Questions Based on Assignments :

Assignment A-1 : Fibonacci series

1. **What is the Fibonacci Sequence of numbers?**
   The Fibonacci Sequence is a series where each number is the sum of the two preceding ones, starting from 0 and 1 (e.g., 0, 1, 1, 2, 3, 5, 8...).

2. **How do the Fibonacci numbers work?**
   Each Fibonacci number is derived by adding the previous two numbers, following the formula $F(n)=F(n-1)+F(n-2)$ with $F(0)=0F(0) = 0F(0)=0$ and $F(1)=1F(1) = 1F(1)=1$.

3. **What is the Golden Ratio?**
   The Golden Ratio, approximately 1.618, is the limit of the ratio of consecutive Fibonacci numbers, representing an ideal proportion in art, nature, and architecture.

4. **What is the Fibonacci Search technique?**
   Fibonacci Search is a search algorithm for sorted arrays, using Fibonacci numbers to split the array, which reduces comparisons and is more efficient for larger datasets.

5. **What is the real application for Fibonacci series?**
   The Fibonacci series is used in algorithm design, financial modeling, biological studies (like branching in trees and leaf arrangements), and computer data structures.

Assignment A-2 : Huffman Encoding

1. **What is Huffman Encoding?**

Huffman Encoding is a compression technique that assigns shorter binary codes to more frequent characters and longer codes to less frequent ones, reducing the overall size of data.

2. **How many bits may be required for encoding the message ‘mississippi’?**

Encoding 'mississippi' requires 28 bits, with shorter codes for frequently occurring letters like 'i' and 's'.

3. **Which tree is used in Huffman encoding? Give one Example**

A binary tree is used in Huffman encoding. For example, in encoding 'mississippi', 'i' and 's' (most frequent) have the shortest codes, while 'm' and 'p' (least frequent) have longer codes.

4. **Why is Huffman coding lossless compression?**

Huffman coding is lossless because it preserves the exact original data, allowing it to be perfectly reconstructed from the compressed data.

Assignment A-3 : fractional Knapsack problem using Greedy

1. **What is Greedy Approach?**
   The Greedy Approach is an algorithmic paradigm that builds a solution incrementally by choosing the locally optimal choice at each step, with the hope of finding a global optimum.

2. **Explain the concept of fractional knapsack.**
   The fractional knapsack problem allows the breaking of items into smaller pieces. Given a set of items, each with a weight and value, the goal is to maximize the total value in the knapsack, where items can be divided.

3. **Difference between Fractional and 0/1 Knapsack.**
   In the Fractional Knapsack, items can be divided into smaller parts, while in the 0/1 Knapsack, items must be taken whole or not at all. This results in different approaches and solutions for each problem.

4. **Solve one example based on Fractional Knapsack.**

   Items:
   - Item 1: Weight = 10 kg, Value = 60
   - Item 2: Weight = 20 kg, Value = 100
   - Item 3: Weight = 30 kg, Value = 120

   Knapsack Capacity: 50 kg

➔ **Solution:**

1. **Calculate value-to-weight ratio:**
   - Item 1: 6 per kg
   - Item 2: 5 per kg
   - Item 3: 4 per kg
2. **Sort items by ratio: Item 1, Item 2, Item 3**.
3. **Fill the knapsack:**
   - Take Item 1 (10 kg, 60).
   - Take Item 2 (20 kg, 100).
   - Take half of Item 3 (15 kg, 60).
4. **Total value = 60 + 100 + 60 = 220.**

Assignment A-4 : 0-1 Knapsack problem using dynamic programming

1. **What is Dynamic Approach?**
   An algorithmic technique that breaks a problem into simpler subproblems, storing results to avoid redundant calculations.
2. **Explain the concept of 0/1 knapsack.**
   A problem where you select items with given weights and values to maximize total value without exceeding a weight limit, with each item being either included or excluded.
3. **Difference between Dynamic and Branch and Bound Approach. Which is best?**
   Dynamic Programming stores results of overlapping subproblems, while Branch and Bound explores branches and prunes unpromising ones. The best approach depends on the problem context.
4. **Solve one example based on 0/1 knapsack.**
   → Example:
   - **Items:**
   - Item 1: Weight = 1 kg, Value = 1
   - Item 2: Weight = 3 kg, Value = 4
   - Item 3: Weight = 4 kg, Value = 5
   - Item 4: Weight = 5 kg, Value = 7
   - **Knapsack Capacity: 7 kg**

   Solution:
   Maximum value is 8, achieved by including Item 4 (Weight = 5 kg, Value = 7) and Item 1 (Weight = 1 kg, Value = 1).

Assignment A-5 : n-Queens matrix

1. **What is backtracking? Give the general Procedure.**
   Backtracking is a method for solving problems by trying different options and going back if a choice doesn't work. The general procedure is to explore all possible solutions, check if they are valid, and undo choices when needed.
2. **Give the problem statement of the n-queens problem. Explain the solution.**
   The n-queens problem asks how to place n queens on an n × n chessboard so that no two queens can attack each other. The solution involves placing queens one row at a time and making sure that no two queens are in the same row, column, or diagonal.

3. **Write an algorithm for N-queens problem using backtracking.**

- Initialize an empty N x N chessboard.
- Check if placing a queen at (row, col) is safe.
- If all queens are placed, print the board.
- For each row in the current column:
- If safe, place a queen and move to the next column.
- If not successful, remove the queen (backtrack).
- Start with the first column.

4. **Why is it applicable to N=4 and N=8 only?**

While backtracking can be used for any N, N=4 and N=8 are popular examples because they are easier to understand and visualize. They help demonstrate how the backtracking method works effectively**.**