1. **DFS (Depth-First Search):**

- **Definition:** DFS is an algorithm for traversing or searching tree or graph data structures. It starts at the root node and explores as far as possible along each branch before backtracking.
- **Algorithm:**
  1. Start at the root node.
  2. Explore the first adjacent unvisited node.
  3. If there are no unvisited adjacent nodes, backtrack to the previous node.
  4. Repeat steps 2-3 until all nodes are visited.
- **Example:** Let's say we have a graph with nodes A, B, C, D, E, and F. Starting from node A, DFS would explore A, then move to an adjacent unvisited node, say B, then to C, then backtrack to B if necessary, and so on until all nodes are visited.

2. **BFS (Breadth-First Search):**

- **Definition:** BFS is another algorithm for traversing or searching tree or graph data structures. It explores all the neighbour nodes at the present depth prior to moving on to the nodes at the next depth level.
- **Algorithm:**
  1. Start at the root node.
  2. Explore all the neighbouring nodes at the present depth.
  3. Move to the next depth level and repeat step 2.
  4. Repeat until all nodes are visited.
- **Example:** Using the same example graph, BFS would start at node A, then visit all immediate neighbours of A (like B and C), then move to their neighbours (like D, E, and F), exploring nodes level by level until all nodes are visited.

3. A* algorithm:

- **Definition:** A* (pronounced "A-star") is a widely used pathfinding algorithm that efficiently finds the shortest path between nodes in a weighted graph. It is particularly effective in scenarios where the cost of traversal between nodes varies.
- **Algorithm:**
  1. Initialize an open list and a closed list.
  2. Add the start node to the open list.
  3. While the open list is not empty:
     - Select the node with the lowest total cost from the open list (total cost = cost from start + heuristic estimate to goal).
     - Remove the selected node from the open list and add it to the closed list.
     - If the selected node is the goal, the path is found.
     - Otherwise, expand the selected node by considering its neighbouring nodes:
       - For each neighbouring node:
         - If it is not walkable or is in the closed list, skip it.
         - If it is not in the open list, compute its cost and add it to the open list.

        - If it is already in the open list, update its cost if the new path to it is shorter.
    4. If the open list is empty and the goal has not been reached, there is no path.
- **Example:** Suppose you have a grid where each cell represents a node, and you want to find the shortest path from the start cell to the goal cell. A* algorithm would evaluate possible paths based on both the cost from the start node and a heuristic estimate of the remaining distance to the goal. By iteratively expanding the nodes with the lowest total cost, it efficiently finds the shortest path.

4. Greedy Search Algorithm:

- **Definition:** Greedy search is a simple algorithm used for optimization problems. It makes the best choice at each step with the hope that this will lead to the globally optimal solution.
- **Algorithm:**
    1. Start with an empty solution.
    2. At each step, choose the best possible choice based on some criterion without considering the global context.
    3. Update the solution by adding the chosen element.
    4. Repeat until a termination condition is met, such as reaching a desired solution or exhausting all choices.
- **Example:** The classic example of the greedy algorithm is the coin change problem, where you aim to give change for a certain amount with the minimum number of coins. At each step, the algorithm selects the largest coin that is not greater than the remaining amount.

5. Prim's Algorithm:

- **Definition:** Prim's algorithm is a greedy algorithm that finds a minimum spanning tree for a connected weighted graph. It starts with an arbitrary node and adds the cheapest edge that connects the tree to a new vertex until all vertices are included.
- **Algorithm:**
    1. Start with an arbitrary node as the initial tree.
    2. While there are still vertices not in the tree:
        - Choose the edge with the lowest weight that connects a vertex in the tree to a vertex outside the tree.
        - Add the selected edge and vertex to the tree.
    3. Repeat until all vertices are in the tree.
- **Example:** Consider a network of cities connected by roads with varying lengths (weights). Prim's algorithm would start with one city and iteratively add the shortest road that connects the current network of cities until all cities are included, forming a minimum spanning tree that connects all cities with minimum total road length.

6. Branch and Bound (B&B) and Backtracking are both techniques used to solve optimization problems, like the N-Queens problem.

Branch and Bound (B&B):

- **Definition:** Branch and Bound is a systematic method for solving optimization problems. It involves a systematic enumeration of all possible solutions, while cutting off branches (subsets) of the search space once it is determined that a solution in a particular subset cannot be better than the best solution found so far.
- **Algorithm:**

1. Divide the problem into smaller subproblems.
2. Solve each subproblem optimally.
3. Use bounds to prune the search tree by eliminating portions of the search space that cannot contain the optimal solution.
4. Keep track of the best solution found so far.
5. Repeat steps 1-4 until all subsets are explored.

Backtracking:

- Definition: Backtracking is a systematic way to search for solutions to problems, especially combinatorial optimization problems. It is based on a depth-first search of the solution space. It incrementally builds candidates to the solutions and abandons a candidate as soon as it determines that the candidate cannot possibly be completed to a valid solution.
- **Algorithm:**

1. Start with an empty solution and place the first queen on the first row.
2. Move to the next row and place the next queen in a column where it doesn't conflict with any other queens.
3. If all queens are placed, a solution is found.
4. If a conflict occurs, backtrack to the previous row and try placing the queen in the next column.
5. Repeat steps 2-4 until all queens are placed or no solution is possible.

N-Queens Problem:

- Problem: Given an N×N chessboard, place N queens on the board so that no two queens attack each other (no two queens share the same row, column, or diagonal).

Both Branch and Bound and Backtracking can be used to solve the N-Queens problem. Backtracking, in particular, is commonly employed due to its simplicity and effectiveness in this context.

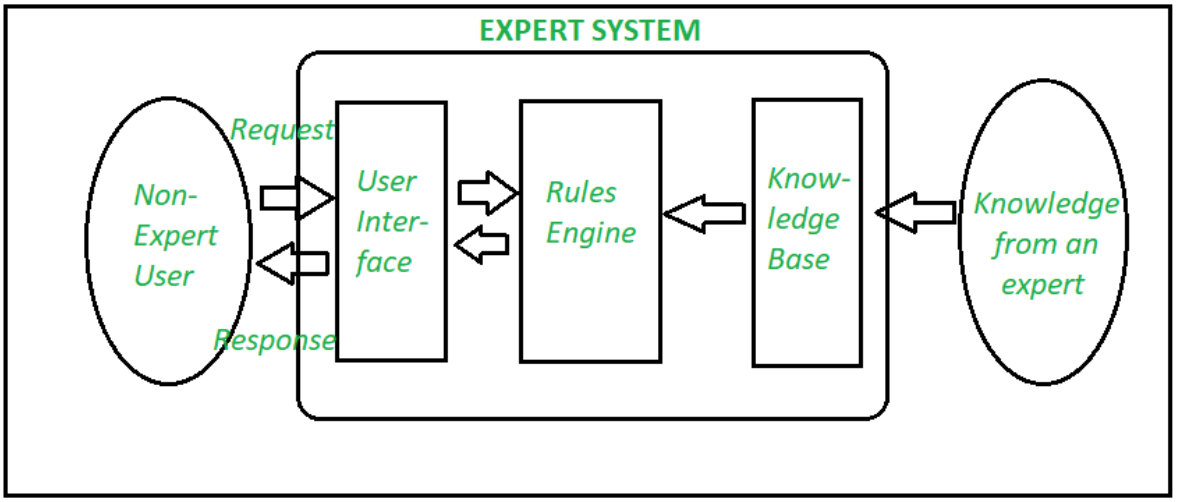
7. Chatbot : a chatbot is a computer program designed to simulate conversation with human users, typically through text or speech interfaces. These programs use various techniques, such as natural language processing and machine learning, to understand user inputs and generate appropriate responses. Chatbots are used for tasks like customer service, information retrieval, entertainment, and more.

8. Expert systems (ESs) are computer programs that use artificial intelligence (AI) to mimic the decision-making of a human expert in a particular field. They are designed to solve complex problems by reasoning through knowledge, represented mainly as if–then rules rather than through conventional procedural code.

ESs have three main components:

- Knowledge base: Stores the information the expert system draws upon

- Inferencing procedure: Uses knowledge about its application domain to solve problems that would otherwise require human competence or expertise
- User interface: Provides a response to the patient



9. Symmetric Algorithm:

- **Definition:** Symmetric algorithms use the same key for both encryption and decryption. The sender and receiver must share this secret key in advance.
- **Operation:** Encrypting plaintext with the key produces ciphertext, and decrypting ciphertext with the same key yields the original plaintext.
- **Key Features:** Fast and efficient for large data volumes, but requires secure key distribution channels.
- **Examples:** DES, AES, 3DES.

10. Asymmetric Algorithm:

- **Definition:** Asymmetric algorithms use a pair of keys - public and private. These keys are mathematically related but cannot be derived from each other.
- **Operation:** Public key is used for encryption, while private key is used for decryption (or vice versa). Messages encrypted with one key can only be decrypted with the other key.
- **Key Features:** Eliminates the need for secure key distribution channels, but slower compared to symmetric algorithms.
- **Examples:** RSA, Diffie-Hellman, Elliptic Curve Cryptography (ECC).

In summary, symmetric algorithms use a single shared key for encryption and decryption, while asymmetric algorithms use a pair of keys, allowing for secure communication without the need for pre-shared keys.

11. Encryption is the process of converting information or data into a code, often using algorithms, to make it unreadable to anyone except those with authorised access. It helps protect sensitive information during

transmission or storage by scrambling it into a format that can only be understood by someone with the correct decryption key or algorithm.

Decryption, on the other hand, is the reverse process of encryption. It involves converting the encrypted data back into its original, readable form using a decryption key or algorithm. This allows authorised users to access and understand the information that was previously encrypted.

12. **DES (Data Encryption Standard)** is a symmetric-key block cipher algorithm used for encryption and decryption of electronic data. DES has been largely replaced by more secure algorithms such as AES (Advanced Encryption Standard).

- **Key Features:**
  - Symmetric-key algorithm: Same key is used for both encryption and decryption.
  - Block cipher: Encrypts data in fixed-size blocks (64 bits).
  - Iterative process: Uses multiple rounds of substitution, permutation, and key mixing.
- **Algorithm Steps:**
  - Key Generation: Generates 56-bit subkeys from a 64-bit initial key.
  - Initial Permutation (IP): Rearranges input bits according to a fixed permutation table.
  - Rounds: Multiple rounds (usually 16) of substitution (S-box), permutation (P-box), and key mixing (XOR).
  - Final Permutation (FP): Reverses the initial permutation to produce the ciphertext.
- **Strengths:**
  - Historically significant: One of the first widely used encryption algorithms.
  - Efficient implementation: Fast execution on hardware and software platforms.
- **Weaknesses:**
  - Small key size: 56-bit key length makes it vulnerable to brute-force attacks.
  - Inadequate for modern security requirements: Vulnerable to advanced cryptanalysis techniques.

13. **RSA algorithm** is an asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e. Public Key and Private Key. As the name describes, the Public Key is given to everyone and the Private key is kept private.

An example of asymmetric cryptography:

1. A client (for example browser) sends its public key to the server and requests some data.
2. The server encrypts the data using the client's public key and sends the encrypted data.
3. The client receives this data and decrypts it.

Key Generation:

1. Choose two large prime numbers p and q
2. Compute their product, $n=p\times q$, known as the modulus.
3. Compute Euler's totient function, $\phi(n)=(p-1)\times(q-1)$.
4. Choose an integer e such that $1<e<\phi(n)$ and $\gcd(e,\phi(n))=1$, where gcd is the greatest common divisor.
5. Compute the modular multiplicative inverse of e modulo $\phi(n)$, denoted as d.
6. Public key is (n,e) & private key is (n,d).

Encryption:
Sender encrypts plaintext message M into ciphertext C using recipient's public key (n,e):
$C \equiv M^e \bmod n$

Decryption:
Recipient decrypts ciphertext C into plaintext M using private key(n,d):
$M \equiv C^d \bmod n$

Security:
Security relies on the difficulty of factoring large integers.
RSA is secure when key sizes are sufficiently large (e.g., 2048 bits or more).

Applications:
Secure data transmission: Encrypting sensitive information for secure communication.
Digital signatures: Signing messages to ensure authenticity and integrity.
Key exchange: Used in protocols like SSL/TLS for secure communication over the internet.
RSA remains one of the most widely used and trusted encryption algorithms in practice.

14. **The Diffie-Hellman key exchange algorithm** is a method for securely exchanging cryptographic keys over a public channel.

Key Features:

1. Key exchange algorithm: Allows two parties to agree on a shared secret key over an insecure communication channel.
2. Public-key cryptography: No prior secret key is required between the parties.
3. Based on the difficulty of the discrete logarithm problem.