Artificial Intelligence Practical File

Experiment 1: Implement Tic-Tac-Toe using A* Algorithm

Aim: To implement Tic-Tac-Toe game using A* search strategy.

Theory: A* is a heuristic-driven search algorithm combining cost and heuristic value.

Algorithm:
1. Initialize board
2. Generate successors
3. Apply heuristic
4. Select optimal move

Conclusion: Tic-Tac-Toe solved effectively using A* algorithm.

Experiment 2: 3 Missionaries and 3 Cannibals Problem using A*

Aim: To solve missionaries and cannibals state-space problem using A*.

Theory: A* finds optimal path by expanding least-cost nodes.

Algorithm:
1. Define state
2. Generate valid moves
3. Apply heuristic
4. Reach goal safely

Conclusion: A* ensures safe and optimal solution path.

Experiment 3: Solve 8-Puzzle Problem using A*

Aim: To solve the 8-puzzle sliding game using A* heuristic search.

Theory: Heuristics like misplaced tiles or Manhattan distance guide the search.

Algorithm:
1. Input puzzle
2. Expand lowest f(n)
3. Move blank tile
4. Reach goal configuration

Conclusion: Puzzle solved using optimal heuristic-driven path.

Experiment 4: Implement Expert System

Aim: To develop a simple rule-based expert system.

Theory: Expert systems use IF-THEN rules to mimic decision-making.

Algorithm:
1. Define knowledge base
2. Input facts
3. Apply inference
4. Generate output

Conclusion: Expert system successfully demonstrates rule-based reasoning.

Experiment 5: Implement Alpha-Beta Pruning

Aim: To optimize minimax search using Alpha-Beta pruning.

Theory: Pruning reduces search space by eliminating unnecessary branches.

Algorithm:
1. Apply minimax
2. Track α and β
3. Prune branches
4. Output best move

Conclusion: Alpha-Beta pruning improves efficiency significantly.

Experiment 6: Develop Elementary Chatbot

Aim: To design a simple pattern-based conversational chatbot.

Theory: Chatbots use pattern matching and scripted responses.

Algorithm:
1. Define patterns
2. Accept input
3. Match rule
4. Produce response

Conclusion: Chatbot interacts successfully with the user.

Experiment 7: Goal-Stack Planning for Blocks World Problem

Aim: To implement goal-stack-based planning for block manipulation.

Theory: Planner decomposes complex goal into subgoals using stack operations.

Algorithm:
1. Initialize stack
2. Push goals
3. Resolve subgoals
4. Achieve final state

Conclusion: Goal-stack planning efficiently organizes block operations.

Experiment 8: Hill Climbing using Heuristic Search

Aim: To implement hill climbing optimization technique.

Theory: Hill climbing chooses best immediate neighbor using heuristic.

Algorithm:
1. Start at initial state
2. Evaluate neighbors
3. Move to best one
4. Stop at local optimum

Conclusion: Hill climbing reaches a good (not always global) solution.