TOP 30 MOST IMPORTANT QUESTIONS

Artificial Intelligence - Mumbai University

Subject Code: 48893 | Semester V

Based on comprehensive analysis of 6 previous year question papers (2022-2025), here are the **30 most important questions** with high probability of appearing in your exam:

SECTION A: HIGH-FREQUENCY QUESTIONS (10 Marks)

1. AI Agents (Types & Architecture) *(Appeared 9 times)*

Explain different types of AI agents with neat diagrams:

- Simple Reflex Agent
- Model-based Reflex Agent
- Goal-based Agent
- Utility-based Agent
- Learning Agent

Compare any two agents (Utility based vs Goal based, Model based vs Simple Reflex). Explain components: Agent function, Agent program, Sensors, Actuators, Environment.

2. PEAS Descriptor *(Appeared 6 times)*

What is PEAS descriptor? Give detailed PEAS descriptor for following systems:

- **Performance Measure**: What constitutes success
- **Environment**: The world the agent operates in
- **Actuators**: Actions the agent can perform
- **Sensors**: Perceptions the agent receives

Common examples asked:

- Medical Diagnosis System
- Movie Ticket Booking System
- Online English Tutor
- Robot Maid for cleaning
- Autonomous Vehicle
- Shopping for AI books online

3. Hill Climbing Algorithm *(Appeared 6 times)*

Explain Hill Climbing algorithm with example. Discuss its inherent limitations:

- **Local Maxima**: Peak that isn't global maximum
- **Plateau**: Flat area where all neighbors have same value
- **Ridges**: Direction of steepest ascent doesn't lead to top

Solutions to overcome limitations:

- Random restart
- Simulated Annealing
- Stochastic Hill Climbing
- First-choice Hill Climbing

Compare Hill Climbing with Simulated Annealing.

4. Genetic Algorithm *(Appeared 5 times)*

Define key terms:

- **Chromosome**: String of genes representing solution
- **Gene**: Single element of chromosome
- **Population**: Set of chromosomes
- **Fitness Function**: Measures quality of solution
- **Selection**: Choosing parents based on fitness
- **Crossover**: Combining two parents to create offspring
- **Mutation**: Random changes to maintain diversity

Algorithm Steps:

1. Initialize random population
2. Calculate fitness for each individual
3. Select parents using selection methods (Roulette Wheel, Tournament)
4. Apply crossover operator
5. Apply mutation operator
6. Replace old population
7. Repeat until termination condition

Example: Solving optimization problems, String matching, N-Queens problem

5. Forward & Backward Chaining *(Appeared 5 times)*

Forward Chaining (Data-driven):

- Start with known facts
- Apply rules to infer new facts
- Continue until goal is reached or no more rules apply

Backward Chaining (Goal-driven):

- Start with goal
- Find rules that conclude goal
- Make premises of rules as new sub-goals
- Continue until facts are reached

Classic Problem: "Robert is criminal"

- Given: It's crime to sell weapons to hostile nations
- Country A is hostile, has missiles
- Robert (American) sold missiles to Country A
- Prove: Robert is criminal

Show proof using both techniques with derivation trees.

6. Environment Types *(Appeared 5 times)*

Classify environments on six dimensions:

Dimension	Types	Example
Observability	Fully/Partially Observable	Chess (Fully), Poker (Partial)
Determinism	Deterministic/Stochastic	Crossword (Det), Dice games (Stoch)
Episodic	Episodic/Sequential	Spam filter (Episodic), Chess (Sequential)

Dimension	Types	Example
Dynamic	Static/Dynamic	Crossword (Static), Self-driving (Dynamic)
Discrete	Discrete/Continuous	Chess (Discrete), Taxi driving (Continuous)
Agents	Single/Multi-agent	Sudoku (Single), Soccer (Multi)

Environment Analysis for:

- 8-Queen Problem: Fully observable, Deterministic, Sequential, Static, Discrete, Single-agent
- Tic-Tac-Toe: Fully observable, Deterministic, Sequential, Static, Discrete, Multi-agent
- Pacman Game: Partially observable, Stochastic, Sequential, Dynamic, Discrete, Multi-agent

7. Expert System *(Appeared 4 times)*

Architecture Components:

1. **Knowledge Base**: Domain knowledge in form of rules (IF-THEN)
2. **Inference Engine**: Reasoning mechanism (Forward/Backward chaining)
3. **Working Memory**: Current facts and data
4. **User Interface**: Communication with user
5. **Explanation Facility**: Explains reasoning process
6. **Knowledge Acquisition Facility**: Updates knowledge base

Real-life Examples:

- MYCIN: Medical diagnosis
- DENDRAL: Chemical analysis
- PROSPECTOR: Mineral exploration
- XCON: Computer configuration

8. Bayesian Belief Network *(Appeared 4 times)*

Definition: Directed Acyclic Graph (DAG) representing probabilistic relationships among variables.

Components:

- **Nodes**: Random variables
- **Edges**: Direct probabilistic dependencies
- **CPT (Conditional Probability Tables)**: Probability of node given parents

Construction Steps:

1. Identify relevant variables
2. Order variables
3. Add nodes in order
4. Add edges for direct influences
5. Define CPT for each node

Example: Medical diagnosis network with symptoms, diseases, and test results.

9. Alpha-Beta Pruning *(Appeared 4 times)*

Enhancement of MinMax algorithm that reduces number of nodes evaluated.

Key Concepts:

- **Alpha (α)**: Best value for MAX found so far
- **Beta (β)**: Best value for MIN found so far
- **Pruning Condition**: If $\beta \leq \alpha$, prune remaining branches

Algorithm:

1. Initialize α = -∞, β = +∞
2. At MAX node: Update α, prune if α ≥ β
3. At MIN node: Update β, prune if β ≤ α

Advantage: Can reduce search space from $O(b^d)$ to $O(b^{d/2})$ with optimal ordering.

Example: Apply on game tree with given values, show pruned branches.

10. Prolog Programming *(Appeared 4 times)*

Write Prolog programs for:

a) Factorial:

```
factorial(0, 1).
factorial(N, F) :-
    N &gt; 0,
    N1 is N - 1,
    factorial(N1, F1),
    F is N * F1.
```

b) Fibonacci:

```
fib(0, 0).
fib(1, 1).
fib(N, F) :-
    N &gt; 1,
    N1 is N - 1,
    N2 is N - 2,
    fib(N1, F1),
    fib(N2, F2),
    F is F1 + F2.
```

c) Family Tree:

```
parent(john, mary).
parent(john, tom).
father(X, Y) :- parent(X, Y), male(X).
mother(X, Y) :- parent(X, Y), female(X).
sibling(X, Y) :- parent(Z, X), parent(Z, Y), X \\= Y.
```

11. Bayes Theorem *(Appeared 4 times)*

Formula:

$$P(A|B) = fracP(B|A)cdotP(A)P(B)$$

Where:

- $P(A|B)$: Posterior probability
- $P(B|A)$: Likelihood
- $P(A)$: Prior probability
- $P(B)$: Evidence

Classic Problem:
"80% students like English, 30% like both English and Math. What percentage of students who like Math also like English?"

Solution:

- P(E) = 0.80 (like English)
- P(E ∩ M) = 0.30 (like both)
- Find: P(E|M) = P(E ∩ M) / P(M)
- Using conditional probability

12. Iterative Deepening Search *(Appeared 4 times)*

Algorithm: Combines benefits of BFS (completeness, optimality) and DFS (space efficiency).

Working:

- Perform DFS with depth limit 0
- If goal not found, increment depth limit
- Repeat until goal found

Performance Measure:

Criterion	IDS
Time Complexity	$O(b^d)$
Space Complexity	$O(bd)$
Complete	Yes (for finite branching factor)
Optimal	Yes (when path cost is non-decreasing)

Advantages:

- Memory efficient like DFS
- Complete and optimal like BFS
- Better than DFS for large search spaces

Compare with Depth Limited Search (DLS).

13. First Order Predicate Logic (FOPL) *(Appeared 4 times)*

Convert English sentences to FOPL:

Examples:

1. **"Everyone likes everyone"**
   $forallxforallytextLikes(x, y)$
2. **"All graduates are unemployed"**
   $forallx(textGraduate(x)rightarrowtextUnemployed(x))$
3. **"Every dolphin is mammal"**
   $forallx(textDolphin(x)rightarrowtextMammal(x))$
4. **"No purple mushroom is poisonous"**
   $forallx((textPurple(x)landtextMushroom(x))rightarrownegtextPoisonous(x))$
5. **"Every person who buys insurance is smart"**
   $forallx((textPerson(x)landtextBuys(x, textInsurance))rightarrowtextSmart(x))$
6. **"At least one student failed History"**
   $existsx(textStudent(x)landtextFailed(x, textHistory))$

14. A Algorithm (Appeared 3 times)*

Evaluation Function:

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

Where:

- $g(n)$: Cost from start to node n
- $h(n)$: Heuristic estimate from n to goal
- $f(n)$: Total estimated cost

Algorithm:

1. Initialize: OPEN = {start}, CLOSED = {}
2. Select node with minimum f(n) from OPEN
3. If goal, return path

4. Expand node, generate children
5. For each child: Calculate f(n), add to OPEN
6. Move current node to CLOSED
7. Repeat

Properties:

- **Complete**: Yes (with finite branching)
- **Optimal**: Yes (if h(n) is admissible)
- **Admissible heuristic**: $h(n)leqh^*(n)$ (never overestimates)

Example: 8-puzzle, pathfinding

15. Resolution Theorem Proving *(Appeared 3 times)*

Steps:

1. Convert statements to CNF (Conjunctive Normal Form)
2. Negate the goal
3. Add to knowledge base
4. Apply resolution rule repeatedly
5. If derive empty clause ($\perp$), goal is proved

Classic Example:

- All graduating people are happy
- All happy people smile
- Someone is graduating
- **Prove**: Someone is smiling

Resolution Rule:

$$fracPlorQ, negPlorRQlorR$$

Draw resolution tree showing each step.

16. Types of Learning in AI *(Appeared 3 times)*

1. Supervised Learning:

- Learning from labeled examples
- Examples: Classification, Regression
- Algorithms: Decision Trees, SVM, Neural Networks

2. Unsupervised Learning:

- Learning from unlabeled data
- Examples: Clustering, Dimensionality reduction
- Algorithms: K-means, PCA, Hierarchical clustering

3. Reinforcement Learning:

- Learning through interaction with environment
- Agent receives rewards/penalties
- Components: State, Action, Reward, Policy
- Examples: Game playing, Robotics

4. Semi-supervised Learning:

- Uses both labeled and unlabeled data
- Useful when labeling is expensive

17. Partial Order Planning *(Appeared 3 times)*

Characteristics:

- Plans are partially ordered sets of actions
- Allows parallel execution of independent actions
- More flexible than total order planning

Algorithm (POP - Partial Order Planner):

1. Start with initial plan: Start → Goal
2. Select open precondition
3. Choose action to achieve it (from existing or add new)
4. Add ordering constraints
5. Add causal links
6. Resolve conflicts
7. Repeat until no open preconditions

Example: Blocks world, Robot navigation

Advantages:

- Least commitment strategy
- Efficient for complex problems
- Allows plan reuse

18. Total Order Planning *(Appeared 3 times)*

Characteristics:

- Actions are totally ordered (linear sequence)
- Simpler to implement
- Easy to execute

Algorithm (Simple Forward Planning):

1. Start with initial state
2. Choose action applicable in current state
3. Apply action, get new state
4. Repeat until goal achieved

Example: STRIPS planner

Comparison with Partial Order:

Feature	Total Order	Partial Order
Flexibility	Low	High
Parallelism	No	Yes
Complexity	Simple	Complex
Efficiency	Less	More

19. AI Applications *(Appeared 3 times)*

Major Application Areas:

1. **Healthcare:**
    - Disease diagnosis
    - Drug discovery

    - Medical imaging analysis
    - Personalized treatment
2. **Education:**
    - Intelligent tutoring systems
    - Automated grading
    - Personalized learning
3. **Finance:**
    - Fraud detection
    - Algorithmic trading
    - Risk assessment
    - Credit scoring
4. **Robotics:**
    - Industrial automation
    - Autonomous vehicles
    - Warehouse management
5. **Natural Language Processing:**
    - Machine translation
    - Chatbots
    - Sentiment analysis
6. **Computer Vision:**
    - Face recognition
    - Object detection
    - Medical image analysis

20. Intelligent Agent *(Appeared 3 times)*

Definition: An agent that perceives its environment through sensors and acts upon it through actuators to achieve goals.

Components:

- **Agent Function**: Maps percept sequences to actions
- **Agent Program**: Concrete implementation
- **Sensors**: Receive perceptions
- **Actuators**: Perform actions
- **Environment**: Where agent operates

Properties of Ideal Intelligent Agent:

- **Autonomous**: Operates independently
- **Reactive**: Responds to environment
- **Proactive**: Takes initiative
- **Social**: Interacts with other agents

Rationality: Agent should choose action that maximizes expected performance measure.

SECTION B: MEDIUM-FREQUENCY QUESTIONS

21. 8-Puzzle Problem *(Appeared 2 times)*

Problem Formulation:

- **Initial State**: Given configuration
- **Actions**: Move blank up, down, left, right
- **Successor Function**: Result of applying action
- **Goal Test**: Check if matches goal configuration

- **Path Cost**: Number of moves

Heuristic Functions:

- **h_1**: Number of misplaced tiles
- **h_2**: Manhattan distance (sum of distances)

Admissibility: Both h_1 and h_2 are admissible (never overestimate).

Solve using A* algorithm with detailed steps.

22. Missionaries & Cannibals Problem *(Appeared 2 times)*

Problem:

- 3 missionaries, 3 cannibals on left bank
- Boat holds 2 people
- Cannibals can't outnumber missionaries on either bank
- Get all to right bank

State Representation: (M_left, C_left, Boat_position)

Operators:

- Move 2 missionaries
- Move 2 cannibals
- Move 1 missionary, 1 cannibal
- Move 1 missionary
- Move 1 cannibal

Draw state space graph with only legal states.

Best Algorithm: BFS (finds shortest solution)

23. AI Perspectives *(Appeared 2 times)*

Four Approaches:

1. **Thinking Humanly:**
    - Cognitive modeling
    - Think like humans
    - Example: GPS (General Problem Solver)
2. **Acting Humanly:**
    - Turing Test approach
    - Behave like humans
    - Requires NLP, Knowledge representation, Reasoning, Learning
3. **Thinking Rationally:**
    - Laws of thought
    - Logical reasoning
    - Example: Logic-based AI
4. **Acting Rationally:**
    - Rational agent approach
    - Do the right thing
    - Maximize expected utility
    - **Modern AI focus**

24. Simulated Annealing *(Appeared 2 times)*

Concept: Inspired by metallurgical annealing process.

Algorithm:

1. Start with random solution
2. Generate random neighbor
3. Calculate ΔE = E(new) - E(current)
4. If ΔE < 0: Accept (better solution)
5. If ΔE > 0: Accept with probability $e^{-DeltaE/T}$
6. Decrease temperature
7. Repeat

Advantages over Hill Climbing:

- Can escape local maxima
- Accepts worse solutions probabilistically
- Probability decreases with temperature

Temperature Schedule:

- Start: High temperature (more exploration)
- End: Low temperature (more exploitation)

25. Problem Formulation *(Appeared 2 times)*

Components:

1. **Initial State**: Starting configuration
2. **Actions**: Possible moves
3. **Transition Model**: Result of action
4. **Goal Test**: Check if goal reached
5. **Path Cost**: Cost of sequence of actions

Well-defined Problem:

- Clear initial state
- Well-defined actions
- Explicit goal
- Measurable cost

Examples:

- 8-puzzle
- Missionaries and Cannibals
- Traveling Salesman
- N-Queens

26-30. Additional Important Topics

26. Conditional Probability - Solve numerical problems

27. CNF Conversion - Convert propositional logic to CNF

28. MinMax Search - Game playing algorithm

29. Knowledge Representation - Semantic networks, Frames, Rules

30. Search Algorithm Comparison - BFS, DFS, IDS, Bidirectional

IMPORTANT FORMULAS

Bayes Theorem:

$$P(A|B) = fracP(B|A)cdotP(A)P(B)$$

A Evaluation:*

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

Alpha-Beta Pruning Condition:

$$textPruneifbetaleqalpha$$

Manhattan Distance (8-puzzle):

$$h(n) = sum|x_i - x_{goal}| + |y_i - y_{goal}|$$

STUDY STRATEGY

Priority 1 (Must Know): Questions 1-10 - Highest frequency
Priority 2 (Very Important): Questions 11-20 - Medium-high frequency
Priority 3 (Important): Questions 21-30 - Moderate frequency

Focus Areas:

1. AI Agents and PEAS (Q1-2)
2. Search Algorithms (Q3, 14, 12, 30)
3. Knowledge & Logic (Q5, 13, 15, 27, 29)
4. Optimization (Q4, 9, 24)
5. Probability (Q8, 11, 26, 31)
6. Planning (Q17-18)
7. Learning (Q16)

Best of luck for your AI exam!