

DAA

1. Fibonacci Numbers and Step Count

Q1. What is the Fibonacci series?

Ans. A sequence where each number is the sum of the two preceding ones, starting from 0 and 1.

Q2. What are the two ways to implement Fibonacci?

Ans. Iterative and recursive methods.

Q3. What is a step count?

A. The number of basic operations executed by an algorithm; used to measure complexity.

Q4. What is the time complexity of recursive Fibonacci?

Ans. $O(2^n)$.

Q5. How does dynamic programming improve Fibonacci?

Ans. It stores previously computed values to avoid recalculation, reducing complexity to $O(n)$.

2. Huffman Coding or Job sequencing



Oral Questions & Answers:

Q1. What is Huffman Encoding?

A1. It is a lossless data compression technique that assigns shorter binary codes to more frequent characters.

Q2. Which approach is used in Huffman Coding?

A2. Greedy approach — it always combines the two smallest frequencies at each step.

Q3. Why do we use a priority queue?

A3. To efficiently pick the two nodes with the smallest frequencies.

Q4. What is the time complexity?

A4. $O(n \log n)$, due to priority queue operations.

Q5. What is meant by lossless compression?

A5. It means no data is lost during compression or decompression.



Oral Questions & Answers:

Q1. What is the objective of Job Sequencing?

A1. To schedule jobs to get the **maximum total profit** within given deadlines.

Q2. What is the greedy choice here?

A2. Always select the job with the **highest profit first**.

Q3. Why do we sort jobs by profit?

A3. Because we want to maximize profit — higher profit jobs are given priority.

Q4. What is the time complexity?

A4. $O(n^2)$, since for each job we check available slots.

Q5. What data structures are used?

A5. Arrays (for slots and result), and a Job class for job details.

Q6. Can deadlines exceed number of jobs?

A6. Yes, but we consider only up to the maximum number of slots available.

3. Fractional Knapsack Problem

◆ Oral Viva Questions and Answers

1. What is the Fractional Knapsack problem?

It is a problem where we have to maximize profit by selecting items based on their value-to-weight ratio, and we are allowed to take fractions of items.

2. Which algorithmic technique is used in this problem?

The **Greedy algorithm** technique is used.

3. Why do we sort by value/weight ratio?

Because items with higher profit per weight give the most benefit; hence, we select them first to maximize total value.

4. What is the difference between 0/1 Knapsack and Fractional Knapsack?

0/1 Knapsack

Fractional Knapsack

Item is either fully taken or left.

Item can be taken partially.

Solved using Dynamic Programming. Solved using Greedy algorithm.

5. What is the time complexity of this algorithm?

The time complexity is **$O(n \log n)$** due to sorting the ratios.

6. What is the greedy choice property?

At each step, choose the item with the **highest value-to-weight ratio**.

7. Can Fractional Knapsack be solved using dynamic programming?

It can, but the greedy approach is more efficient and optimal for this problem.

8. What data structures are used here?

- Arrays for storing values and weights.
 - 2D array for storing value/weight ratios.
-

9. What does `Arrays.sort(ratio, (a,b) -> Double.compare(b[0],a[0]))` do?

It sorts the ratio array in descending order of value-to-weight ratio using a lambda comparator.

4. 0/1 Knapsack Problem using Dynamic Programming

Conceptual Questions & Answers

1. What is the 0/1 Knapsack problem?

→ It is a problem where we must select items with given weights and values to fill a knapsack of limited capacity such that the total value is maximized. Each item can either be included (1) or excluded (0).

2. Why is it called “0/1” Knapsack?

→ Because every item can be either taken (1) or not taken (0), no fractional part is allowed.

3. What is the difference between 0/1 Knapsack and Fractional Knapsack?

→ In **0/1 Knapsack**, items cannot be divided.

In **Fractional Knapsack**, we can take fractional parts of items.

4. What is the input for the Knapsack problem?

→ The weights and values of items, and the total capacity of the knapsack.

5. What is the output of the Knapsack problem?

→ The maximum value that can be carried in the knapsack without exceeding the weight capacity.

6. Which algorithmic technique is used to solve 0/1 Knapsack?

→ **Dynamic Programming.**

7. What is the time complexity of the 0/1 Knapsack algorithm?

→ $O(n \times W)$ where n is the number of items and W is the knapsack capacity.

8. What is the space complexity?

→ $O(n \times W)$ for the 2D DP table.

9. What is the formula used in the DP approach?

→

if ($wt[i-1] \leq w$)

$dp[i][w] = \max(val[i-1] + dp[i-1][w - wt[i-1]], dp[i-1][w]);$

else

$dp[i][w] = dp[i-1][w];$

10. What does $dp[i][w]$ represent?

→ It represents the **maximum value** that can be achieved using the first i items and a knapsack capacity of w .

11. Can the 0/1 Knapsack problem be solved using recursion?

→ Yes, but it is less efficient. Dynamic Programming is preferred to avoid repeated calculations.

12. What is the output for given input ($val=\{60,100,120\}$, $wt=\{10,20,30\}$, $W=50$)?

→ Maximum value in knapsack = **220**.

13. What happens if all items are heavier than the knapsack capacity?

→ The total value will be **0** because no item can be included.

14. Can greedy method be applied to 0/1 Knapsack?

→ No, greedy does not work correctly for 0/1 Knapsack because we can't take fractional items.

15. Why is dynamic programming better than recursion here?

→ Because it avoids recalculating the same subproblems, saving time and improving efficiency.

5. 8-Queens Problem

Q1. What is the 8-Queens problem?

Ans. Place 8 queens on a chessboard so that no two attack each other.

Q2. Which technique is used?

Ans. Backtracking.

Q3. When is a position safe?

Ans. No other queen exists in the same row, column, or diagonal.

Q4. What does backtracking mean?

Ans. Systematically searching for a solution by exploring and undoing wrong choices.

Q5. Time complexity?

Ans. Approximately $O(N!)$.

ML

1. Uber

Q. What is outlier?

Ans. An **outlier** is a **value that doesn't fit the pattern** of the rest of the data.

Example (Uber Dataset):

If most Uber fares are between ₹100 and ₹1000,
but one fare is ₹10,000 that ₹10,000 is an **outlier**.

Q. What is Linear Regression ?

Ans. **Linear Regression** is a **supervised machine learning algorithm** used to **predict a continuous (numeric) value** based on one or more input variables.

It is relationship between a dependent variable and one or more independent variables.

Equation of Linear Regression:

$$Y = mX + c$$

Where:

- $Y \rightarrow$ Target variable (e.g., fare amount)
- $X \rightarrow$ Input variable (e.g., distance)
- $m \rightarrow$ Slope (how much fare increases per unit distance)
- $c \rightarrow$ Intercept (base fare when distance is zero)

Example (Uber Dataset):

You want to **predict the fare amount** based on the **distance** between pickup and drop location.

Linear Regression finds the **best straight-line relationship** between these two variables.

Q.What is Random Forest Algorithm?

Ans. **Random Forest Algorithm** is a **supervised machine learning algorithm** used for both **classification and regression** tasks.

Concept:

- It is called a “forest” because it combines many decision trees to make a more accurate and stable prediction.
- Each tree in the forest gives a result, and the final output is based on the majority vote (for classification) or average (for regression).

Q. What is data preprocessing?

Ans. **ChatGPT said:**

Data Preprocessing is the **first and most important step** in machine learning and data analysis.

It means **cleaning, organizing, and preparing raw data** so it can be used effectively by machine learning models.

2.Email Spam Detection using KNN and SVM (Binary Classification)

Question	Answer
----------	--------

1. What is binary classification?	It is a task where the output variable has only two possible values, such as Spam or Not Spam.
2. What is the purpose of KNN in this experiment?	KNN is used to classify emails based on the majority of nearest neighbors' classes.
3. What is the role of SVM here?	SVM separates spam and non-spam emails using a hyperplane that maximizes the margin between two classes.
4. What is the value of 'K' in KNN?	K represents the number of nearest neighbors considered for voting, usually chosen by testing different values.
5. Why do we use CountVectorizer?	It converts email text into numerical form that machine learning models can understand.
6. Which model performed better in your output?	The SVM model, as it showed higher accuracy and better precision.
7. What is accuracy?	Accuracy is the ratio of correctly predicted instances to the total instances.
8. What is overfitting?	When a model performs well on training data but poorly on new, unseen data.
9. What is a hyperplane in SVM?	A hyperplane is a decision boundary that separates the classes in the feature space.
10. What type of learning is used here?	Supervised Learning.

3. customer churn prediction using a neural network (ANN).

Question	Answer
1. What is the goal of this experiment?	To predict whether a bank customer will leave the bank (churn) using a neural network.
2. What type of learning is used here?	Supervised learning, because we use labeled data (Exited column).
3. Why do we normalize the data?	To ensure all features are on the same scale, improving model convergence.
4. What activation functions are used and why?	ReLU in hidden layers (fast and effective), Sigmoid in output layer (for binary output).
5. What is binary cross-entropy?	It measures the difference between actual and predicted probabilities in binary classification.

6. What is the purpose of using the Adam optimizer?	It combines the advantages of AdaGrad and RMSProp for efficient training.
7. What is a confusion matrix?	It shows the number of correct and incorrect predictions for each class.
8. How can accuracy be improved?	By tuning hyperparameters, adding dropout, or increasing epochs.
9. What does the output '1' or '0' mean?	'1' means the customer will leave, and '0' means the customer will stay.
10. Why do we use a sigmoid function in the output layer?	Because it converts the output into a probability between 0 and 1.

4. K-Nearest Neighbors algorithm on the diabetes.csv dataset.

1. What is K-Nearest Neighbors?

Answer:

KNN is a supervised machine learning algorithm that classifies new data points based on the majority class of its **K nearest neighbors** in the feature space.

2. How does KNN work?

Answer:

It calculates the distance between the new data point and all other data points, selects the **K closest points**, and assigns the most common class among them to the new data point.

3. What is the default value of K in scikit-learn's KNN classifier?

Answer:

The default value of K (n_neighbors) is **5**.

4. What type of distance is used in KNN?

Answer:

By default, **Euclidean distance** is used, though other metrics like Manhattan or Minkowski can also be applied.

5. What happens if K is too small or too large?

Answer:

- **Too small K:** Model becomes sensitive to noise (overfitting).
- **Too large K:** Model becomes too smooth and may misclassify (underfitting).

6. What is the purpose of the confusion matrix?

Answer:

It helps visualize the performance of a classification model by showing how many predictions were correct and incorrect.

7. What are Precision and Recall?

Answer:

- **Precision** measures how many predicted positives are actually correct.
- **Recall** measures how many actual positives were correctly identified.

8. Why did we replace zero values with the mean?

Answer:

Zero is not a valid value for medical measurements like glucose or blood pressure. Replacing with the mean maintains consistency and avoids bias in training.

9. What is the error rate?

Answer:

It is the fraction of predictions that are incorrect:

$$\text{Error Rate} = 1 - \text{Accuracy}$$

10. Is KNN a parametric or non-parametric algorithm?

Answer:

KNN is **non-parametric**, meaning it does not make assumptions about the data distribution.

BT

1.smart contract on a test network, for Bank account of a customer

Oral Viva Questions and Answers

1. What is a Smart Contract?

A smart contract is a self-executing program stored on a blockchain that automatically enforces and executes the terms of an agreement without intermediaries.

2. What is the purpose of the mapping keyword in Solidity?

mapping is used to store key–value pairs. In this program, it maps an address (key) to its balance (value).

3. Why is the deposit function marked as payable?

Because it needs to receive Ether from the sender's wallet. Without payable, a function cannot accept Ether.

4. What is the use of the require statement?

require checks a condition before executing further. If the condition fails, it reverts the transaction and refunds unused gas.

5. How can we check our account balance?

By calling the `getBalance()` function, which returns the stored balance of the caller's address.

6. What happens if we try to withdraw more than our balance?

The transaction fails because the require condition is not met, preventing over-withdrawal.

7. What is `msg.sender` and `msg.value`?

- `msg.sender` → The address of the account calling the function.
 - `msg.value` → The amount of Ether sent with the transaction.
-

8. What network can we use to test this contract?

You can use **Remix VM**, or test networks like **Goerli**, **Sepolia**, or **Ganache** for local testing.

9. What does the `transfer()` function do?

It transfers Ether from the contract to a specified address securely.

10. Why do we use `view` keyword in the `getBalance()` function?

Because it only reads data from the blockchain and does not modify it.

2. program in solidity to create Student data

Oral Questions with Answers

1. What is a structure in Solidity?

Answer:

A structure is a user-defined data type that groups variables of different data types under a single name, useful for representing complex data like a student record.

2. Why do we use arrays in this program?

Answer:

Arrays store multiple student records dynamically, allowing easy addition and retrieval of data.

3. What is the purpose of the fallback function?

Answer:

The fallback function is executed when no matching function is found or when Ether is sent directly to the contract.

4. What happens if you send Ether to this contract?

Answer:

The fallback function will be called, and the Ether will be stored in the contract's balance.

5. How can you check the balance of the contract?

Answer:

By calling the `getBalance()` function, which returns the Ether amount held by the contract.

6. What is 'gas' in Ethereum?

Answer:

Gas is the computational cost required to perform operations on the Ethereum blockchain. It ensures that resources are used efficiently.

7. What is the transaction fee?

Answer:

Transaction fee = Gas Used × Gas Price.

It's paid by the sender for executing operations on the blockchain.

8. What version of Solidity is used here?

Answer:

Solidity version ^0.8.0, which includes built-in overflow checks and updated compiler features.

9. What is the use of the require statement in `getStudent()` function?

Answer:

It ensures that the provided index is valid; if not, it throws an error and reverts the transaction.

10. How are structures different from classes?**Answer:**

Structures only store data, whereas classes (not in Solidity) can also contain complex behaviors. Solidity uses contracts as class-like entities.

5. K-Means clustering/ hierarchical clustering on sales_data_sample.csv dataset.

Question	Answer
1. What is clustering?	Clustering is an unsupervised learning technique that groups similar data points together.
2. What is K-Means algorithm?	It partitions data into k clusters, where each data point belongs to the cluster with the nearest mean (centroid).
3. What is the purpose of the Elbow Method?	It helps to determine the optimal number of clusters by observing where the inertia curve starts to flatten.
4. What is Inertia in K-Means?	Inertia measures the sum of squared distances between each data point and its cluster's centroid. Lower inertia means better clustering.
5. Why do we standardize the data before clustering?	To ensure that all features contribute equally to the distance calculation, preventing features with larger scales from dominating.
6. What are centroids?	Centroids are the mean points representing each cluster's center.

7. Is K-Means supervised or unsupervised?	K-Means is an unsupervised algorithm.
8. What are the limitations of K-Means?	It requires specifying k beforehand, is sensitive to outliers, and may converge to local minima.
9. What is hierarchical clustering?	It builds a tree-like structure (dendrogram) to show how data points can be grouped step-by-step without predefining k .
10. Which distance metric does K-Means use?	It generally uses the Euclidean distance to measure closeness between points.