AI-ML MODULE-1

(1) Define AI & list the task domains of AI

(A) Def: Artificial Intelligence is the study of how to make computers do things, which at the moment people do better.

Domains of AI

Mundane (Ordinary Tasks)	Formal Tasks	Expert Tasks
Perception • computer vision • speech, voice.	Formal Task • Math • Geometry • Logic • Integration & Differentiation	• Engineering • Fault Finding • Manufacturing • Monitoring.
Natural Language Processing • understanding • Language generation • Language Translation	Games • chess • checkers	Scientific Analysis
common sense Reasoning Planning Robotics • Locomotive	Verification Theorem Proving	Financial analysis Medical Diagnosis Creativity.

② State and explain algorithm of Best First Search Algorithm with example.

Algorithm:

BestFirstSearch (Graph g, Node start)

i) Create an empty PriorityQueue
   PriorityQueue pq;

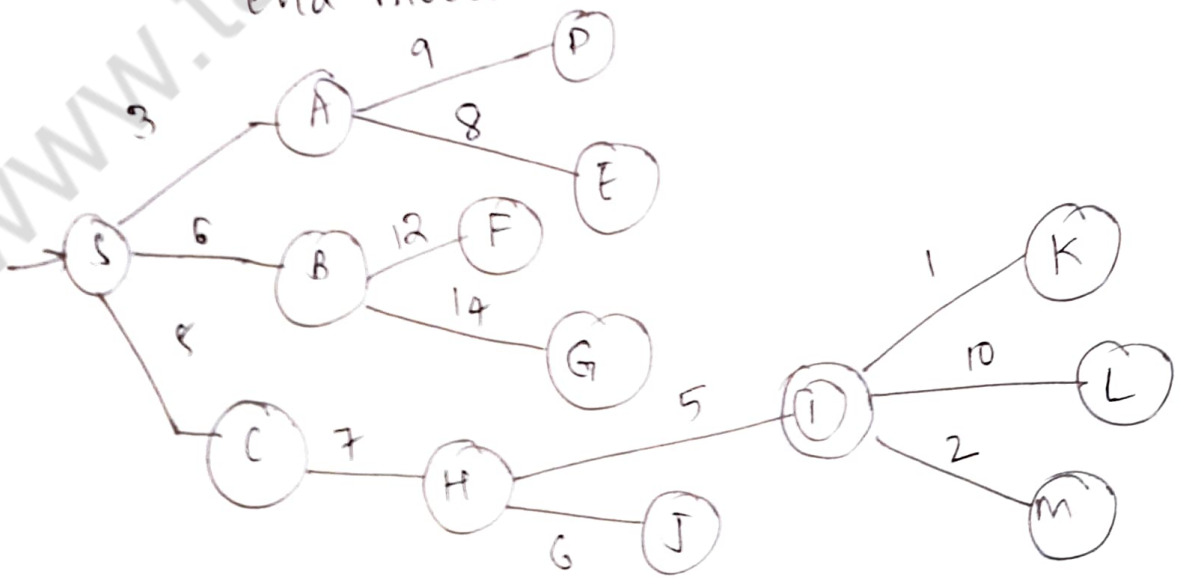
ii) Insert "start" in pq
   pq.insert(start)

iii) Until PriorityQueue is empty
   u = PriorityQueue.DeleteMin

   If u is the goal
      Exit

   Else
      Foreach neighbour v of u
         if v "unvisited"
            Mark v "visited"
            pq.insert(v)
      Mark v "Examined"

End Procedure



(3) Water Jug Problem.

The State Space = (four, three)

four = 0, 1, 2, 3 or 4

three = 0, 1, 2 or 3.

Start State = (0, 0)

goal State = (2, n)

n may be any number but is limited to 3

Production Rules:

	Goal
i) (four, three) if four < 4	fill 4
ii) (four, three) if three < 3	fill 3
iii) (four, three) if four > 0	empty 4
iv) (four, three) if three > 0	empty 3
v) (four, three) if four + three < 4	empty 3 to 4
vi) (four, three) if four + three < 3	empty 4 to 3
vii) (0, three) if three > 0	empty 3 to 4
viii) (four, 0) if four > 0	empty 4 to 3
ix) (0, 2)	empty 3 to 4
x) (2, 0)	empty 4 to 3
xi) (four, three) if four < 4	pour difference from 4 to 3
xii) (three, four) if three < 3	pour diff from 3 to 4

Solution:

Four Jug	Three Jug	Rules
0	0	—
0	3	2
3	0	7
3	3	2
4	2	11
0	2	3
2	0	10

(4) Simulated Annealing:

- It is a variation of hill climbing, at the beginning of the process, some downhill moves may be made.
- In the beginning some hill moves may be made. The idea is to do enough exploration of the whole space early on - so that the final solution in relatively insensitive to the ~~start~~ starting state.

By doing so we can lower the chances of getting caught at local maximum, plateau or a ridge.

- In this we attempt to minimize rather than maximize the value of the objective function.
- Thus this process is one of valley descending in which the object function is the energy level.

(5) Problem characteristics with respect to Heuristic search.

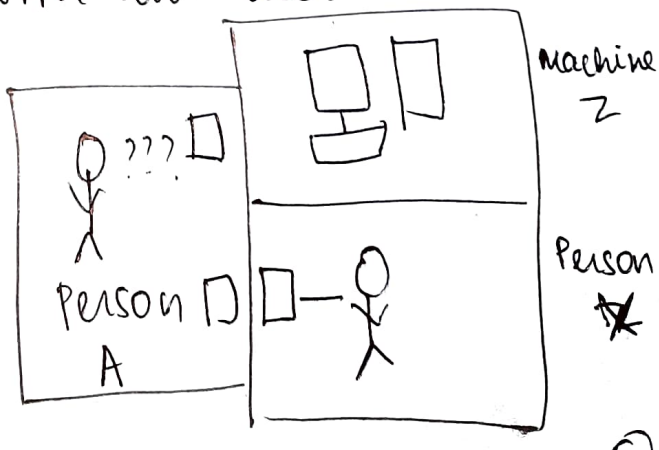
- Heuristics cannot be generalized
- It is domain specific
- Production systems provide ideal techniques for representation in the form of IF-THEN
- Some heuristics are used to define the control structure that guides the search process
- Heuristic search can be encoded in the rules to represent domain knowledge.

To use heuristic search for problem solving, the following considerations are applied

- Decomposability of problem into independent smaller subproblems
- Possibility of undoing soln steps.
- Predictability of problem
- Possibility of obtaining an obvious problem without comparision of all other solns possible
- Type of solution: whether it's a state or a path
- Role of knowledge in problem solving
- Nature of solution process: with or without interacting with the user.

(6) Turing Test

Alan Turing proposed this method to determ -ine whether a machine can think



To conduct the test, we need two people, and a machine to be evaluated.

Turing Test provides a definition of intelligence in a machine and compares the intelligence behaviour of human being with computer.

Person A plays the role of an interrogator, who is in a separate room from the computer and other person.

The interrogator can ask set of questions to both the computer Z and person X by typing questions & receiving typed responses. The interrogator knows them as Z and X and aims to determine who the person is and who the machine is

The aim of the machine is to fool the interrogator into believing that it is the person.

If the machine succeeds we conclude that machines can think. The machine is allowed to do whatever it can to fool the interrogator

The interrogator just gets two sets of answers does not know which set is from ~~or~~ human and which is from computer.

If interrogator cannot tell who is human and computer, then computer passed the test

(21) Heuristic Search For Travelling Salesman Problem:

State Space:

i) A set of states of problem.
ii) A set of operators to operate b/w states.
iii) start and goal state.

- The information of each state node is used to consider visiting a node or not.
- This information ~~of~~ is represented by heuristic function.
- Best first search can be used.

$g(x)$ → Distance from root node to x

$g(Y) = g(x) + d(x, y)$

$d(x, y)$ → distance b/w x & y.

Algorithm:

i) Append root node to VISIT
ii) If VISIT is empty, search fails, EXIT
iii) Get a node from the head of VISIT, calls N.
iv) If N is the goal state, then search succeeds output the result. Exit.
v) Push all child nodes of N to VISIT sort all elements x of VISIT in ascending order of $g(x)$
vi) Go to 2.

(8) Breadth First Search

- In this method no viable solutions are omitted and therefore it is guaranteed that an optimal solution is found.
- It is not feasible for search space that is large

Algorithm:

1) Create a variable called LIST & set it to be the start state.
2) Loop until a goal state is found or LIST is empty

Do,

a) Remove the first element from the list and call it E.

if LIST is empty, quit

b) For every path each rule can match the state E, Do,

i) Apply the rule to generate a new state.

ii) If the new state is a goal state, quit & return this state.

iii) Otherwise, add the new state to the end of the LIST.

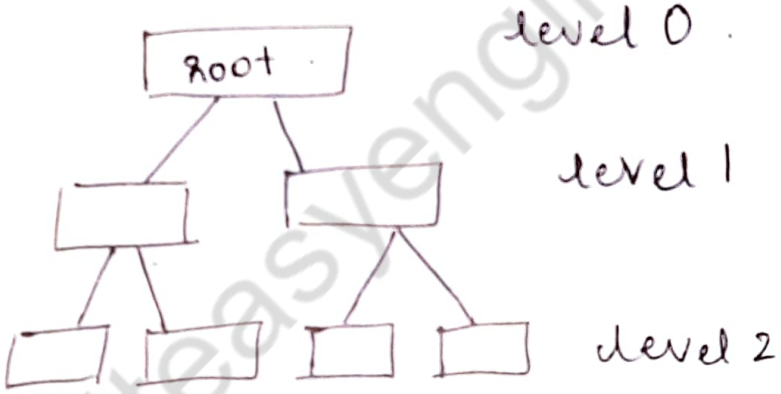
BFS starts at the highest layer, each layer is searched completely before moving to the next.

Advantages:

- Guaranteed optimal solution. (In the shortest amount of steps to reach goal)
- Can always find a goal node if one exists.

Disadvantages:

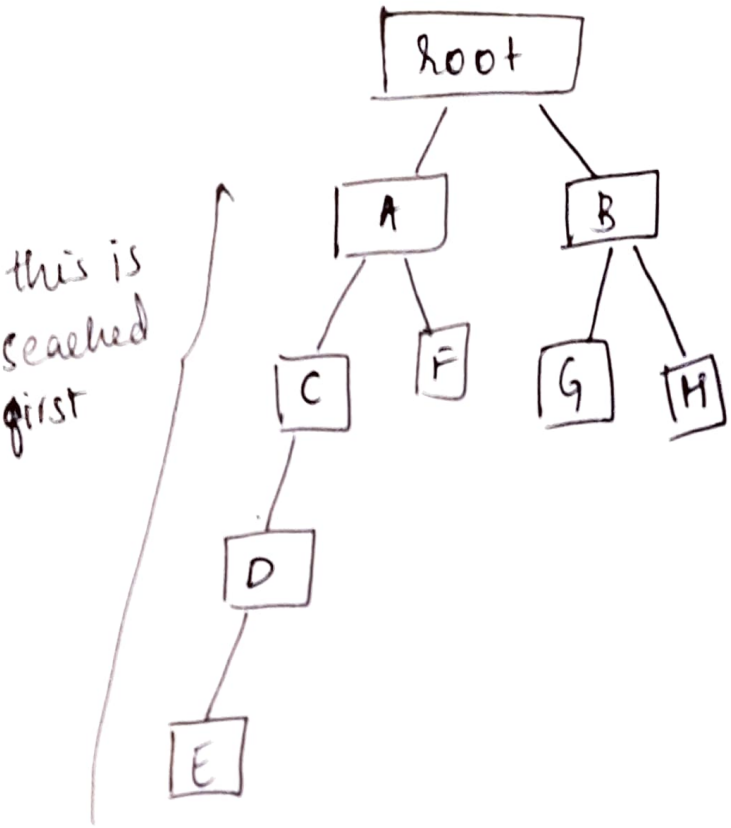
- High storage requirement that grows expenentially with the depth.



After the completion of level 1, level 2 is searched completely.

Depth First Search:

The search strategy is to extend the current path as far as possible before backtracking to the last choice point & try the next alternative path.



Algorithm:

- DFS applies operators to each newly generated state, trying to drive directly towards the goal

1.) If the starting state is goal state quit and return Success.

2.) Otherwise, Do,

a.) Generate a successor E to start state. If no more successor. signal failure.

b.) Call Depth-First-Search with E as start state.

c.) If success is returned. signal success else continue loop.

Advantages:

- Low storage requirements, linear with the depth.
- easy to program.

Disadvantages:

- May find a sub-optimal solution.
- Incomplete: without a bound on the depth, may not find a solution even if one exists.

(9) Production System:

They provide appropriate structures for performing and describing search processes.

It has 4 basic components:

- A set of rules each consisting of a left side that determines the applicability of the rule & right side that describes the operation to be performed.
- A database of current facts established during inference.
- A control strategy that specifies the order in which the rules will be compared with facts and how to resolve conflicts.

- A rule firing module.

Production Rules operate on knowledge DB.
each rule has a precondition, ie. either satisfied or not by knowledge DB.
If precondition is satisfied, rule can be applied.
Application of rule changes knowledge DB.
The control system chooses which applicable rule should be applied & ceases computation when a termination condition on the knowledge DB is satisfied.

Control Strategy.

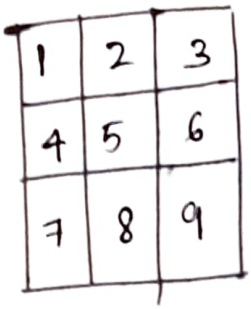
A strategy is defined by picking the order in which nodes expand.
The search strategies are evaluated along the following dimensions.

- completeness
- Time.
- complexity.
- Space complexity -
- Optimality.

Optimizes the cost of function with the hope of finding the optimal solution at the end.

(10) Tic-Tac-Toe Problem

i) The First Approach:



The tic-tac-toe game consists of a nine element vector called BOARD. It represents the numbers 1 to 9 in three rows.

An element contains the value 0 for Blank, 1 for X and 2 for O.

A MOVETABLE vector consists of 19683 (3^9) elements and is needed where each element is a nine element vector

Algorithm:

i) View the vector as a ternary number and convert it to a decimal number

ii) Use the decimal number as an index in MOVETABLE and access the vector

iii) Set BOARD to this vector indicating how the board looks after the move.

disadvantages:

- It takes more space and is very difficult to find the decimal number
- It is for this specific game and cannot be used for other games.

Hill Climbing:

Given a large set of inputs and a good heuristic function, it tries to find a sufficiently good solution to the problem.

This solution may not be the global optimal maximum

- Mathematical optimization problems implies the hill climbing solves the problems where we need to maximize or minimize a given real function by choosing values from the given inputs.
- Heuristic search means that this search algorithm may not find the optimal solution to the problem.
  However it finds a good solution in reasonable amount of time

Features

- Variant of generat & test algorithm
  ie;
  - generate a possible solution
  - test to see if this is expected
  - If the solution has been found quit or else goto 1
- Uses the Greedy Approach.
  At any point in state space, the search moves in that direction only which

Second approach:

2 is used for Blank
3 for X and 5 for O.
The variable called TURN indicates 1 for first move & 9 for the last.
The algorithm consists of three actions.

- MAKE2 which returns 5 if the center square is blank. Otherwise it returns any blank non-corner square 2, 4, 6, 8.
- POSSWIN(p) returns 0 if player p cannot win on the next move and otherwise returns the number of the square that gives a winning move.

It checks each line using products
$3 * 3 * 2 = 18$ gives a win for X.
$5 * 5 * 2 = 50$ gives a win for O.
The winning move is the holder of the blank.

- GO(n) makes a move to square n setting BOARD[n] to 3 or 5.

It is more involved and takes longer but is more efficient in storage which compensates for longer time.

Final Approach:

The structure of the data consists of BOARD which contains 9 element vectors, a

MODULE-2

Concept Learning & hypothesis space of Find-S

a list of board positions that could result from the next move and a number representing an estimation of how the board position leads to an ultimate win for the player to move.

This algorithm looks ahead to make a decision on the next move by deciding which the most promising move or the ~~position~~ most suitable move at any stage would be and select the same.

Considers all possible moves & replies that the program can make. Continue this process for as long as time permits until a winner emerges & then choose the move that leads to the computer program winning, if possible in the shortest time.