1. Differentiate between informed and uninformed search strategies. Give one example of each.

Uninformed Search
Knowledge used : Does not use any problem-specific knowledge.
Goal awareness Only knows the goal state condition.
Efficiency:Slower; explores more nodes.
Example: Breadth-First Search (BFS)
* **Uninformed:** BFS explores all nodes level by level until it finds the goal.

Aspect Informed Search
Knowledge used : Uses heuristic (extra) knowledge to guide the search.
Goal awareness Estimates how close a state is to the goal.Only knows the goal state condition.
Efficiency:Faster; explores fewer nodes
Example: A* or Greedy Best-First Search

* **Informed:** A* uses a heuristic function `f(n) = g(n) + h(n)` to find the shortest path efficiently.

---

2. Explain the Breadth-First Search (BFS) algorithm. Provide its time and space complexity and discuss its advantages and disadvantages.

Algorithm Steps:

1. Start from the initial (root) node.
2. Visit all neighboring nodes (level 1).

4. Stop when the goal node is found.

Pseudocode:

```
BFS(start, goal):
    create an empty queue
    enqueue(start)
    mark start as visited
    while queue not empty:
        node = dequeue()
        if node == goal:
            return success
        for each child of node:
            if child not visited:
                enqueue(child)
                mark visited
```

Complexity:

* **Time:** $O(b^d)$
* **Space:** $O(b^d)$

where `b` = branching factor, `d` = depth of the shallowest goal.

Advantages:

* Always finds the shortest path (if cost = 1 for each step).
* Systematic and complete.

Disadvantages:

* Requires large memory (space heavy).
* Slow for deep or infinite search spaces.

---

3. Discuss the limitations of a standard DFS and explain how the Depth-Limited Search helps.

Limitations of DFS:

* May go infinitely deep in infinite search spaces.
* Not guaranteed to find the shallowest (shortest) solution.
* Can get stuck in cycles if not handled.

Depth-Limited Search (DLS):

* Adds a **limit (L)** to the depth DFS can explore.
* Prevents infinite descent by cutting off nodes beyond depth *L*.

Advantages:

* Reduces infinite loop problem.
* Uses less memory than BFS.

Disadvantages:

* If *L* is too small, the goal might not be found.
* If *L* is too large, becomes inefficient like DFS.

---

4. What is a heuristic function? Why are heuristic searches more efficient than blind searches? Describe the A* search algorithm.

Heuristic Function (h(n)):
A function that estimates the cost (distance) from the current node *n* to the goal.
It provides "educated guesses" to guide the search.

Why more efficient:

* It focuses the search toward promising paths.
* Reduces the number of nodes explored compared to uninformed searches.

A* Search Algorithm:
Uses:

```
f(n) = g(n) + h(n)
```

where

* `g(n)` = cost from start to current node
* `h(n)` = estimated cost to goal

Algorithm Steps:

1. Start with the initial node.
2. Choose the node with the lowest `f(n)` value.
3. Expand it and calculate `f(n)` for its neighbors.
4. Repeat until the goal node is reached.

Properties:

* **Complete** (if branching factor finite)
* **Optimal** (if heuristic is admissible)

---

5. Explain the Minimax algorithm for game playing. Using a sample game tree, trace the algorithm to show how it finds the optimal move for the maximizing player.

Purpose:
Used in **two-player games** (like chess, tic-tac-toe) to choose the optimal move assuming both players play optimally.

Players:

* **MAX:** Tries to maximize the score.
* **MIN:** Tries to minimize the score.

Algorithm Steps:

1. Generate the game tree up to a certain depth.
2. Apply an evaluation function to the leaf nodes.
3. Propagate scores upward:

* MAX node → choose the **maximum** value from children.
* MIN node → choose the **minimum** value from children.
4. The move with the highest final value at the root is chosen by MAX.

Example:

```
MAX
/ | \
3 5 2
```

MAX chooses **5** because it's the highest – that's the optimal move.

Advantages:

* Finds the best possible move.
* Forms the base for Alpha-Beta Pruning (optimized version).

Would you like me to make these answers **formatted for Word submission** (e.g., numbered Q&A style with clean layout, ready to paste)?