Editorial-W9A2: Core Concepts of Python, NumPy, Pandas & Algorithm Analysis

Question 1 (MCQ)

Sarah is working on a project where she needs to create a NumPy array to store the ages of 5 students: 18, 20, 22, 19, and 21. Which code snippet correctly creates this array?

A) np.array(18, 20, 22, 19, 21)

B) np.array([18, 20, 22, 19, 21])

C) np.array({18, 20, 22, 19, 21})

D) np.array((18, 20, 22, 19, 21))

Answer: B

Explanation: In NumPy, arrays are created using np.array() with a list or tuple of elements. Option B uses a list [18, 20, 22, 19, 21], which is the correct syntax. Option A is invalid (no list/tuple), Option C uses a set (not typical for ordered data), and Option D uses a tuple (valid but less common). NumPy requires a sequence like a list or tuple to create an array. The list syntax in B is the most standard approach.

Question 2 (MCQ)

John has a Pandas DataFrame with student names and scores. He wants to find the average score. His DataFrame is:

Name	Score
John	85
Lisa	90
Mark	75

Which code calculates the mean score?

A) df.mean()

B) df['Score'].mean()

C) df['Score'].average()

D) df.average()

Answer: B

Explanation: The correct method is df['Score'].mean() because it selects the 'Score' column and computes its mean (85 + 90 + 75) / 3 = 83.33. Option A computes the mean of all numeric columns, but we only want 'Score'. Options C and D are invalid (no average() method).

In Pandas, mean() calculates the average of a Series (a single column), making B the precise choice.

Question 3 (NAT)

Alice is using NumPy to calculate the sum of an array of sales: [[10, 20], [30, 40]]. What is the result of np.sum(sales)?

Answer: 100

Explanation: np.sum([10, 20, 30, 40]) adds the elements: 10 + 20 + 30 + 40 = 100.

The np.sum() function in NumPy computes the total of all elements in the array, so the answer is 100.

Question 4 (MSQ)

Tom is sorting a list of numbers [5, 2, 9, 1] using Python's built-in sort() method. Which of the following are true? (Select all that apply)

A) The sorted list will be [1, 2, 5, 9]

B) The time complexity of sort() is O(n log n)

C) The original list will be modified

D) The time complexity is $O(n^2)$

Answers: A, B, C

Explanation:

- A: sort() arranges elements in ascending order: [1, 2, 5, 9].
- B: Python's sort() uses Timsort, with O(n log n) time complexity.
- C: sort() modifies the list in place.
- D: $O(n^2)$ is incorrect; it applies to algorithms like bubble sort, not Timsort.

Python's sort() is efficient and modifies the original list, making A, B, and C correct.

Question 5 (MCQ)

Emma is analyzing a dataset in Pandas with 100 rows. She wants to see the first 5 rows to check the data. Which method should she use?

A) df.head()

B) df.tail()

C) df.first()

D) df.top()

Answer: A

Explanation: df.head() displays the first 5 rows of a DataFrame by default. df.tail() shows the last 5, and C/D are invalid methods.

head() is the standard Pandas method for previewing the start of a DataFrame.

Question 6 (MCQ)

Mike is building a Python program to check whether a number exists in a sorted list of IDs using binary search. He uses the list [1, 3, 5, 7, 9] and searches for the value 7. He tracks how many comparisons the algorithm performs before it finds the target.

How many comparisons will it take in the worst case to find the number 7 using binary search?

A) 1

B) 2

C) 3

D) 4

Answer: C

Explanation: Binary search halves the search space each step:

- Step 1: Check 5 (middle), 7 > 5, go right [7, 9].
- Step 2: Check 7 (middle), found! With implementation details, it may take up to 3 steps ($\log_2(5) \approx 3$).

Binary search is O(log n), and for 5 elements, it takes at most 3 comparisons.

Question 7 (MSQ)

Rachel is learning about time complexity. She writes a function to find the maximum value in a list by comparing each element. Which statements are true? (Select all that apply)

A) The time complexity is O(n)

B) The time complexity is $O(n^2)$

C) It requires n-1 comparisons for a list of n elements

D) It can be optimized to O(log n)

Answers: A, C

Explanation:

- A: Linear search for the maximum is O(n).
- B: $O(n^2)$ is for nested loops, not this case.
- C: n-1 comparisons are needed to check all elements.
- D: O(log n) applies to binary search, not max-finding.

Finding the max in an unsorted list is a linear operation, confirming A and C.

Question 8(MCQ)

Sophie has a Pandas Series of temperatures: [25, 30, 28, 27]. What is the result of temperatures.median()?

A) 27

B) 27.5

C) 28

D) 30

Answer: B

Explanation: Sort the series: [25, 27, 28, 30]. Median of 4 numbers is the average of the middle two: (27 + 28) / 2 = 27.5.

The median is the middle value(s) in an ordered list, averaged if even-length.

Question 9 (MCQ)

Liam is testing a hypothesis about coin flips. He assumes the coin is fair (p = 0.5), and flips it 10 times. His assumption is an example of:

A) Null hypothesis

B) Alternative hypothesis

C) Type I error

D) Type II error

Answer: A

Explanation: The null hypothesis (H_o) is the default assumption, here that the coin is fair (p = 0.5). B is the alternative ($p \neq 0.5$), and C/D are errors in testing.

Hypothesis testing starts with a null hypothesis, which Liam assumes.

Question 10(MSQ)

Olivia is using bubble sort on [4, 3, 2, 1]. Which are true? (Select all that apply)

A) The first pass results in [3, 4, 1, 2]

B) The time complexity is $O(n^2)$

C) It will take 3 passes to fully sort

D) It compares adjacent elements

Answers: B, D

Explanation:

- A: First pass is [3, 2, 1, 4], not [3, 4, 1, 2].
- B: Bubble sort is $O(n^2)$.
- C: It takes 3 passes, but this depends on implementation (not guaranteed).
- D: Bubble sort swaps adjacent elements.Bubble sort's key features are its $O(n^2)$ complexity and adjacent comparisons, making B and D correct.

Question 11 (MCQ)

Noah is filtering a Pandas DataFrame of employees to find those with salaries > 50000. His DataFrame has a 'Salary' column. Which code works?

A) df[df['Salary'] > 50000]

B) df['Salary'] > 50000

C) df.filter('Salary' > 50000)

D) df[df.Salary > 50000]

Answer: A

Explanation: df[df['Salary'] > 50000] filters rows where 'Salary' exceeds 50000. B returns a boolean Series, C is invalid syntax, and D needs quotes around 'Salary'.

Boolean indexing in Pandas uses the syntax in A to filter DataFrames.

Question 12 (MCQ)

Ava is analyzing time complexity. She has an algorithm that doubles its input size but takes 4 times longer to run. What is its time complexity?

A) O(n)

B) $O(n^2)$

C) O(log n)

D) O(n log n)

Answer: B

Explanation: If input size doubles ($n \rightarrow 2n$) and runtime quadruples ($t \rightarrow 4t$), then $t \propto n^2$, so $O(n^2)$.

Quadratic time complexity means runtime grows with the square of input size, matching the scenario.

Question 13(MSQ)

Ethan is working with NumPy arrays. He has arr = np.array([1, 2, 3]). Which operations are valid? (Select all that apply)

A) arr + 2

B) arr * 3

C) arr.append(4)

D) arr[0] = 5

Answers: A, B, D

Explanation:

- A: Adds 2 to each element: [3, 4, 5].
- B: Multiplies each by 3: [3, 6, 9].
- C: NumPy arrays don't have append() (lists do).
- D: Assigns 5 to index 0: [5, 2, 3].NumPy supports element-wise operations and indexing, but not list-like methods like append().

Question 14 (NAT)

Isabella is analyzing the runtime of an algorithm implemented in Python. The algorithm has a time complexity of $O(n^2)$, and she runs it on a dataset containing 10 elements. She measures that each basic operation takes exactly 1 millisecond.

Assuming the algorithm performs exactly as many operations as the time complexity suggests, how many milliseconds will the entire algorithm take to complete?

Ans⇒ 100 ms

Explanation: An algorithm with $O(n^2)$ complexity performs approximately n^2 operations for an input of size n.

Here, n = 10, so number of operations = $10^2 = 100$

Each operation takes 1 ms, so total time = 100 × 1 = 100 ms

This makes the final answer 100 ms.

Question 15(MSQ)

Mia is merging two Pandas DataFrames on a common 'ID' column. Which statements are true? (Select all that apply)

A) She can use df1.merge(df2, on='ID')

B) The result includes only matching rows if using default settings

C) She must sort the DataFrames first

D) She can specify how='outer' for all rows

Answers: A, B, D

Explanation:

- A: Correct syntax for merging.

- B: Default is inner join (matching rows only).
- C: Sorting isn’t required for merging.
- D: how='outer' includes all rows.Pandas merge() is flexible, with inner join as default and outer as an option.