for all $n \in S$, is _____. (Answer in integer)

Correct Answer: 10

Solution:

Step 1: Understand what the condition means.
A function f is a bijection from S to itself with f(f(n)) = n for every n in S. Applying f twice always brings you

back to the start, so f must be its own inverse. A permutation with this property is called an involution. In an involution every element is either a fixed point, where f(n) = n, or part of a pair that swaps with each other, where f(a) = b and f(b) = a for a not equal to b. No cycle of length 3 or more is allowed, because a 3-cycle needs three applications of f to return to the starting element, not two.

Step 2: Split into cases by how many swapped pairs are used.
The set S has 4 elements. A swapped pair uses up exactly 2 elements, so the number of pairs used can be 0, 1, or 2 (using 3 pairs would need 6 elements).

Step 3: Count each case.
Case with 0 pairs: every element is fixed. This is just the identity function, so there is exactly 1 way.
Case with 1 pair: pick 2 of the 4 elements to swap, and leave the other 2 fixed. The number of ways to choose which 2 elements swap is $\binom{4}{2} = 6$.
Case with 2 pairs: all 4 elements are split into two swapped pairs, with no element left fixed. The number of ways to split 4 labeled elements into 2 unordered pairs is 3, since for {1,2,3,4} the possible pairings are {1,2}{3,4}, {1,3}{2,4}, and {1,4}{2,3}.

Final Answer:
Add up the three cases: $1 + 6 + 3 = 10$.

$$\boxed{10}$$

Quick Tip: f(f(n)) = n means f is an involution: count fixed points plus swapped pairs among 4 elements.

34. Let X be an exponentially distributed random variable with mean $\lambda(> 0)$. If $P(X > 5) = 0.35$, then the conditional probability $P(X > 10 \mid X > 5)$ is _____. (Rounded off to two decimal places)

Correct Answer: 0.35

Solution:

Step 1: Recall the memoryless property of the exponential distribution.
The exponential distribution has a special property called memorylessness: for any $s, t \geq 0$, $P(X > s + t \mid X > s) = P(X > t)$. This says that once you know X has already exceeded s, the chance it goes on to exceed $s + t$ more is the same as the chance a fresh copy of X exceeds t from zero. It does not matter how long you have already waited.

Step 2: Match the numbers to the property.
We want $P(X > 10 \mid X > 5)$. Here $s = 5$ and $s + t = 10$, so $t = 5$. By memorylessness,

$$P(X > 10 \mid X > 5) = P(X > 5 + 5 \mid X > 5) = P(X > 5)$$

Step 3: Substitute the given value.

We are told $P(X > 5) = 0.35$, so the conditional probability equals 0.35 directly, with no need to even find λ.

Step 4: Verify using the exponential formula.

For X exponential, $P(X > x) = e^{-x/\lambda}$. From $P(X > 5) = e^{-5/\lambda} = 0.35$, squaring gives $P(X > 10) = e^{-10/\lambda}$ $= (e^{-5/\lambda})^2 = 0.35^2 = 0.1225$. Since $X > 10$ automatically means $X > 5$,

$$P(X > 10 \mid X > 5) = \frac{P(X > 10)}{P(X > 5)} = \frac{0.1225}{0.35} = 0.35$$

This matches Step 3, confirming the answer.

Final Answer:

$$\boxed{0.35}$$

Quick Tip: Use the memoryless property of the exponential distribution: P(X>10|X>5) reduces to P(X>5).

35. The value of $\sum_{i=0}^{\infty}\sum_{j=1}^{\infty} 2^{-i}3^{-j}$ is _____. (Answer in integer)

Correct Answer: 1

Solution:

Step 1: Split the double sum into a product.

The summand $2^{-i}3^{-j}$ is a product of a term that depends only on i and a term that depends only on j. Since i and j range over independent, separate index sets, the double sum splits into a product of two single sums:

$$\sum_{i=0}^{\infty}\sum_{j=1}^{\infty} 2^{-i}3^{-j} = \left(\sum_{i=0}^{\infty} 2^{-i}\right)\left(\sum_{j=1}^{\infty} 3^{-j}\right)$$

Step 2: Evaluate the first geometric series.

$\sum_{i=0}^{\infty} 2^{-i}$ is a geometric series starting at $i = 0$ with first term 1 and common ratio 1/2. Its sum is

$$\sum_{i=0}^{\infty} 2^{-i} = \frac{1}{1 - 1/2} = 2$$

Step 3: Evaluate the second geometric series.

$\sum_{j=1}^{\infty} 3^{-j}$ is a geometric series starting at $j = 1$ with first term 1/3 and common ratio 1/3. Its sum is

$$\sum_{j=1}^{\infty} 3^{-j} = \frac{1/3}{1 - 1/3} = \frac{1/3}{2/3} = \frac{1}{2}$$

Step 4: Multiply the two results.

$$2 \times \frac{1}{2} = 1$$

Final Answer:

$$\boxed{1}$$

Quick Tip: Split the double sum into a product of two independent geometric series, one in i and one in j.

36. Let four points in three-dimensional space be:
$P1 : [2, 3, -1]$, $P2 : [3, 1, 1]$, $P3 : [5, -2, 3]$ and $P4 : [3, 3, 3]$.
Hierarchical Agglomerative Clustering is used to cluster the above points. If Manhattan Distance is used as the distance metric during clustering, which of the following options indicates the two points that will be merged first?

(A) $P1, P2$

(B) $P2, P3$

(C) $P3, P4$

(D) $P2, P4$

Correct Answer: (D) $P2, P4$

Solution:

Step 1: Recall how the first merge in agglomerative clustering works.
Agglomerative clustering starts with every point as its own cluster, and at each step it merges the two clusters that are closest under the chosen distance metric. For the very first merge, this just means finding the pair of points with the smallest pairwise Manhattan distance, since no clusters have combined yet.

Step 2: Recall the Manhattan distance formula.
For two points $A = (a_1, a_2, a_3)$ and $B = (b_1, b_2, b_3)$, the Manhattan distance is

$$d(A, B) = |a_1 - b_1| + |a_2 - b_2| + |a_3 - b_3|$$

Step 3: Compute all 6 pairwise distances.
$d(P1, P2) = |2 - 3| + |3 - 1| + |-1 - 1| = 1 + 2 + 2 = 5$
$d(P1, P3) = |2 - 5| + |3 - (-2)| + |-1 - 3| = 3 + 5 + 4 = 12$
$d(P1, P4) = |2 - 3| + |3 - 3| + |-1 - 3| = 1 + 0 + 4 = 5$

$d(P2, P3) = |3-5| + |1-(-2)| + |1-3| = 2+3+2 = 7$
$d(P2, P4) = |3-3| + |1-3| + |1-3| = 0+2+2 = 4$
$d(P3, P4) = |5-3| + |-2-3| + |3-3| = 2+5+0 = 7$

Step 4: Pick the smallest distance.
The distances are 5, 12, 5, 7, 4, and 7. The smallest of these is 4, between P2 and P4, so this pair merges first. Option (A) has distance 5, not the minimum. Options (B) and (C) both have distance 7. Option (D) has the true minimum of 4.

Final Answer:
P2 and P4 merge first. This is option (D).

Quick Tip: Compute all 6 pairwise Manhattan distances between the points and pick the smallest one.

37. Which of the following statements is true for Ridge Regression?

(A) The regularizer in the objective function of Ridge Regression is used to guard against scenarios where the model works well for the test data, but poorly for the training data.

(B) The regularizer of Ridge Regression uses L_1 norm.

(C) Ridge Regression aims to reduce the number of parameters that have negative values.

(D) The regularizer of Ridge Regression may increase the bias of the model, but it helps in reducing the variance in predictions.

Correct Answer: (D) The regularizer of Ridge Regression may increase the bias of the model, but it helps in reducing the variance in predictions.

Solution:

Step 1: Recall what Ridge Regression's regularizer does.
Ridge Regression adds a penalty term $\lambda \sum w_i^2$ to the ordinary least squares cost function, where w_i are the model weights. This is the squared L_2 norm of the weight vector, not the L_1 norm. Adding this penalty discourages the weights from growing too large.

Step 2: Connect the regularizer to bias and variance.
Shrinking the weights makes the model less sensitive to small fluctuations in the training data, which lowers its variance since it will not swing wildly for different training samples. At the same time, shrinking the weights away from their unrestricted least-squares values introduces some systematic error, which raises the bias slightly. This is the classic bias-variance trade-off that regularization exploits to fight overfitting.

Step 3: Check each option against this.
Option (A) describes the opposite of overfitting: overfitting is when a model works well on training data but poorly on test data, not the other way around, so (A) is wrong.
Option (B) is wrong because Ridge Regression's penalty is the L_2 norm (sum of squared weights); the L_1 norm penalty belongs to Lasso Regression, not Ridge.

Option (C) is wrong because Ridge Regression shrinks weights toward zero but almost never sets them exactly to zero, so it does not reduce the count of negative-valued parameters, and that is not even its goal; sparsity and feature elimination is a property of Lasso, not Ridge.
Option (D) correctly states that the regularizer trades a small increase in bias for a reduction in variance, which is exactly why Ridge Regression helps control overfitting.

Final Answer:
Option (D) is the true statement about Ridge Regression.

Quick Tip: Ridge Regression's penalty is the squared L2 norm; think about what shrinking weights does to bias and variance.

38. Assume that a Creative (C) person will Succeed (S) if the person is also Disciplined (D), but will not succeed otherwise. Now, consider the following statements:
(i) $C \wedge S \Leftrightarrow D$
(ii) $C \Rightarrow (S \Leftrightarrow D)$
(iii) $C \Leftrightarrow ((D \Rightarrow S) \vee \neg S)$
Which of the following options is correct?

(A) Both (i) and (ii) are TRUE

(B) Only (ii) is TRUE

(C) Both (ii) and (iii) are TRUE

(D) Only (iii) is TRUE

Correct Answer: (B) Only (ii) is TRUE

Solution:

Step 1: Translate the given rule into logic.
The rule says: if a person is Creative, then they succeed exactly when they are also Disciplined. In symbols, this is $C \Rightarrow (S \Leftrightarrow D)$, which is precisely statement (ii). So statement (ii) is just a direct, correct restatement of the given rule, and must be TRUE.

Step 2: List the valid combinations of C, S, D allowed by the rule.
When C is false, the rule places no restriction, so S and D can be anything: this gives 4 valid combinations $(C=0, S=0, D=0)$, $(0,0,1)$, $(0,1,0)$, $(0,1,1)$.
When C is true, we need $S \Leftrightarrow D$, so S and D must match: this gives 2 more valid combinations $(C=1, S=0, D=0)$ and $(1,1,1)$.
In total there are 6 valid worlds consistent with the rule.

Step 3: Test statement (i), $C \wedge S \Leftrightarrow D$, against these worlds.
Take the valid world $(C=0, S=0, D=1)$. Here $C \wedge S = 0$ but $D = 1$, so $C \wedge S \Leftrightarrow D$ evaluates to false. Since (i) fails in a world where the original rule still holds, (i) is not always TRUE, so (i) is FALSE overall.

Step 4: Test statement (iii), $C \Leftrightarrow ((D \Rightarrow S) \vee \neg S)$, against these worlds.
Take the valid world $(C{=}0, S{=}0, D{=}0)$. Here $D \Rightarrow S$ is $0 \Rightarrow 0 = 1$, and $\neg S = 1$, so the right side is $1 \vee 1 = 1$. But $C = 0$, so $C \Leftrightarrow 1$ is false. Since (iii) fails in a valid world, (iii) is not always TRUE, so (iii) is FALSE overall.

Step 5: Conclude.
Only statement (ii) holds in every world consistent with the given rule; (i) and (iii) each fail in at least one such world.

Final Answer:
Only (ii) is TRUE, which is option (B).

Quick Tip: Rewrite the given rule as C => (S <=> D), then test each statement against the valid (C,S,D) combinations.

39. A recursive function in Python is given.

```
def mystery(n):
    if n <= 0:
        return 1
    else:
        return mystery(n-1) + mystery(n-2)
```

Now, consider the following function call:
`mystery(4)`
Assume that a typical runtime stack is used to manage function calls. Each function call is pushed onto the stack and removed only after it finishes execution.
Which of the following options denotes the total number of function calls (i.e., the total number of stack activations), including the initial call, to compute `mystery(4)`?

(A) 5
(B) 9
(C) 15
(D) 17

Correct Answer: (C) 15

Solution:

Step 1: Set up a formula for the total number of calls.
Let $T(n)$ be the total number of function-call activations made while computing $mystery(n)$, counting the call itself. For $n \leq 0$, the function returns immediately with no further recursive calls, so $T(n) = 1$ in that base case. For $n > 0$, the call makes 1 activation for itself, plus it calls $mystery(n-1)$ and $mystery(n-2)$, contributing all of their activations too. So

$$T(n) = 1 + T(n-1) + T(n-2), \quad n > 0$$

Step 2: Compute the base cases.
$T(0) = 1$ (base case, no recursive calls).
$T(-1) = 1$ (also a base case, since $-1 \leq 0$).

Step 3: Build up $T(1)$ through $T(4)$ using the recursion.
$T(1) = 1 + T(0) + T(-1) = 1 + 1 + 1 = 3$
$T(2) = 1 + T(1) + T(0) = 1 + 3 + 1 = 5$
$T(3) = 1 + T(2) + T(1) = 1 + 5 + 3 = 9$
$T(4) = 1 + T(3) + T(2) = 1 + 9 + 5 = 15$

Step 4: Double check by picturing the call tree shape.
mystery(4) calls *mystery*(3) and *mystery*(2); *mystery*(3) calls *mystery*(2) and *mystery*(1); and so on, exactly mirroring the Fibonacci recursion tree. Counting every node in this tree, including leaves that hit the base case, gives the same total of 15.

Step 5: Compare with the options.
5 would be $T(2)$, and 9 would be $T(3)$; both undercount by stopping the recursion too early. 17 does not match any correct partial sum of this recursion. 15 matches $T(4)$ computed above.

Final Answer:
The total number of function calls to compute *mystery*(4) is 15, which is option (C).

Quick Tip: Set up T(n) = 1 + T(n-1) + T(n-2) with T(n)=1 for n<=0, then build up to T(4).

40. Consider a directed graph $G = (V, E)$, where V is the finite set of vertices and E is the set of directed edges between the vertices. G may contain cycles but there is no self-loop. Further, G may not be strongly connected. Let G^R be the graph obtained by reversing the directions of all the edges in G without changing the set of vertices. Assume that Breadth First Search (BFS) or Depth First Search (DFS) from any given vertex v of a graph visits only the reachable vertices from v in that graph.
Which of the following statements must always be true, regardless of the structure of G?

(A) If u is a reachable vertex in the BFS of G^R from v, then u is also a reachable vertex in the DFS of G from v.

(B) In G^R, the BFS traversal from v will visit exactly the same set of vertices as the DFS from v in G.

(C) The order of vertices visited in the BFS of G^R from v is the reverse of the order of vertices visited in the DFS of G from v.

(D) If u is a reachable vertex in the DFS of G from v, then v is also a reachable vertex in the BFS of G^R from u.

Correct Answer: (D) If u is a reachable vertex in the DFS of G from v, then v is also a reachable vertex in the BFS of G^R from u.

Solution:

Step 1: Translate reachability into a path statement.
Saying "u is reachable from v in G" means there is a directed path $v \to \cdots \to u$ using edges of G, each edge pointing forward along the path. A BFS or DFS from v visits exactly the set of vertices reachable from v, by the assumption given in the question.

Step 2: Understand what reversing edges does to a path.
If there is a directed path from v to u in G, then reversing every edge on that path, which is exactly what forming G^R does to every edge of G, turns it into a directed path from u to v in G^R, with the same vertices visited in the opposite order. So: u reachable from v in G if and only if v reachable from u in G^R.

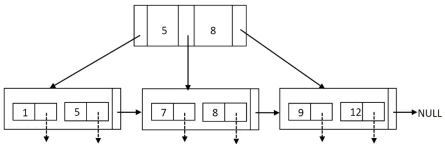
Step 3: Check option (D) against this fact.
Option (D) says: if u is reachable from v by DFS in G (i.e. there is a path $v \to u$ in G), then v is reachable from u by BFS in G^R (i.e. there is a path $u \to v$ in G^R). This is exactly the fact proven in Step 2, so option (D) must always be true, regardless of how G is shaped, whether it has cycles, or whether it is strongly connected.

Step 4: Rule out the other options with a small counterexample.
Take a simple graph with edges $v \to a$ and $b \to v$, so a is reachable from v in G, but nothing lies on a path from v toward b in G. In G^R the edges become $a \to v$ and $v \to b$, so from v, BFS in G^R reaches b, not a.
Option (A) fails: reachable sets from the same vertex v in G and in G^R can be entirely different (here $\{v, a\}$ versus $\{v, b\}$), so a vertex reachable in BFS of G^R from v need not be reachable via DFS of G from that same v.
Option (B) fails for the same reason: the reachable set from v in G is $\{v, a\}$, while the reachable set from v in G^R is $\{v, b\}$; these are different sets, so BFS of G^R from v does not visit the same vertices as DFS of G from v.
Option (C) fails too: since the visited sets can differ entirely, there is no meaningful "reverse of the order" relationship between them in general, and even when the sets coincide, BFS and DFS can visit vertices in different orders.

Final Answer:
Option (D) is the statement that must always be true.

Quick Tip: Reversing every edge on a path from v to u in G turns it into a path from u to v in G^R.

41. Consider a B+ Tree where the maximum number of key values in each leaf node is 2 and the maximum number of pointers in each non-leaf node is 3. Let the content of the B+ Tree be as shown in the figure.



Which of the following options denotes the key value(s) stored in the root node after inserting a key value 3 in the given B+ Tree?

(A) 5

(B) 8

(C) 3 and 5

(D) 3, 5 and 8

Correct Answer: (A) 5

Solution:

Step 1: Read off the tree structure and the search rule.

The leaf capacity is 2 keys and the non-leaf capacity is 3 pointers, so every internal node can hold at most 2 keys.

The current leaves are [1, 5], [7, 8] and [9, 12], joined left to right, and the root holds the two separator keys 5 and 8.

Reading the pointers against the leaves shows the rule used here: the first pointer covers keys ≤ 5, the middle pointer covers keys in $(5, 8]$, and the last pointer covers keys > 8.

Step 2: Find the leaf that receives the new key.

We need to insert 3. Since $3 \leq 5$, it goes down the first pointer into the leaf [1, 5].

Inserting 3 into that leaf gives [1, 3, 5], which now has 3 keys, one more than the leaf limit of 2.

Step 3: Split the overflowing leaf.

With 3 keys to share out, the left leaf keeps the first $\lceil 3/2 \rceil = 2$ keys, [1, 3], and the right leaf keeps what remains, [5].

The largest key of the left leaf, 3, is copied up into the parent as the new separator between these two leaves.

Step 4: Check whether the root also overflows.

Before the insert, the root [5, 8] already pointed to 3 children, the maximum allowed for a non-leaf node with pointer limit 3.

The leaf split adds one more child, so the root would need to hold 3 keys, [3, 5, 8], and point to 4 children, which breaks the pointer limit.

So the root itself has to split as well.

Step 5: Split the root.

For an internal (non-leaf) node split, the middle key is pushed up alone to form a new root; it is not left behind in either child, unlike a leaf split.

Of the three keys [3, 5, 8], the middle one is 5, so 5 moves up by itself.

The left internal node keeps key 3, pointing at leaves [1, 3] and [5]; the right internal node keeps key 8, pointing at leaves [7, 8] and [9, 12].

Final Answer:

The new root now holds only the single key 5, which matches option (A).

$$\boxed{\text{Root} = \{5\}}$$

Quick Tip: First find which leaf 3 falls into using the $\leq$ rule on the root separators, then check the leaf split, and finally check if the root itself now has too many pointers and must split too.

42. Consider the given relations X, Y and Z. The relation X has three columns P, Q and R. The relation Y has three columns P, Q and S. The relation Z has two columns P and T.

Relation X:

P	Q	R
P1	Q1	R1
P2	Q2	R2
P3	Q3	R2

Relation Y:

P	Q	S
P1	Q1	2
P1	Q2	5
P2	Q1	6
P3	Q3	1

Relation Z:

P	T
P1	T1
P3	T2
P4	T3
P4	NULL

Consider the relational algebra expression

$$\Pi_{P,R,S}\left[\left(\sigma_{(Q=Q3 \vee R=R2)}[X \bowtie Y]\right) \bowtie \left(\sigma_{(S>1)}[Y \bowtie Z]\right)\right]$$

where $\bowtie$ denotes natural join operation.

Which of the following options is the correct output for the given expression?

(A) Two rows (P1, R1, 2) and (P1, R1, 5)

(B) Three rows (P1, R1, 2), (P1, R1, 5) and (P2, R2, 6)

(C) One row (P1, R1, 2)

(D) Zero rows

Correct Answer: (D) Zero rows

Solution:

Step 1: Join X and Y on their shared columns.

X and Y share the columns P and Q, so $X \bowtie Y$ keeps only rows where both P and Q match.

X(P1,Q1,R1) matches Y(P1,Q1,2), giving (P1,Q1,R1,2). X(P2,Q2,R2) has no Y row with P=P2 and Q=Q2 (the only P2 row in Y is Q1), so it drops out. X(P3,Q3,R2) matches Y(P3,Q3,1), giving (P3,Q3,R2,1).

So $X \bowtie Y = \{(P1, Q1, R1, 2), (P3, Q3, R2, 1)\}$.

Step 2: Apply the selection $Q = Q3 \vee R = R2$.

For (P1,Q1,R1,2): Q is Q1, not Q3, and R is R1, not R2, so the condition is false and this row is dropped.

For (P3,Q3,R2,1): Q is Q3, so the condition is true and this row survives.

So the left branch gives $\{(P3, Q3, R2, 1)\}$.

Step 3: Join Y and Z on their shared column P.

Y(P1,Q1,2) matches Z(P1,T1), giving (P1,Q1,2,T1). Y(P1,Q2,5) also matches Z(P1,T1), giving (P1,Q2,5,T1). Y(P2,Q1,6) has no Z row with P=P2, so it drops. Y(P3,Q3,1) matches Z(P3,T2), giving (P3,Q3,1,T2).

So $Y \bowtie Z = \{(P1, Q1, 2, T1), (P1, Q2, 5, T1), (P3, Q3, 1, T2)\}$.

Step 4: Apply the selection $S > 1$.

(P1,Q1,2,T1) has S=2, which is greater than 1, so it stays. (P1,Q2,5,T1) has S=5, so it stays. (P3,Q3,1,T2) has S=1, which is not greater than 1, so it is dropped.

So the right branch gives $\{(P1, Q1, 2, T1), (P1, Q2, 5, T1)\}$.

Step 5: Join the two branches.

The left branch has columns P, Q, R, S with the single row P3,Q3,R2,1. The right branch has columns P, Q, S, T. The natural join matches on the shared columns P, Q and S.

The left row has P=P3, but every row on the right has P=P1, so no pair of rows agrees on P. Since P already fails to match, the join has nothing to output.

Step 6: Project and conclude.

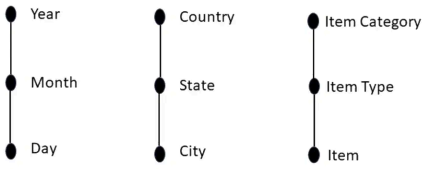
Projecting P, R, S on an empty relation still gives an empty relation.

So the final result has zero rows, which is option (D).

Zero rows

Quick Tip: Compute $X \bowtie Y$ and $Y \bowtie Z$ separately, apply each selection, then check whether the surviving P values on the two sides can ever match before joining them.

43. Consider the concept hierarchies as shown in the figure.



Which of the following options denotes the total number of possible data cuboids from these concept hierarchies?

(A) 4^3

(B) 2^3

(C) $2!$

(D) $4!$

Correct Answer: (A) 4^3

Solution:

Step 1: Recall what a data cuboid represents.

In OLAP data cube modelling, a cuboid is a group-by result at some level of aggregation for every dimension at once. Each dimension can be viewed at any level of its concept hierarchy, or fully generalized to a single "all" value that ignores that dimension.

Step 2: Count the aggregation levels available per dimension.

The figure gives three dimensions, each with a 3-level hierarchy: Year, Month, Day for the time dimension; Country, State, City for the location dimension; Item Category, Item Type, Item for the product dimension. Besides these 3 concrete levels, every dimension also has the implicit top level "all", which rolls that whole dimension up into a single group. So each dimension offers $3 + 1 = 4$ usable levels.

Step 3: Multiply the level counts across dimensions.

Since a cuboid is formed by picking one level independently from each of the three dimensions, the total number of cuboids is the product of the per-dimension level counts:

$$4 \times 4 \times 4 = 4^3$$

Step 4: Rule out the other options.

2^3 would only be correct if each dimension offered just 2 usable levels, but here each hierarchy has 3 concrete levels plus "all", giving 4, not 2. $2!$ and $4!$ are permutation counts and have no role in counting cuboids, since a cuboid is an independent choice of level per dimension, not an ordering.

Final Answer:
The total number of possible data cuboids is 4^3, option (A).

$$\boxed{4^3}$$

Quick Tip: Each dimension has 3 hierarchy levels plus an implicit "all" level, giving 4 choices per dimension; multiply these across the 3 independent dimensions.

44. Let X and Y be two independent random variables. X follows $Bernoulli(p = 0.3)$ distribution and Y follows $Normal(\mu = 0, \sigma^2 = 100)$ distribution.

Which of the following options is the variance of $(2X - 1)Y$?

(A) 100

(B) 90

(C) 49

(D) 21

Correct Answer: (A) 100

Solution:

Step 1: Simplify the factor $(2X - 1)$ first.
Let $Z = 2X - 1$. Since X is Bernoulli, it only takes the values 0 or 1, so Z only takes the values $2(0) - 1 = -1$ (when $X = 0$, probability 0.7) or $2(1) - 1 = 1$ (when $X = 1$, probability 0.3).
The key fact here is that Z only ever equals -1 or 1, so $Z^2 = 1$ with probability 1, no matter what X turns out to be.

Step 2: Find the mean of Z and confirm independence from Y.
$E[Z] = 2E[X] - 1 = 2(0.3) - 1 = -0.4$.
Since X and Y are independent, any function of X (such as Z) is also independent of Y.

Step 3: Find the mean of $W = ZY$.
For independent variables, $E[ZY] = E[Z]E[Y]$.
Since Y is Normal with mean 0, $E[Y] = 0$, so $E[W] = (-0.4)(0) = 0$.

Step 4: Find $E[W^2]$.

$$E[W^2] = E[Z^2Y^2] = E[Z^2]\,E[Y^2]$$

(using independence again, this time between Z^2 and Y^2).
We already found $Z^2 = 1$ always, so $E[Z^2] = 1$.
For Y, $E[Y^2] = \text{Var}(Y) + (E[Y])^2 = 100 + 0 = 100$.
So $E[W^2] = 1 \times 100 = 100$.

Step 5: Compute the variance.

$$\text{Var}(W) = E[W^2] - (E[W])^2 = 100 - 0^2 = 100$$

Final Answer:
The variance of $(2X - 1)Y$ is 100, option (A). The other options come from partial or incorrect use of the variance-of-a-product rule; none of them account correctly for the fact that $(2X - 1)^2$ is identically 1.

$$\boxed{100}$$

Quick Tip: Notice that $2X - 1$ only takes values -1 or 1, so its square is always 1; use this with independence to find $E[W]$ and $E[W^2]$ for $W = (2X - 1)Y$.

45. Let

$$L = \lim_{n\to\infty} \sum_{k=0}^{n} \frac{e^{-n}n^k}{k!}$$

Which of the following is the value of L?

(A) 0.5

(B) 1.0

(C) 0

(D) e^{-1}

Correct Answer: (A) 0.5

Solution:

Step 1: Recognize the sum as a Poisson probability.
The term $\frac{e^{-n}n^k}{k!}$ is exactly the probability mass function of a Poisson random variable N with mean n, evaluated at k.
So the full sum $\sum_{k=0}^{n} \frac{e^{-n}n^k}{k!}$ is $P(N \le n)$, the probability that a Poisson(n) random variable is at most its own mean, n.

Step 2: Use the fact that a Poisson mean and variance are equal.
For $N \sim Poisson(n)$, both the mean and the variance equal n, so the standard deviation is $\sqrt{n}$.

Step 3: Apply the Central Limit Theorem as $n \to \infty$.
A Poisson(n) variable can be built as the sum of n independent Poisson(1) variables, so as $n \to \infty$, the

standardized variable $\frac{N-n}{\sqrt{n}}$ converges in distribution to a standard Normal variable, by the Central Limit Theorem.

Step 4: Evaluate the limit at the mean.

$$P(N \le n) = P\left(\frac{N-n}{\sqrt{n}} \le 0\right) \to \Phi(0)$$

as $n \to \infty$, where Φ is the standard Normal CDF.
Since the standard Normal distribution is symmetric about 0, $\Phi(0) = 0.5$.

Step 5: Rule out the other options.
$L = 1.0$ would only hold if the sum ran all the way to infinity instead of stopping at $k = n$; stopping exactly at the mean captures only about half the probability mass. $L = 0$ is wrong because a nontrivial chunk (about half) of the Poisson(n) mass sits at or below its own mean for every n. e^{-1} is just the single $k = 0$ term of the sum, not the full sum up to $k = n$.

Final Answer:

$$\boxed{L = 0.5}$$

Quick Tip: The sum is the Poisson(n) CDF evaluated at its own mean n; as n grows, the Poisson distribution becomes approximately Normal and symmetric about its mean, so roughly half the probability lies at or below it.

46. Let $\gamma_1, \gamma_2, \gamma_3$ be the eigenvalues of the matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos t & \sin t \\ 0 & -\sin t & \cos t \end{bmatrix}$$

where $t \in [-\pi, \pi]$ is in radians.

Which one of the following options lists all the possible values of t satisfying $\gamma_1 + \gamma_2 + \gamma_3 = 1 + \sqrt{2}$?

(A) $\{\pi/3, -\pi/4\}$

(B) $\{\pi/4, -\pi/3\}$

(C) $\{\pi/4, -\pi/4\}$

(D) $\{\pi/3, -\pi/3\}$

Correct Answer: (C) $\{\pi/4, -\pi/4\}$

Solution:

Step 1: Recall how the sum of eigenvalues relates to the matrix.
For any square matrix, the sum of its eigenvalues always equals the trace of the matrix, that is, the sum of

the entries on the main diagonal. This holds regardless of whether the eigenvalues are real or complex.

Step 2: Compute the trace of the given matrix.
The diagonal entries are 1, $\cos t$ and $\cos t$, so

$$\gamma_1 + \gamma_2 + \gamma_3 = \text{trace} = 1 + \cos t + \cos t = 1 + 2\cos t$$

Step 3: Set the trace equal to the given value and solve for $\cos t$.

$$1 + 2\cos t = 1 + \sqrt{2}$$

$$2\cos t = \sqrt{2}$$

$$\cos t = \frac{\sqrt{2}}{2} = \frac{1}{\sqrt{2}}$$

Step 4: Find all t in $[-\pi, \pi]$ with this cosine value.
$\cos t = 1/\sqrt{2}$ corresponds to a reference angle of $\pi/4$ (45 degrees), and cosine is positive in both the first and fourth quadrants. Within $[-\pi, \pi]$, this gives

$$t = \frac{\pi}{4} \quad \text{or} \quad t = -\frac{\pi}{4}$$

Step 5: Rule out the other options.
$t = \pi/3$ gives $\cos t = 1/2$, which does not satisfy $\cos t = 1/\sqrt{2}$, so any option mixing in $\pm\pi/3$ is wrong. Only the pair $\{\pi/4, -\pi/4\}$ satisfies the equation.

Final Answer:
The correct option is (C).

$$\boxed{t \in \{\pi/4, -\pi/4\}}$$

Quick Tip: The sum of eigenvalues equals the matrix trace; set $1 + 2\cos t = 1 + \sqrt{2}$ and solve for $\cos t$, then find all matching angles in $[-\pi, \pi]$.

47. Consider that 20 stories of Author X and 10 stories of Author Y were kept together without mentioning the names of the authors. A classifier was then asked to predict the author (X or Y) of each of these stories. Let, out of X's stories, 6 were classified as that of Y. On the other hand, out of Y's stories, 2 were classified as that of X.

Considering X and Y as two classes, which of the following statements is/are true?

(A) Accuracy of the classifier is 11/15.

(B) Precision of Class X is higher than the Precision of Class Y.

(C) Recall of Class X is higher than the Recall of Class Y.

(D) Accuracy of the classifier is 14/15.

Correct Answer: (A) Accuracy of the classifier is 11/15. ; (B) Precision of Class X is higher than the Precision of Class Y.

Solution:

Step 1: Build the confusion matrix from the given numbers.

There are 20 stories by X and 10 by Y, 30 in total. Out of X's 20 stories, 6 are wrongly called Y, so $20 - 6 = 14$ are correctly called X. Out of Y's 10 stories, 2 are wrongly called X, so $10 - 2 = 8$ are correctly called Y.

	Predicted X	Predicted Y
Actual X (20 stories)	14	6
Actual Y (10 stories)	2	8

Step 2: Compute the overall accuracy.

Accuracy is the fraction of all stories correctly classified:

$$\text{Accuracy} = \frac{14 + 8}{30} = \frac{22}{30} = \frac{11}{15}$$

This matches statement (A), so (A) is true, and it also rules out statement (D), which claims 14/15.

Step 3: Compute Precision for class X and class Y.

Precision of a class is (correctly predicted as that class) divided by (all stories predicted as that class).

For X: predicted X totals $14 + 2 = 16$ (the 14 correct plus the 2 Y stories wrongly called X), so

$$\text{Precision}_X = \frac{14}{16} = 0.875$$

For Y: predicted Y totals $8 + 6 = 14$ (the 8 correct plus the 6 X stories wrongly called Y), so

$$\text{Precision}_Y = \frac{8}{14} \approx 0.571$$

Since $0.875 > 0.571$, Precision of X is indeed higher than Precision of Y, so statement (B) is true.

Step 4: Compute Recall for class X and class Y.

Recall of a class is (correctly predicted as that class) divided by (all stories that actually belong to that class).

$$\text{Recall}_X = \frac{14}{20} = 0.7, \qquad \text{Recall}_Y = \frac{8}{10} = 0.8$$

Here Recall of X (0.7) is lower than Recall of Y (0.8), not higher, so statement (C) is false.

Final Answer:
Statements (A) and (B) are true; (C) and (D) are false.

$$\boxed{\text{(A) and (B)}}$$

Quick Tip: Build the 2x2 confusion matrix from the story counts first (14, 6, 2, 8), then compute accuracy, precision per class, and recall per class from it.

48. Let $P(x)$ be a predicate.

Which of the following statements is/are NOT valid in first-order logic?

(A) $\forall x\, P(x) \Rightarrow \exists x\, P(x)$

(B) $\exists x\, P(x) \Rightarrow \forall x\, P(x)$

(C) $\exists x\, P(x) \Leftrightarrow \forall x\, P(x)$

(D) $\forall x\, P(x) \Rightarrow \exists x\, \neg P(x)$

Correct Answer: (B) $\exists x\, P(x) \Rightarrow \forall x\, P(x)$; (C) $\exists x\, P(x) \Leftrightarrow \forall x\, P(x)$; (D) $\forall x\, P(x) \Rightarrow \exists x\, \neg P(x)$

Solution:

Step 1: Recall what "valid" means in first-order logic.
A formula is valid if it comes out true under every possible interpretation, that is, for every non-empty domain and every way of assigning meaning to P. To show a formula is NOT valid, it is enough to find just one interpretation that makes it false.

Step 2: Check statement (A), $\forall x\, P(x) \Rightarrow \exists x\, P(x)$.
Assume the domain is non-empty, as standard in first-order logic. If $P(x)$ is true for every element of the domain, then in particular it is true for at least one element, so $\exists x\, P(x)$ must also be true. The implication can never be made false, so this statement is valid, and is not one of the NOT-valid answers.

Step 3: Check statement (B), $\exists x\, P(x) \Rightarrow \forall x\, P(x)$.
Take a domain $\{a, b\}$ with $P(a)$ true and $P(b)$ false. Then $\exists x\, P(x)$ is true (because of a), but $\forall x\, P(x)$ is false (because of b). A true antecedent with a false consequent makes the whole implication false, so this interpretation breaks the statement. It is NOT valid.

Step 4: Check statement (C), $\exists x\, P(x) \Leftrightarrow \forall x\, P(x)$.
Using the same interpretation as Step 3, $\exists x\, P(x)$ is true while $\forall x\, P(x)$ is false, so the two sides disagree and the biconditional is false. This statement is NOT valid either.

Step 5: Check statement (D), $\forall x\, P(x) \Rightarrow \exists x\, \neg P(x)$.

Take a domain where $P(x)$ is true for every element, for example a single-element domain $\{a\}$ with $P(a)$ true. Then $\forall x P(x)$ is true, but $\exists x \neg P(x)$ is false since there is no element where P fails. A true antecedent and false consequent again make the implication false, so this statement is NOT valid. In fact, this implication is false in every interpretation where $\forall x P(x)$ holds, so it can never be valid.

Final Answer:
Statements (B), (C) and (D) are NOT valid, while (A) is valid.

$$\boxed{\text{(B), (C), (D)}}$$

Quick Tip: Try a small two-element domain with mixed truth values for P; a formula is valid only if it stays true under every such assignment.

49. Consider the problem of sorting the given array in ascending order:

$P = [1, 2, 3, 5, 4]$

Consider two sorting algorithms Bubble Sort (BS) and Insertion Sort (IS).

Let N_1 be the total number of comparisons done by BS on the elements of P and N_2 be the total number of comparisons done by IS on the elements of P.

Which of the following options is/are correct?

(A) $N_1 = 10, N_2 = 4$

(B) $N_1 > N_2$

(C) IS on P will perform only one swap

(D) Both BS and IS on P will make at least one unnecessary comparison (i.e., comparing elements that are already in correct order)

Correct Answer: (B) $N_1 > N_2$; (C) IS on P will perform only one swap ; (D) Both BS and IS on P will make at least one unnecessary comparison (i.e., comparing elements that are already in correct order)

Solution:

Step 1: Count the comparisons made by Bubble Sort.
The array is $P = [1, 2, 3, 5, 4]$, with 5 elements. A standard bubble sort makes $(n-1)$ passes, and pass i (starting at 0) compares $(n-1-i)$ adjacent pairs.
Pass 1 compares 4 pairs: (1,2), (2,3), (3,5), (5,4). Only the last pair is out of order, so one swap happens and the array becomes [1, 2, 3, 4, 5].
Pass 2 compares 3 pairs: (1,2), (2,3), (3,4). No swaps needed.
Pass 3 compares 2 pairs: (1,2), (2,3). No swaps needed.
Pass 4 compares 1 pair: (1,2). No swap needed.
Total comparisons $N_1 = 4 + 3 + 2 + 1 = 10$.

Step 2: Count the comparisons made by Insertion Sort.
Insertion sort scans from left to right, inserting each new element into its correct place among the already-sorted elements to its left.
$i = 1$: key = 2, compare with 1. $1 > 2$ is false, so no shift. 1 comparison.
$i = 2$: key = 3, compare with 2. $2 > 3$ is false, so no shift. 1 comparison.
$i = 3$: key = 5, compare with 3. $3 > 5$ is false, so no shift. 1 comparison.
$i = 4$: key = 4, compare with 5. $5 > 4$ is true, so shift 5 right (1 comparison, 1 swap), then compare with 3. $3 > 4$ is false, stop (1 more comparison). This step used 2 comparisons.
Total comparisons $N_2 = 1 + 1 + 1 + 2 = 5$.

Step 3: Check option (A).
Option (A) claims $N_1 = 10, N_2 = 4$. We found $N_1 = 10$ but $N_2 = 5$, not 4, so option (A) is wrong.

Step 4: Check option (B).
$N_1 = 10$ and $N_2 = 5$, so $N_1 > N_2$ holds. Option (B) is correct.

Step 5: Check option (C).
Looking at the insertion sort trace in Step 2, the only place data actually moved was at $i = 4$, where 5 was shifted one place to the right to make room for 4. That is the single swap in the whole run, so option (C) is correct.

Step 6: Check option (D).
In bubble sort's first pass, the comparisons (1,2), (2,3), and (3,5) all compare elements that are already in the correct relative order, so they are unnecessary. Similarly, insertion sort's comparisons at $i = 1, 2, 3$ (1 vs 2, 2 vs 3, 3 vs 5) compare elements already in order. Both algorithms make at least one unnecessary comparison, so option (D) is correct.

Final Answer:
The correct options are (B), (C), and (D).

$$\boxed{N_1 = 10,\ N_2 = 5,\ \text{(B), (C), (D) are correct}}$$

Quick Tip: Trace both algorithms step by step on P = [1, 2, 3, 5, 4] and count every comparison made, including the ones between elements already in the right order.

50. Consider the given Python program.

```
def outer():
    x = []
    def inner(val):
        x.append(val)
        return x
    return inner

f1 = outer()
f2 = outer()
print(f1(10))   # Line P
print(f1(20))   # Line Q
print(f2(30))   # Line R
print(f1(40))   # Line S
```

Which of the following options is/are correct?

(A) f1 and f2 share the same list x

(B) Output of Line Q is [10, 20]

(C) Output of Line R is [10, 20, 30]

(D) Output of Line S is [10, 20, 40]

Correct Answer: (B) Output of Line Q is [10, 20] ; (D) Output of Line S is [10, 20, 40]

Solution:

Step 1: Understand what outer() creates.

Each time outer() is called, Python creates a brand new local variable x = [] and a brand new inner function that closes over that particular x. The inner function does not share x with any other call to outer(); each call gets its own private list captured in the closure.

Step 2: Trace f1 = outer() and f2 = outer().

f1 = outer() creates one x (call it x1 = []) and binds f1 to the inner function that appends to x1.

f2 = outer() creates a completely separate x (call it x2 = []) and binds f2 to the inner function that appends to x2.

So f1 and f2 never touch the same list, they each keep growing their own list across repeated calls.

Step 3: Trace each print statement.

Line P: f1(10) appends 10 to x1, so x1 = [10]. Output: [10].

Line Q: f1(20) appends 20 to the same x1 (because f1 remembers x1 between calls), so x1 = [10, 20]. Output: [10, 20].

Line R: f2(30) appends 30 to x2, which is still empty at this point since f2 was never called before, so x2 = [30]. Output: [30].

Line S: f1(40) appends 40 to x1 again, so x1 = [10, 20, 40]. Output: [10, 20, 40].

Step 4: Check each option.
(A) claims f1 and f2 share the same list x. From Step 2 this is false, they have separate lists.
(B) claims Line Q prints [10, 20]. This matches the trace exactly, so it is true.
(C) claims Line R prints [10, 20, 30]. The trace shows Line R prints [30] only, so this is false.
(D) claims Line S prints [10, 20, 40]. This matches the trace, so it is true.

Final Answer:
Options (B) and (D) are correct.

$$\boxed{\text{(B), (D)}}$$

Quick Tip: Remember that each call to outer() creates a brand new local list; f1 and f2 do not share state, but repeated calls to the same function do.

51. Consider a table Employee(EmpID, TeamID), where the column EmpID (ID of an employee) is the primary key. The column TeamID denotes the team ID of the team of which the employee is a member. TeamID is a NOT NULL column.

We want to display the size of the team (denoted as TeamSize) in which each employee is a member by using SQL. As an example, the desired output for the given Employee table is also shown in tabular form.

Which of the following is/are correct?

Employee table:

EmpID	TeamID
1	8
2	8
3	8
4	7
5	7
6	9

Output table:

EmpID	TeamSize
1	3
2	3
3	3
4	2
5	2
6	1

(A)

```
SELECT E.EmpID, B.TeamSize
FROM Employee AS E, (SELECT TeamID, COUNT(TeamID) AS TeamSize FROM Employee GROUP BY TeamID) AS B
WHERE E.TeamID = B.TeamID
```

(B)

```
SELECT A.EmpID, COUNT(B.TeamID) AS TeamSize
FROM Employee AS A, Employee AS B
WHERE A.TeamID = B.TeamID AND A.EmpID = B.EmpID
GROUP BY A.EmpID
```

(C)

```
SELECT B.EmpID, B.TeamSize
FROM (SELECT EmpID, COUNT(TeamID) AS TeamSize
FROM Employee GROUP BY EmpID) AS B
```

(D)

```
SELECT A.EmpID, B.TeamSize
FROM Employee AS A, (SELECT COUNT(TeamID) AS TeamSize FROM Employee GROUP BY TeamID) AS B
WHERE A.TeamID = B.TeamID
```

Correct Answer: (A)

```
SELECT E.EmpID, B.TeamSize
FROM Employee AS E, (SELECT TeamID, COUNT(TeamID) AS TeamSize FROM Employee GROUP BY TeamID) AS B
WHERE E.TeamID = B.TeamID
```

Solution:

Step 1: Understand what the correct query needs to do.
Each employee's TeamSize should equal the number of employees who share that employee's TeamID. So we first need to count how many employees fall in each TeamID group, then attach that count back to every employee row with the matching TeamID.

Step 2: Check option (A).
The subquery B groups the Employee table by TeamID and computes COUNT(TeamID) for each group, giving TeamID 8 a count of 3, TeamID 7 a count of 2, and TeamID 9 a count of 1. The outer query joins every employee E to B using E.TeamID = B.TeamID, so each employee picks up the TeamSize of their own team. This reproduces the desired output exactly, so option (A) is correct.

Step 3: Check option (B).
This query self joins Employee as A and B, but the WHERE clause requires both A.TeamID = B.TeamID AND A.EmpID = B.EmpID. Since EmpID is the primary key, A.EmpID = B.EmpID forces A and B to be the exact same row every time, so COUNT(B.TeamID) always evaluates to 1 for every employee, regardless of

their real team size. This does not match the desired output, so option (B) is incorrect.

Step 4: Check option (C).
Here the inner query groups by EmpID instead of TeamID. Since EmpID is already unique for every row (it is the primary key), grouping by it does nothing, each group has exactly one row, so COUNT(TeamID) is always 1. The result gives TeamSize = 1 for every employee, which does not match the desired output, so option (C) is incorrect.

Step 5: Check option (D).
The subquery B only selects COUNT(TeamID) AS TeamSize, grouped by TeamID, it never includes TeamID itself in the SELECT list. The outer query then tries to reference B.TeamID in the WHERE clause, but that column does not exist in B's result set. This makes the query invalid, it would fail with an unknown column error, so option (D) is incorrect.

Final Answer:
Only option (A) produces the correct TeamSize output; it is the only query that both runs correctly and matches the desired table.

(A) only

Quick Tip: Work out what each subquery actually returns on the sample data before joining it back to Employee; watch for grouping by the wrong column or a self-join that inadvertently matches only identical rows.

52. Let $M = \left(I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)$ be a matrix, where $\mathbf{1} = (1, 1, 1, \ldots, 1)^T \in \mathbb{R}^n$ and I_n is the identity matrix of order n.

Which of the following options is/are correct?

(A) $M^T = M$

(B) $M^2 = I_n$

(C) $\text{Trace}(M) = n$

(D) M is a projection matrix

Correct Answer: (A) $M^T = M$; (D) M is a projection matrix

Solution:

Step 1: Understand the matrix M.
$M = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ is the well known centering matrix. $\mathbf{1}\mathbf{1}^T$ is an $n \times n$ matrix of all ones, so M subtracts the average of the entries of a vector from each of its components when applied. Before checking the options, note the key scalar fact $\mathbf{1}^T\mathbf{1} = n$, since it is a sum of n ones.

Step 2: Check option (A), symmetry.
I_n is symmetric, and $\mathbf{1}\mathbf{1}^T$ is symmetric because $(\mathbf{1}\mathbf{1}^T)^T = \mathbf{1}\mathbf{1}^T$ (the transpose of an outer product of a vector with itself is itself). A difference of two symmetric matrices is symmetric, so $M^T = M$. Option (A) is correct.

Step 3: Check option (B), whether M squared is the identity.
Expand M^2:

$$M^2 = \left(I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)^2 = I_n - \frac{2}{n}\mathbf{1}\mathbf{1}^T + \frac{1}{n^2}\mathbf{1}\mathbf{1}^T\mathbf{1}\mathbf{1}^T$$

Use $\mathbf{1}^T\mathbf{1} = n$, so $\mathbf{1}\mathbf{1}^T\mathbf{1}\mathbf{1}^T = \mathbf{1}(\mathbf{1}^T\mathbf{1})\mathbf{1}^T = n\mathbf{1}\mathbf{1}^T$.

$$M^2 = I_n - \frac{2}{n}\mathbf{1}\mathbf{1}^T + \frac{1}{n^2} \cdot n\mathbf{1}\mathbf{1}^T = I_n - \frac{2}{n}\mathbf{1}\mathbf{1}^T + \frac{1}{n}\mathbf{1}\mathbf{1}^T = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T = M$$

So $M^2 = M$, not I_n (for $n > 1, M \neq I_n$). Option (B) is incorrect.

Step 4: Check option (C), the trace.

$$\text{Trace}(M) = \text{Trace}(I_n) - \frac{1}{n}\text{Trace}(\mathbf{1}\mathbf{1}^T)$$

$\text{Trace}(I_n) = n$, and $\text{Trace}(\mathbf{1}\mathbf{1}^T) = \mathbf{1}^T\mathbf{1} = n$ (the trace of an outer product equals the inner product).

$$\text{Trace}(M) = n - \frac{1}{n} \cdot n = n - 1$$

This is $n - 1$, not n, so option (C) is incorrect.

Step 5: Check option (D), whether M is a projection matrix.
A matrix is called a projection matrix when it is symmetric and idempotent, meaning $M^T = M$ and $M^2 = M$. Step 2 showed $M^T = M$ and Step 3 showed $M^2 = M$. Both conditions hold, so M is indeed a (orthogonal) projection matrix, it projects any vector onto the subspace orthogonal to $\mathbf{1}$. Option (D) is correct.

Final Answer:
Options (A) and (D) are correct.

$$\boxed{\text{(A), (D)}}$$

Quick Tip: Use the identity $\mathbf{1}^T\mathbf{1} = n$ to expand M^2 and $\text{Trace}(M)$; a symmetric idempotent matrix is by definition a projection matrix.

53. Let $X_1, X_2, \ldots, X_n$ be n independent random variables. Each of the random variables follows $Normal(\mu = 0, \sigma^2 = 1)$ distribution. Define $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$.

Which of the following statements is/are correct?

(A) $\sum_{i=1}^{n} X_i^2$ follows Chi-square distribution with n degrees of freedom.

(B) $\sum_{i=1}^{n} (X_i - \bar{X})^2$ follows Chi-square distribution with $(n-1)$ degrees of freedom.

(C) $X_1^2 + X_n^2$ follows exponential distribution with mean 2.

(D) $(\sqrt{n}\bar{X})^2$ follows Chi-square distribution with 2 degrees of freedom.

Correct Answer: (A) $\sum_{i=1}^{n} X_i^2$ follows Chi-square distribution with n degrees of freedom. ; (B) $\sum_{i=1}^{n} (X_i - \bar{X})^2$ follows Chi-square distribution with $(n-1)$ degrees of freedom. ; (C) $X_1^2 + X_n^2$ follows exponential distribution with mean 2.

Solution:

Step 1: Recall the definition of the Chi-square distribution.

If $Z_1, Z_2, \ldots, Z_k$ are independent standard normal random variables, then by definition $\sum_{i=1}^{k} Z_i^2$ follows a Chi-square distribution with k degrees of freedom. Here each $X_i \sim N(0,1)$ is already standard normal, since $\mu = 0$ and $\sigma^2 = 1$.

Step 2: Check option (A).

$X_1, \ldots, X_n$ are n independent standard normal variables, so by the definition in Step 1, $\sum_{i=1}^{n} X_i^2$ follows Chi-square with n degrees of freedom exactly. Option (A) is correct.

Step 3: Check option (B).

This is a classical result used to build the sample variance: when $X_1, \ldots, X_n$ are iid $N(\mu, \sigma^2)$, the sum of squared deviations from the sample mean, $\sum_{i=1}^{n} (X_i - \bar{X})^2/\sigma^2$, follows a Chi-square distribution with $(n-1)$ degrees of freedom, one degree of freedom is used up in estimating $\bar{X}$ from the data. Here $\sigma^2 = 1$, so $\sum_{i=1}^{n} (X_i - \bar{X})^2$ itself follows Chi-square with $(n-1)$ degrees of freedom. Option (B) is correct.

Step 4: Check option (C).

Each X_i^2 individually follows Chi-square with 1 degree of freedom (a single squared standard normal). The sum of two independent Chi-square variables with 1 degree of freedom each gives a Chi-square with $1 + 1 = 2$ degrees of freedom, so $X_1^2 + X_n^2 \sim \chi_2^2$ (note X_1 and X_n are independent since all X_i are independent). A Chi-square distribution with 2 degrees of freedom is a special case of the Gamma distribution with shape 1 and scale 2, which is exactly the Exponential distribution with scale 2, whose mean is 2. So $X_1^2 + X_n^2$ does follow an Exponential distribution with mean 2. Option (C) is correct.

Step 5: Check option (D).

$\sqrt{n}\bar{X} = \sqrt{n} \cdot \frac{1}{n} \sum_{i=1}^{n} X_i = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} X_i$. Since the X_i are iid $N(0,1)$, $\sum X_i \sim N(0, n)$, so dividing by $\sqrt{n}$ gives $\sqrt{n}\bar{X} \sim N(0,1)$, a single standard normal variable. Squaring a single standard normal gives a Chi-square with 1 degree of freedom, not 2. Option (D) is incorrect.

Final Answer:

Options (A), (B), and (C) are correct.

$$\boxed{\text{(A), (B), (C)}}$$

Quick Tip: Remember that a sum of k independent squared standard normals is Chi-square with k degrees of freedom, and Chi-square with 2 degrees of freedom is the same as Exponential with mean 2.

54. Let X be a discrete valued random variable with cumulative distribution function $F(x)$.

Which of the following statements is/are correct?

(A) $F(x)$ is always a positive function.

(B) $F(x)$ is a non-decreasing function.

(C) $F(x)$ has jump discontinuity.

(D) $F(x)$ is a left continuous function.

Correct Answer: (B) $F(x)$ is a non-decreasing function. ; (C) $F(x)$ has jump discontinuity.

Solution:

Step 1: Recall the definition of a CDF.
For any random variable X, the cumulative distribution function is $F(x) = P(X \leq x)$. Since X is discrete, $F(x)$ is built by adding up the probabilities of every value X can take that is less than or equal to x.

Step 2: Check option (A).
$F(x)$ is a probability, so it always satisfies $0 \leq F(x) \leq 1$, it can equal 0 (for any x smaller than the minimum value X can take) and it can equal 1 (for any x at or above the maximum value X can take). Since $F(x) = 0$ is possible, $F(x)$ is not "always positive" in the strict sense, it is only ever non-negative. Option (A) is incorrect.

Step 3: Check option (B).
For $x_1 < x_2$, the event $\{X \leq x_1\}$ is a subset of $\{X \leq x_2\}$, so $P(X \leq x_1) \leq P(X \leq x_2)$, meaning $F(x_1) \leq F(x_2)$. This holds for every CDF, discrete or continuous, so $F(x)$ is always non-decreasing. Option (B) is correct.

Step 4: Check option (C).
Because X is discrete, it only takes a countable set of specific values, each with positive probability. As x crosses one of these values, $F(x)$ instantly jumps up by the probability mass at that point, rather than rising smoothly. This gives $F(x)$ a staircase shape with jump discontinuities exactly at the values X can take. Option (C) is correct.

Step 5: Check option (D).
By convention and by construction, a CDF is right continuous: $\lim_{h \to 0^+} F(x + h) = F(x)$, because $P(X \leq x)$ already includes the point x itself, so approaching from the right does not add anything new. Approaching from the left, however, $\lim_{h \to 0^+} F(x - h)$ does not equal $F(x)$ at a jump point, it equals $F(x)$ minus the probability mass at x. So $F(x)$ is right continuous, not left continuous. Option (D) is incorrect.

Final Answer:
Options (B) and (C) are correct.

(B), (C)

Quick Tip: Check the CDF definition $F(x) = P(X \leq x)$ directly against each claim: it is non-negative and non-decreasing always, and being right continuous with jumps is specific to discrete random variables.

55. Consider that Linear Ridge Regression is being used to learn a prediction function $y_{pred} = w^T x$, where $w, x \in \mathbb{R}^2$ and Mean Absolute Error (MAE) is used to measure the prediction error. A weight of 0.20 is associated with the regularizer.

At an intermediate step of the training process, assume that the parameter $w = [-3.00,\ 4.00]^T$. In the next step, for the input $x = [1.00,\ 2.00]^T$, the predicted value of y is noted. Let the relation between $x = [x_1,\ x_2]^T$ and the true value of y be $y_{true} = x_1 + x_2$.

The value of the overall regularized loss function for this instance is _____ . (Rounded off to two decimal places)

Correct Answer: 7.00

Solution:

Step 1: Understand the loss function for Ridge Regression with MAE.
Ridge regression adds an L2 penalty on the weights to whatever base error metric is used. Here the base error is Mean Absolute Error, so for a single data point the overall regularized loss is:

$$L = |y_{pred} - y_{true}| + \lambda \|w\|_2^2$$

where $\lambda = 0.20$ is the given regularization weight and $\|w\|_2^2 = w_1^2 + w_2^2$ is the squared L2 norm of the weight vector.

Step 2: Compute the predicted value.
$y_{pred} = w^T x = (-3.00)(1.00) + (4.00)(2.00) = -3.00 + 8.00 = 5.00$

Step 3: Compute the true value.
$y_{true} = x_1 + x_2 = 1.00 + 2.00 = 3.00$

Step 4: Compute the Mean Absolute Error term.

$$|y_{pred} - y_{true}| = |5.00 - 3.00| = 2.00$$

Step 5: Compute the regularization term.

$$\|w\|_2^2 = (-3.00)^2 + (4.00)^2 = 9.00 + 16.00 = 25.00$$

$$\lambda \|w\|_2^2 = 0.20 \times 25.00 = 5.00$$

Step 6: Add both terms to get the overall regularized loss.

$$L = 2.00 + 5.00 = 7.00$$

Final Answer:
The overall regularized loss for this instance is 7.00.

$$\boxed{7.00}$$

Quick Tip: Compute the MAE term |w^T x - y_true| and the L2 penalty lambda times ||w||^2 separately, then add them together.

56. Consider a fully-connected feed-forward multi-layer perceptron. It has 30 neurons in the input layer, followed by two hidden layers and an output layer. The first hidden layer has 4 neurons and the second hidden layer has 3 neurons. The output layer has only one neuron. Assume that no bias parameters are used.

The number of learnable parameters in the multi-layer perceptron is _____ . (Answer in integer)

Correct Answer: 135

Solution:

Step 1: Identify the layer sizes.
Input layer: 30 neurons.
Hidden layer 1: 4 neurons.
Hidden layer 2: 3 neurons.
Output layer: 1 neuron.
Since the network is fully connected, every neuron in one layer connects to every neuron in the next layer with a separate weight, and no bias terms are used, so the only learnable parameters are these connection weights.

Step 2: Count the weights between the input layer and hidden layer 1.
Each of the 30 input neurons connects to each of the 4 neurons in hidden layer 1:

$$30 \times 4 = 120$$

Step 3: Count the weights between hidden layer 1 and hidden layer 2.
Each of the 4 neurons in hidden layer 1 connects to each of the 3 neurons in hidden layer 2:

$$4 \times 3 = 12$$

Step 4: Count the weights between hidden layer 2 and the output layer.
Each of the 3 neurons in hidden layer 2 connects to the single output neuron:

$$3 \times 1 = 3$$

Step 5: Add up all the weights.

$$120 + 12 + 3 = 135$$

No bias parameters are added anywhere, since the problem states none are used.

Final Answer:
The multi-layer perceptron has 135 learnable parameters.

$$\boxed{135}$$

Quick Tip: With no bias terms, count only the weight connections between each pair of consecutive layers (input to output) and add them up: 30x4 + 4x3 + 3x1.

57. A clinic specializes in testing for a disease D. The result of the test can be either positive or negative.

A study revealed that if a person suffers from the disease D, the test result in that clinic comes out positive 80% of the time, and negative 20% of the time. If a person is not suffering from the disease D, the test comes out positive 10% of the time and negative 90% of the time. It is also known that among the general population, the disease D occurs in 30% of the individuals.

If a person tests positive for D in that clinic, the probability that he/she actually suffers from the disease D is _____. (Rounded off to two decimal places)

Correct Answer: 0.77

Solution:

Step 1: Write down the given probabilities.
Let D be the event that a person has the disease and D^c be the event that a person does not have the disease.
Let P be the event that the test comes out positive.

From the question: $P(D) = 0.30$, so $P(D^c) = 0.70$.
The test is positive 80% of the time when the person has the disease: $P(P \mid D) = 0.80$.
The test is positive 10% of the time when the person does not have the disease: $P(P \mid D^c) = 0.10$.

Step 2: Set up Bayes' theorem.
We want $P(D \mid P)$, the chance the person actually has the disease given a positive result. Bayes' theorem gives

$$P(D \mid P) = \frac{P(P \mid D)\,P(D)}{P(P \mid D)\,P(D) + P(P \mid D^c)\,P(D^c)}$$

The denominator is the total probability of testing positive, made up of positives coming from sick people plus positives coming from healthy people.

Step 3: Plug in the numbers.
Numerator: $P(P \mid D)\,P(D) = 0.80 \times 0.30 = 0.24$
Denominator, first term: $0.80 \times 0.30 = 0.24$
Denominator, second term: $P(P \mid D^c)\,P(D^c) = 0.10 \times 0.70 = 0.07$
Total probability of a positive test: $0.24 + 0.07 = 0.31$

Step 4: Compute the final probability.

$$P(D \mid P) = \frac{0.24}{0.31} = 0.7742\ldots$$

Rounded to two decimal places, this is 0.77.

Final Answer:
The probability that a person who tests positive actually has the disease is about 0.77.

$$\boxed{0.77}$$

Quick Tip: Use Bayes' theorem: $P(D|\text{positive}) = \frac{P(\text{positive}|D)P(D)}{P(\text{positive}|D)P(D) + P(\text{positive}|D^c)P(D^c)}$.

58. Consider the given Python program.

```
def fun(L, i=0):
    if i >= len(L)-1:
        return 0
    if L[i] > L[i+1]:
        L[i+1], L[i] = L[i], L[i+1]
        return 1+fun(L, i+1)
    else:
        return fun(L, i+1)

data = [5, 3, 4, 1, 2]
count = 0
for _ in range(len(data)):
    count += fun(data)
print(count)
```

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

Correct Answer: 8

Solution:

Step 1: Understand what fun does in one call.

The function fun(L, i) walks through the list L starting at index i. At each position it compares L[i] with L[i+1]. If L[i] is bigger, it swaps the two elements and adds 1 to the count, then moves on to i+1. If not, it just moves on to i+1 without adding anything. It stops when i reaches the last valid index.

This is exactly one left-to-right pass of bubble sort: it walks the whole list once, and counts how many swaps happened during that single pass.

Step 2: Note the outer loop.

data = [5, 3, 4, 1, 2] has 5 elements, so range(len(data)) runs 5 times. Each iteration calls fun(data) once (i defaults back to 0 each time) and adds its return value to count. So the program runs 5 full bubble-sort passes over data and adds up the total number of swaps across all 5 passes.

Step 3: Trace Pass 1.

Start: [5, 3, 4, 1, 2]

i=0: 5 > 3, swap -> [3, 5, 4, 1, 2], swaps so far = 1

i=1: 5 > 4, swap -> [3, 4, 5, 1, 2], swaps so far = 2

i=2: 5 > 1, swap -> [3, 4, 1, 5, 2], swaps so far = 3

i=3: 5 > 2, swap -> [3, 4, 1, 2, 5], swaps so far = 4

Pass 1 total swaps = 4. count = 4. List is now [3, 4, 1, 2, 5].

Step 4: Trace Pass 2.

Start: [3, 4, 1, 2, 5]
i=0: 3 > 4? No, no swap.
i=1: 4 > 1, swap -> [3, 1, 4, 2, 5], swaps so far = 1
i=2: 4 > 2, swap -> [3, 1, 2, 4, 5], swaps so far = 2
i=3: 4 > 5? No, no swap.
Pass 2 total swaps = 2. count = 4 + 2 = 6. List is now [3, 1, 2, 4, 5].

Step 5: Trace Pass 3.
Start: [3, 1, 2, 4, 5]
i=0: 3 > 1, swap -> [1, 3, 2, 4, 5], swaps so far = 1
i=1: 3 > 2, swap -> [1, 2, 3, 4, 5], swaps so far = 2
i=2: 3 > 4? No. i=3: 4 > 5? No.
Pass 3 total swaps = 2. count = 6 + 2 = 8. List is now [1, 2, 3, 4, 5], fully sorted.

Step 6: Passes 4 and 5.
The list [1, 2, 3, 4, 5] is already sorted, so every comparison in pass 4 and pass 5 finds L[i] <= L[i+1] and no swap happens. Both passes add 0 to count. count stays at 8 after all 5 iterations.

Final Answer:
print(count) outputs 8.

8

Quick Tip: Each call to fun performs one bubble-sort pass and counts its swaps; running it 5 times over a 5-element list is enough to fully sort it, so the total equals the number of inversions in the original list.

59. Let there be two relations X and Y as shown. X has three columns P, Q and R. Y has two columns P and S.

Relation X:

P	Q	R
P1	Q1	R1
P2	Q2	R2
P3	Q3	R2

Relation Y:

P	S
P1	10
P1	15
P2	20
P3	1

Consider that the following tuple relational calculus expression is evaluated.

$$\{t \mid t \in X \wedge \exists z \in X(t[P] = z[P]) \wedge \exists m \in Y(m[P] = t[P] \wedge m[S] > 1)\}$$

The number of tuples that will be returned is _____. (Answer in integer)

Correct Answer: 2

Solution:

Step 1: Understand each part of the expression.
The expression picks out tuples t from relation X that satisfy two conditions.
The middle condition $\exists z \in X(t[P] = z[P])$ asks: does some tuple z in X exist with the same P value as t? Since t itself is already a tuple of X, we can always choose $z = t$, and then $t[P] = z[P]$ is trivially true. So this condition holds for every tuple in X and does not filter out anything.
The real filter is the last condition: $\exists m \in Y(m[P] = t[P] \wedge m[S] > 1)$. This asks whether some row m in Y has the same P value as t and has $S > 1$.

Step 2: Check the condition for each tuple of X.
X has three tuples: $(P1, Q1, R1)$, $(P2, Q2, R2)$, $(P3, Q3, R2)$. Check the Y rows for each P value.
For $P1$: Y has rows $(P1, 10)$ and $(P1, 15)$. Both have $S > 1$. So a matching m exists, and the condition is true.
For $P2$: Y has the row $(P2, 20)$. Since $20 > 1$, the condition is true.
For $P3$: Y has only the row $(P3, 1)$. Here $S = 1$, and $1 > 1$ is false. There is no other row in Y with $P = P3$, so no matching m exists. The condition is false for $P3$.

Step 3: Collect the tuples that pass.
The tuple with $P1$, namely $(P1, Q1, R1)$, passes.

The tuple with $P2$, namely $(P2, Q2, R2)$, passes.
The tuple with $P3$, namely $(P3, Q3, R2)$, fails and is excluded.

Final Answer:
Exactly 2 tuples of X satisfy the expression.

2

Quick Tip: The self-reference on X is always true; the real filter checks whether each tuple's P value appears in Y with an S value strictly greater than 1.

60. Let Account be a relation as shown.

Table: Account

AccNo	Balance
A1	5000
A2	5000
A3	10000
A4	15000
A5	18000

Consider the given SQL query.

```
SELECT AccNo FROM Account AS A
WHERE (SELECT COUNT(*) FROM Account AS B
WHERE A.Balance < B.Balance) >= (SELECT COUNT(*)
FROM Account AS C WHERE A.Balance > C.Balance)
```

The number of rows returned by the SQL query is _____. (Answer in integer)

Correct Answer: 3

Solution:

Step 1: Understand what the two subqueries compute for each row.
The query goes through every row A of Account, one at a time, and checks a condition built from two correlated subqueries.
The first subquery, (SELECT COUNT(*) FROM Account AS B WHERE A.Balance < B.Balance), counts how many accounts in the whole table have a balance strictly greater than A's balance. Call this "countGreater".
The second subquery, (SELECT COUNT(*) FROM Account AS C WHERE A.Balance > C.Balance), counts how many accounts have a balance strictly less than A's balance. Call this "countLess".

A row A is kept in the output only when countGreater >= countLess, meaning at least as many accounts beat A as A beats.

Step 2: List the balances and compute countGreater and countLess for each account.
The balances are A1=5000, A2=5000, A3=10000, A4=15000, A5=18000.
For A1 (5000): accounts with a strictly greater balance are A3, A4, A5, so countGreater = 3. Accounts with a strictly smaller balance: none, since the only other 5000 (A2) is equal, not smaller, so countLess = 0.
For A2 (5000): by the same reasoning, countGreater = 3 and countLess = 0.
For A3 (10000): accounts with a strictly greater balance are A4 and A5, so countGreater = 2. Accounts with a strictly smaller balance are A1 and A2, so countLess = 2.
For A4 (15000): only A5 (18000) is greater, so countGreater = 1. A1, A2, A3 are all smaller, so countLess = 3.
For A5 (18000): no account is greater, so countGreater = 0. A1, A2, A3, A4 are all smaller, so countLess = 4.

Step 3: Apply the condition countGreater >= countLess to each account.
A1: 3 >= 0, true. Kept.
A2: 3 >= 0, true. Kept.
A3: 2 >= 2, true. Kept.
A4: 1 >= 3, false. Dropped.
A5: 0 >= 4, false. Dropped.

Final Answer:
The query returns the AccNo of A1, A2 and A3, which is 3 rows in total.

$\boxed{3}$

Quick Tip: For each account, compare how many accounts have a strictly higher balance against how many have a strictly lower balance; only rows at or below the median pass.

61. Consider an ER model with the entities E1(A11, A12, A13) and E2(A21, A22, A23), where A11, A12, A13 are the attributes of E1, and A21, A22, A23 are the attributes of E2. Let A22 be a multi-valued attribute. A11 and A21 are the primary keys of E1 and E2, respectively.

Let R12 be a many-to-many relationship between E1 and E2. Participation of both E1 and E2 in R12 is total.

The minimum number of relations required to convert the ER model into relational model (assuming there is no other functional dependency) where each relation is in third normal form (3NF) is _____. (Answer in integer)

Correct Answer: 4

Solution:

Step 1: Recall the standard ER-to-relational mapping rules.
Three rules matter here: every strong entity becomes its own relation, every multi-valued attribute needs its

own separate relation because a single column cannot hold a set of values, and every many-to-many relationship needs its own separate relation, regardless of whether participation is total or partial, because it cannot be merged into either entity's table.

Step 2: Build the relation for E1.
E1 has attributes A11, A12, A13, with A11 as the primary key. None of these are multi-valued, so E1 converts directly into one relation: E1(A11, A12, A13). This is relation 1.

Step 3: Build the relation for E2, pulling out the multi-valued attribute.
E2 has attributes A21, A22, A23, with A21 as the primary key, and A22 is multi-valued. A multi-valued attribute cannot sit inside the main entity relation, since storing a set of values in one column would break first normal form. So it is removed from the main E2 relation and placed in a relation of its own that references the entity's key.
Main E2 relation: E2(A21, A23). This is relation 2.
Multi-valued attribute relation: A22_table(A21, A22), where A21 is a foreign key referencing E2, and the primary key of this table is the combination (A21, A22). This is relation 3.

Step 4: Build the relation for the many-to-many relationship R12.
Even though participation of both entities in R12 is total, a many-to-many relationship can never be absorbed into either participating entity's table, because doing so would force repeating groups of foreign keys inside that table. It always needs its own bridge relation carrying the primary keys of both entities: R12(A11, A21), with the pair (A11, A21) forming the primary key of this bridge table. This is relation 4.

Step 5: Check that no relation can be merged further.
E1 and E2 cannot merge (they are separate strong entities). The multi-valued attribute table cannot merge into E2 without violating 1NF. R12 cannot merge into E1 or E2 even though participation is total, because it is a many-to-many relationship, not one-to-many. So all four relations are required at minimum, and each is already in 3NF by the question's assumption.

Final Answer:
The minimum number of relations needed is 4.

4

Quick Tip: Count separately: one relation per strong entity, one relation for the multi-valued attribute, and one relation for the many-to-many relationship, since M:N relationships always need their own table regardless of total participation.

62. For a given data set $\{x_1, x_2, \ldots, x_n\}$, where $n = 100$, it is known that

$$\frac{1}{2000}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - x_j)^2 = 99$$

Let us denote $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

The value of $\frac{1}{99}\sum_{i=1}^{n}(x_i - \bar{x})^2$ is _____. (Answer in integer)

Correct Answer: 10

Solution:

Step 1: Recall the identity linking the pairwise sum to the variance-type sum.

There is a standard identity connecting the double sum of squared pairwise differences to the sum of squared deviations from the mean:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - x_j)^2 = 2n\sum_{i=1}^{n}(x_i - \bar{x})^2$$

We can see why this holds by expanding the left side. Write $S_2 = \sum x_i^2$ and $S_1 = \sum x_i$.

$$\sum_i\sum_j(x_i - x_j)^2 = \sum_i\sum_j(x_i^2 - 2x_ix_j + x_j^2) = nS_2 - 2S_1^2 + nS_2 = 2nS_2 - 2S_1^2$$

On the other hand, the sum of squared deviations from the mean expands as $\sum_i(x_i - \bar{x})^2 = S_2 - \frac{S_1^2}{n}$, so $2n\sum_i(x_i - \bar{x})^2 = 2nS_2 - 2S_1^2$, which matches the left side exactly. This confirms the identity.

Step 2: Find the value of the double sum from the given information.

We are given

$$\frac{1}{2000}\sum_i\sum_j(x_i - x_j)^2 = 99$$

Multiply both sides by 2000:

$$\sum_i\sum_j(x_i - x_j)^2 = 99 \times 2000 = 198000$$

Step 3: Use the identity to find the sum of squared deviations from the mean.

From Step 1, with $n = 100$:

$$198000 = 2n\sum_i(x_i - \bar{x})^2 = 2(100)\sum_i(x_i - \bar{x})^2 = 200\sum_i(x_i - \bar{x})^2$$

Divide both sides by 200:

$$\sum_{i=1}^{n}(x_i-\bar{x})^2=\frac{198000}{200}=990$$

Step 4: Compute the requested quantity.

$$\frac{1}{99}\sum_{i=1}^{n}(x_i-\bar{x})^2=\frac{990}{99}=10$$

Final Answer:
The value asked for is 10.

$$\boxed{10}$$

Quick Tip: Use the identity $\sum_i\sum_j(x_i-x_j)^2=2n\sum_i(x_i-\bar{x})^2$ to convert the given double sum into the sum of squared deviations from the mean.

63. Let X be a random variable that follows $Uniform(-1,1)$ distribution. The conditional distribution of the random variable Y given $X=x$ is the $Uniform(x^2-0.1,x^2+0.1)$ distribution.

The value of $correlation(X,Y)$ is _____. (Answer in integer)

Correct Answer: 0

Solution:

Step 1: Write down what we know about X and the conditional distribution of Y.
$X\sim Uniform(-1,1)$, so its density is symmetric about 0. This means $E[X]=0$, and also that the distribution of X is symmetric: X and $-X$ have the same distribution.
Given $X=x$, Y is drawn from $Uniform(x^2-0.1,x^2+0.1)$, an interval of width 0.2 centered at x^2.

Step 2: Find the conditional mean of Y.
For a uniform distribution, the mean is the midpoint of the interval. So

$$E[Y\mid X=x]=\frac{(x^2-0.1)+(x^2+0.1)}{2}=x^2$$

Step 3: Recall the formula for correlation and what we need.

$$correlation(X,Y)=\frac{Cov(X,Y)}{\sigma_X\sigma_Y}$$

We only need the value of $Cov(X, Y)$, because if it turns out to be 0, the correlation is 0 as long as σ_X and σ_Y are finite and nonzero, which they are here.

$$Cov(X, Y) = E[XY] - E[X]E[Y]$$

Step 4: Compute E[XY] using the tower property (law of total expectation).

$$E[XY] = E\Big[E[XY \mid X]\Big] = E\Big[X \cdot E[Y \mid X]\Big] = E[X \cdot X^2] = E[X^3]$$

Now, $X \sim Uniform(-1, 1)$ is symmetric about 0, and X^3 is an odd function. For a distribution symmetric about 0, the expectation of any odd function of X is 0, because the positive and negative contributions cancel exactly. Explicitly:

$$E[X^3] = \int_{-1}^{1} x^3 \cdot \frac{1}{2}\, dx = \frac{1}{2}\left[\frac{x^4}{4}\right]_{-1}^{1} = \frac{1}{2}\left(\frac{1}{4} - \frac{1}{4}\right) = 0$$

So $E[XY] = 0$.

Step 5: Compute Cov(X,Y) and hence the correlation. We already have $E[X] = 0$. So

$$Cov(X, Y) = E[XY] - E[X]E[Y] = 0 - (0)(E[Y]) = 0$$

Since the covariance is exactly 0, and both σ_X and σ_Y are finite positive numbers here, the correlation is

$$correlation(X, Y) = \frac{0}{\sigma_X \sigma_Y} = 0$$

Final Answer:
The correlation between X and Y is 0.

$$\boxed{0}$$

Quick Tip: Use the tower property: $E[XY] = E[X \cdot E[Y|X]] = E[X^3]$, and note that X^3 is an odd function of a symmetric variable, so its expectation is 0, making the covariance and hence the correlation 0.

64. Let $A_{5\times5}$ be a matrix such that each of its elements follows $Bernoulli(p = 0.50)$ distribution independently.

The probability that the row-sum of the second row and the column-sum of the third column are both equal to 3 is _____. (Rounded off to two decimal places)

Correct Answer: 0.10

Solution:

Step 1: Identify the cells involved.

The row-sum of the second row is the sum of the 5 entries $A_{21}, A_{22}, A_{23}, A_{24}, A_{25}$.

The column-sum of the third column is the sum of the 5 entries $A_{13}, A_{23}, A_{33}, A_{43}, A_{53}$.

Both sums share exactly one common cell, A_{23}, which sits at the intersection of row 2 and column 3.

So the row-sum and the column-sum are not independent, they are linked through this one shared entry.

Step 2: Split the row and column into the shared cell and the rest.

Let $X = A_{23}$, a single $Bernoulli(0.5)$ variable.

Let R be the sum of the other 4 cells in row 2, that is $A_{21} + A_{22} + A_{24} + A_{25}$.

Let C be the sum of the other 4 cells in column 3, that is $A_{13} + A_{33} + A_{43} + A_{53}$.

Since all 25 entries are independent, X, R and C are three independent random variables, and both R and C follow $Binomial(4, 0.5)$.

Row-sum $= R + X$ and column-sum $= C + X$.

We need $P(R + X = 3 \text{ and } C + X = 3)$.

Step 3: Write the Binomial(4, 0.5) probabilities.

For $k = 0, 1, 2, 3, 4$, $P(Binomial(4, 0.5) = k) = \binom{4}{k}\left(\frac{1}{2}\right)^4 = \frac{\binom{4}{k}}{16}$.

$$P(0) = \frac{1}{16}, \quad P(1) = \frac{4}{16}, \quad P(2) = \frac{6}{16}, \quad P(3) = \frac{4}{16}, \quad P(4) = \frac{1}{16}$$

Step 4: Condition on the value of the shared cell X.

Case $X = 0$ (probability 0.5): we need $R = 3$ and $C = 3$.

$$P(R = 3) \cdot P(C = 3) = \frac{4}{16} \cdot \frac{4}{16} = \frac{16}{256} = 0.0625$$

Contribution: $0.5 \times 0.0625 = 0.03125$

Case $X = 1$ (probability 0.5): we need $R = 2$ and $C = 2$.

$$P(R = 2) \cdot P(C = 2) = \frac{6}{16} \cdot \frac{6}{16} = \frac{36}{256} = 0.140625$$

Contribution: $0.5 \times 0.140625 = 0.0703125$

Step 5: Add the two cases.

$$P = 0.03125 + 0.0703125 = 0.1015625$$

Rounded to two decimal places, this is 0.10.

Final Answer:

The probability that the row-sum of row 2 and the column-sum of column 3 are both equal to 3 is 0.10.

0.10

Quick Tip: Split the row-2 sum and column-3 sum around the one cell they share, A_{23}. Condition on A_{23} being 0 or 1, then use $Binomial(4, 0.5)$ for the remaining four cells in each.

65. Let $A = \left(I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)$ be a matrix, where $\mathbf{1} = (1, 1, 1, \ldots, 1)^T \in \mathbb{R}^n$ and I_n is the identity matrix of order n.

The value of $\max_S x^T A x$, where $S = \{x \in \mathbb{R}^n \mid x^T x = 1\}$, is _____. (Answer in integer)

Correct Answer: 1

Solution:

Step 1: Recognize the structure of the matrix.

The matrix $A = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ is the well known centering matrix used in statistics to subtract the mean from a vector.

It is symmetric, since $A^T = I_n^T - \frac{1}{n}(\mathbf{1}\mathbf{1}^T)^T = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T = A$.

It is also idempotent, meaning $A^2 = A$, which we can check directly.

Step 2: Show A is idempotent.

$$A^2 = \left(I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)\left(I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right) = I_n - \frac{2}{n}\mathbf{1}\mathbf{1}^T + \frac{1}{n^2}\mathbf{1}(\mathbf{1}^T\mathbf{1})\mathbf{1}^T$$

Since $\mathbf{1}^T\mathbf{1} = n$ (the dot product of the all ones vector with itself, summing n ones), the last term becomes $\frac{1}{n^2}\mathbf{1} \cdot n \cdot \mathbf{1}^T = \frac{1}{n}\mathbf{1}\mathbf{1}^T$.

$$A^2 = I_n - \frac{2}{n}\mathbf{1}\mathbf{1}^T + \frac{1}{n}\mathbf{1}\mathbf{1}^T = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T = A$$

So A is idempotent and symmetric, which makes it a projection matrix.

Step 3: Use the property of symmetric idempotent matrices.

A real symmetric matrix is orthogonally diagonalizable, and all its eigenvalues are real.

For an idempotent matrix, if λ is an eigenvalue with eigenvector v, then $Av = \lambda v$ implies $A^2v = \lambda^2 v$, but also $A^2v = Av = \lambda v$.

So $\lambda^2 v = \lambda v$, which forces $\lambda^2 = \lambda$, so λ can only be 0 or 1.

For a symmetric matrix, $x^T A x$ for a unit vector x is maximized exactly at the largest eigenvalue of A (the Rayleigh quotient result).

So we only need to check whether A actually has an eigenvalue equal to 1.

Step 4: Find an eigenvector with eigenvalue 1.

Take any vector v that is orthogonal to $\mathbf{1}$, meaning $\mathbf{1}^T v = 0$ (its entries sum to zero). Such vectors exist whenever $n \geq 2$.

$$Av = \left(I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)v = v - \frac{1}{n}\mathbf{1}(\mathbf{1}^T v) = v - \frac{1}{n}\mathbf{1}\cdot 0 = v$$

So every vector orthogonal to $\mathbf{1}$ is an eigenvector of A with eigenvalue 1. This subspace has dimension $n-1$, so 1 is indeed an eigenvalue, with multiplicity $n-1$.
Picking any unit vector x from this subspace gives $x^T Ax = x^T x = 1$.

Step 5: Conclude the maximum.
Since the eigenvalues of A are only 0 and 1, and 1 is attained, the largest eigenvalue is 1.
So the maximum of $x^T Ax$ over all unit vectors x is exactly 1.

Final Answer:
The value of $\max_S x^T Ax$ is 1.

$$\boxed{1}$$

Quick Tip: Check that A is symmetric and idempotent ($A^2 = A$), so its eigenvalues can only be 0 or 1. The maximum of $x^T Ax$ on the unit sphere equals the largest eigenvalue.