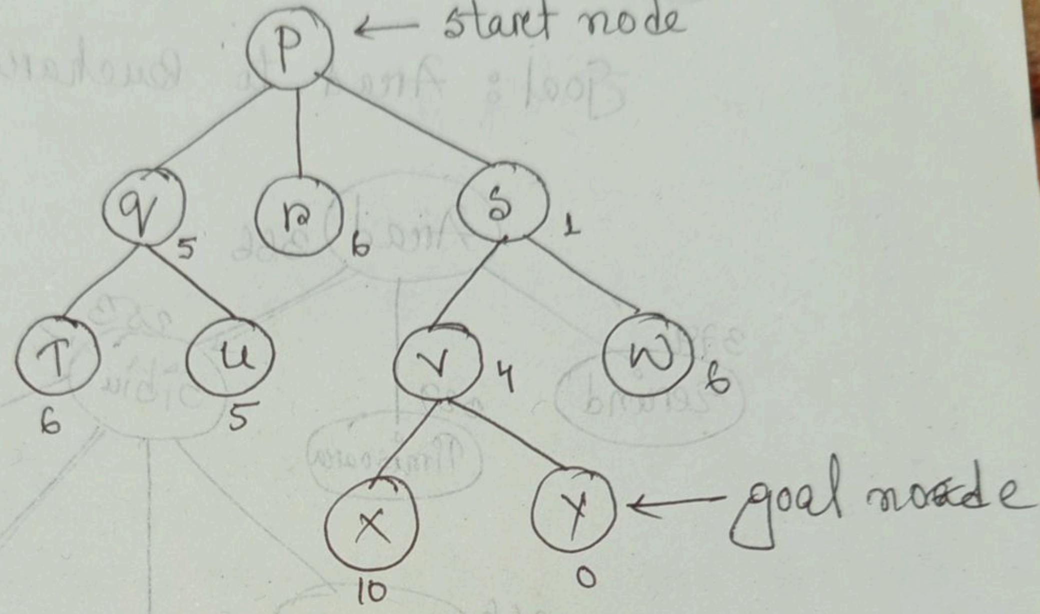
Best-First Search:



Open-list:

S q r

V q r W

Y q r W X

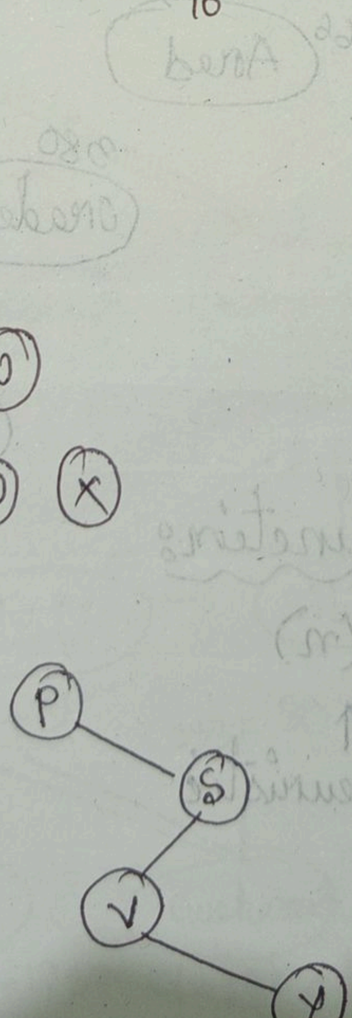
Close-list

P

P S

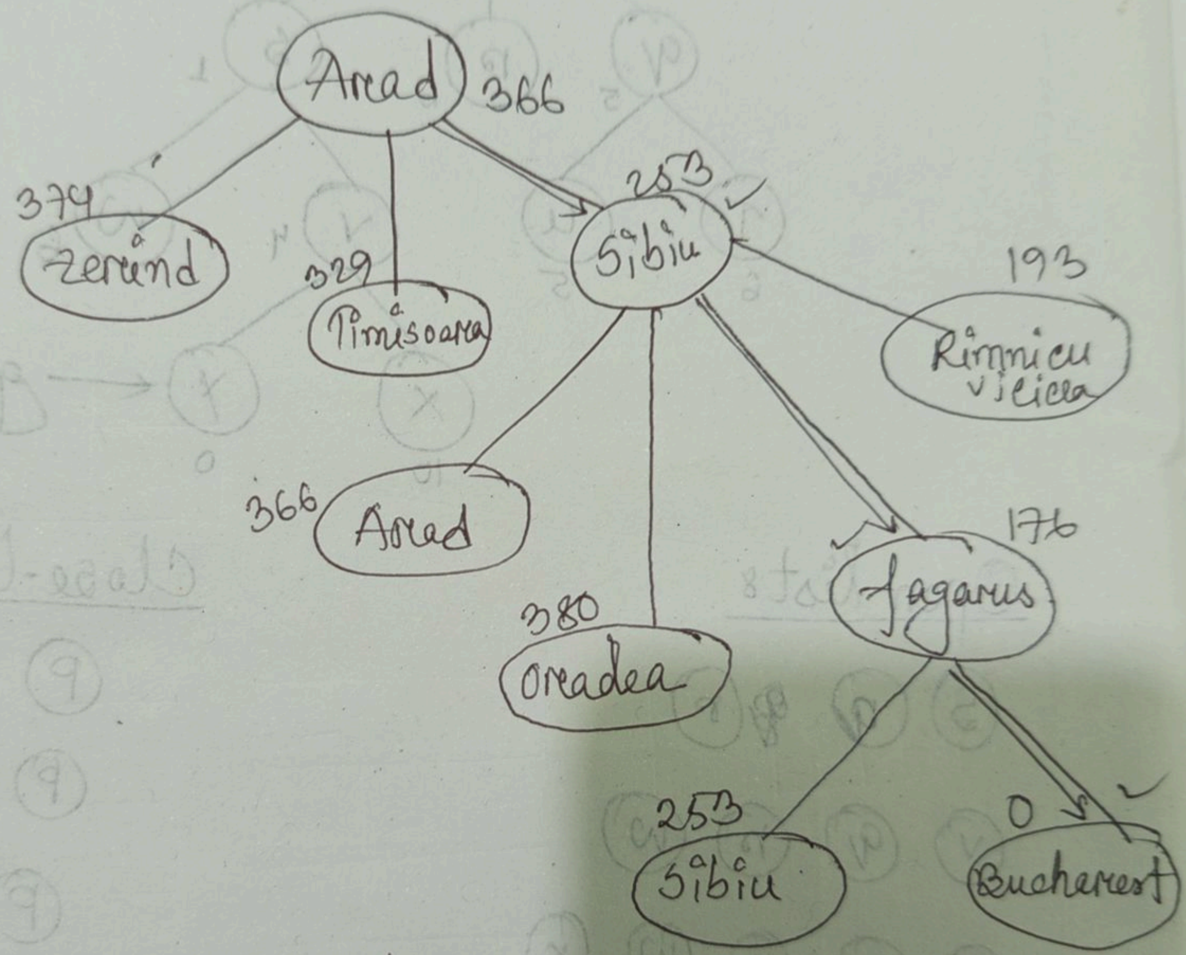
P S V

P S V Y



Greedy Best first Search:

Goal: Arad to Bucharest



Evaluation function:

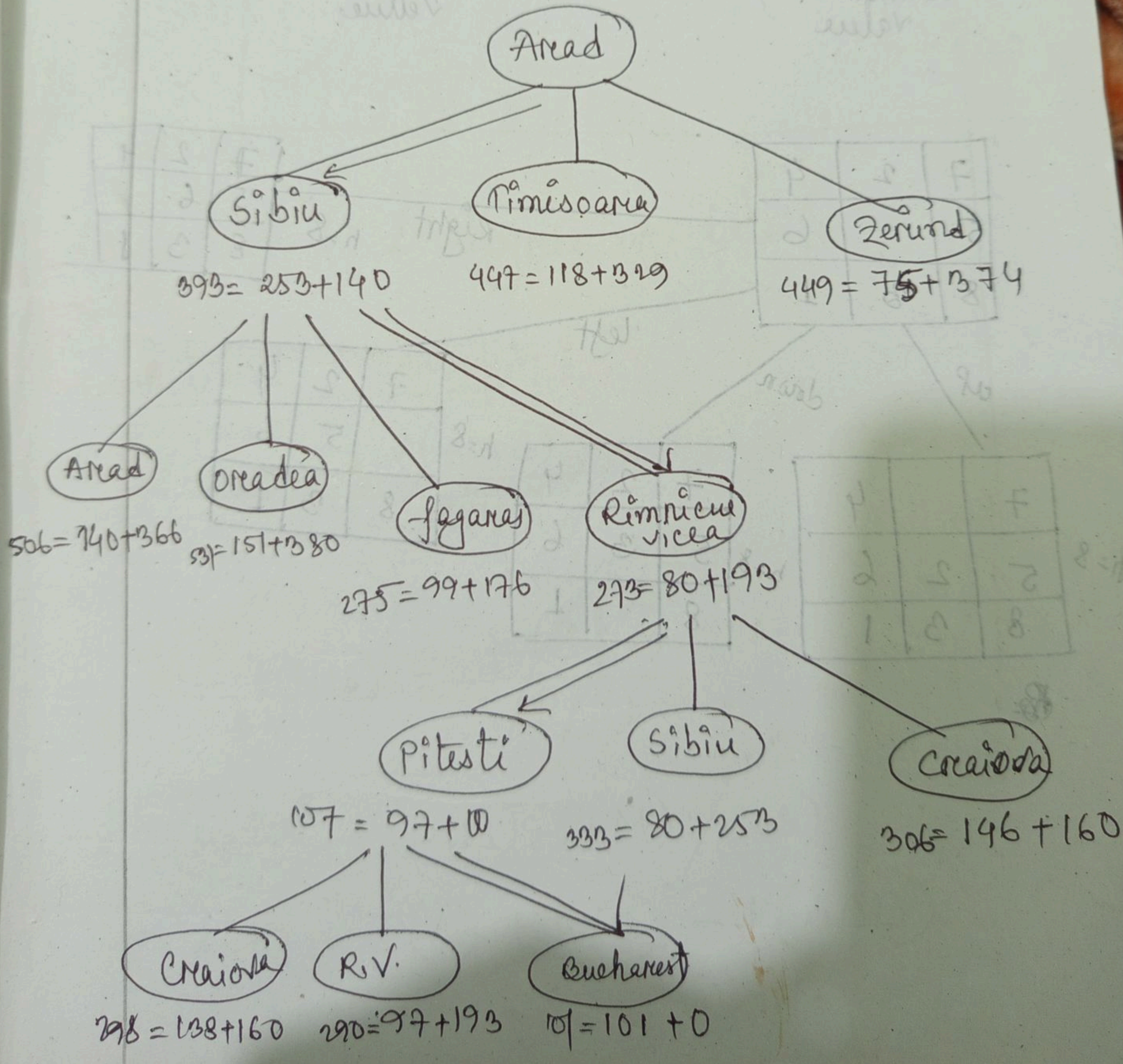
→ $f(n) = h(n)$

↑
heuristic

A* Search:

Evaluation function:

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



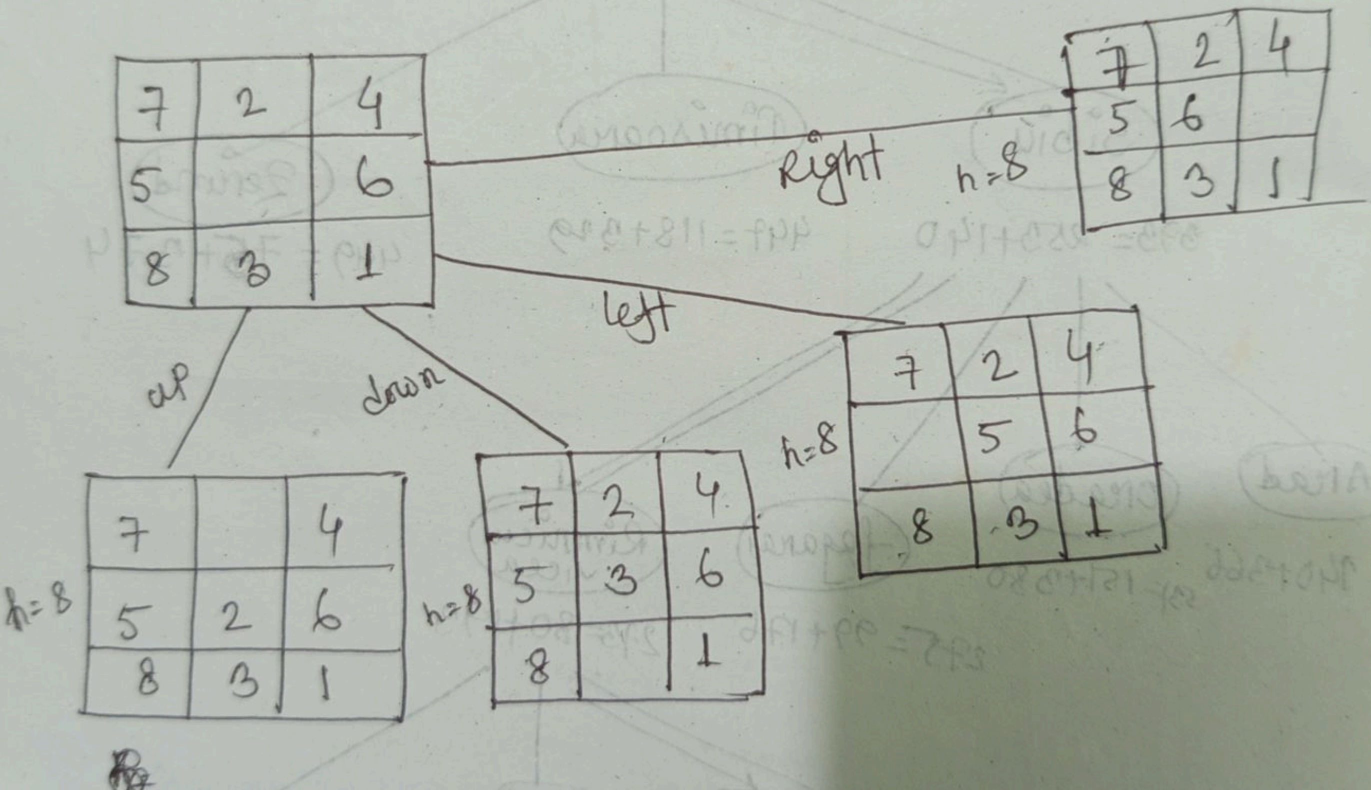
Admissible heuristics:

$h(n) \le h^*(n)$ – underestimation

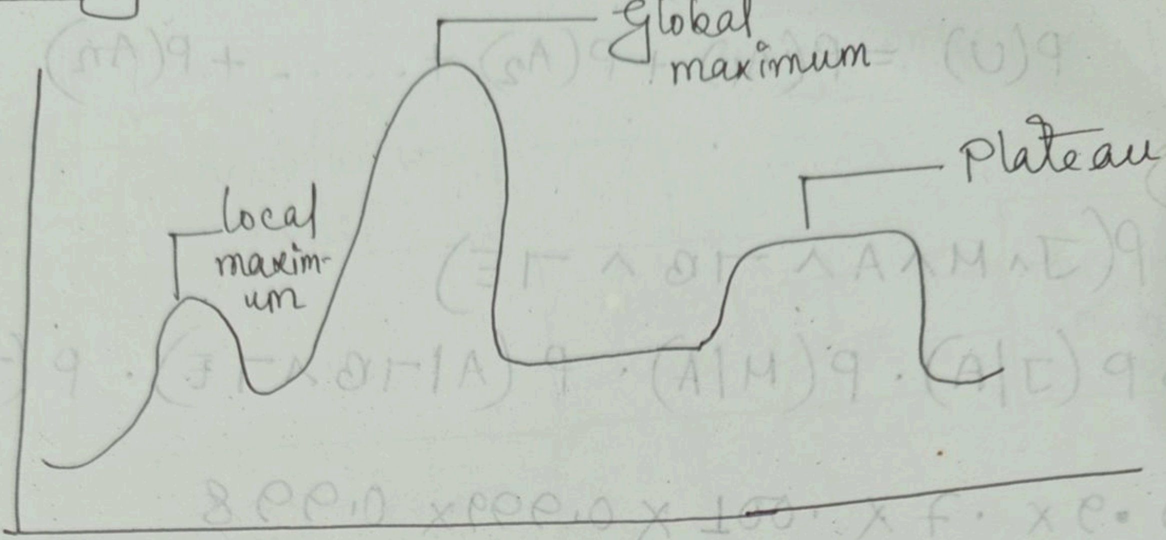
× $h(n) \ge h^*(n)$ – overestimation

estimated value

True/actual value



Hill climbing Search:



Probability = a degree of beleif.

Conditional probability:

$$P(A|B) = \frac{P(A \wedge B)}{P(B)}$$

$$P(M|S) = \frac{\overset{P(M \wedge S)}{\cancel{P(S \wedge M)}}}{P(S)}$$

$$P(M|S) = P(S|M) \cdot P(M) / P(S)$$

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

$P(\bar{A}) = 1 - P(A)$

$P(U) = P(A_1) + P(A_2) + \ldots\ldots + P(A_n)$

(ii) $P(J \wedge M \wedge A \wedge \neg B \wedge \neg E)$

$\Rightarrow P(J|A) \cdot P(M|A) \cdot P(A|\neg B \wedge \neg E) \cdot P(\neg B) \cdot P(\neg E)$

$\Rightarrow .9 \times .7 \times .001 \times 0.999 \times 0.998$

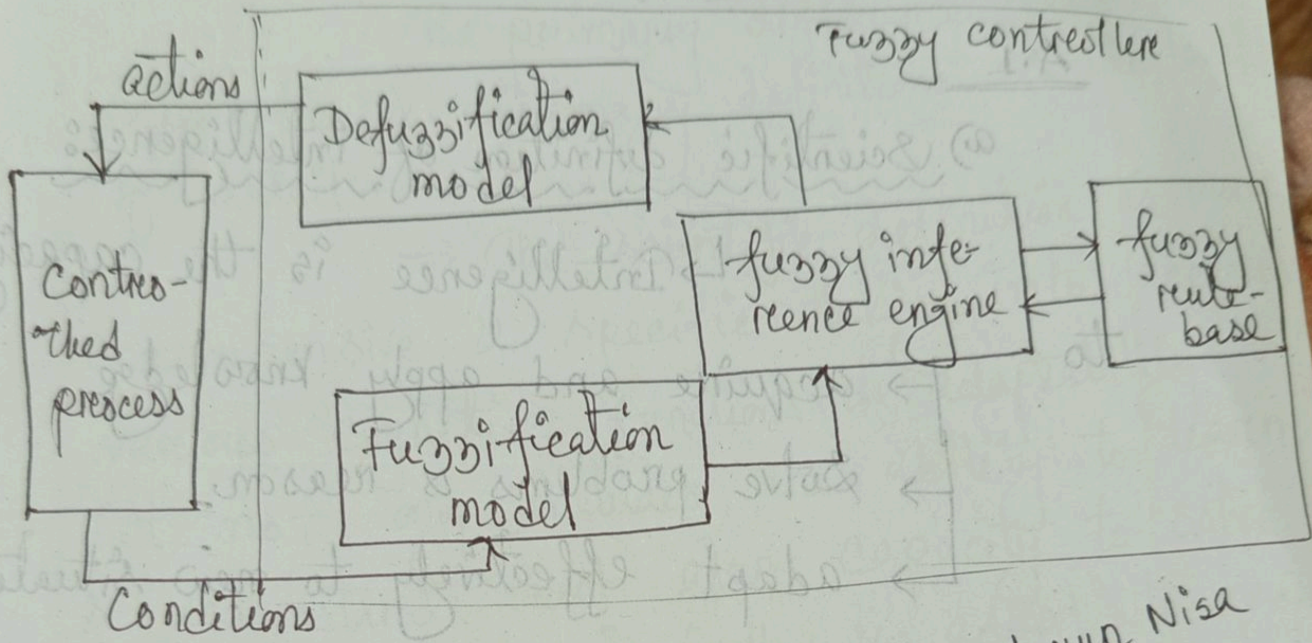
$\Rightarrow 0.00062$

(iii) $P(J \wedge M \wedge A \wedge B \wedge \neg E)$

$= P(J|A) \cdot P(M|A) \cdot P(A|B \wedge \neg E) \cdot P(B) \cdot P(\neg E)$

$= 0.9 \times 0.7 \times 0.94 \times 0.001 \times 0.998$

$= 0.00059$



Meherun Nisa

2019

A.1

a) Scientific definition of intelligence:

↳ Intelligence is the capacity

to
→ acquire and apply knowledge
→ Solve problems & reason
→ adapt effectively to new situation

↳ It involves ability to learn understand & use information to make sound judgement & decisions.

Dictionary definition of intelligence:

↳ Intelligence is the ability to learn or understand or to deal with new situations

The primary difference betn the scientific & dictionary definitions is the level of detail.

→ The scientific definition is more comprehensive & specific, taking into account various cognitive functions & adaptability. It not only includes the ability to learn & underestand but also capacity to solve problems, reason logically & adapt to new situations. It also underscores the use of knowledge fore making informed decisions.

→ On the other hand, the dictionary definition is a concise & simplified explanation of the concept. It just includes the ability to learn, underestand & apply knowledge

6.)

Sophia, the social humanoid robot, possess the following features & signs of intelligence:

1.) Natural Language processing:

Sophia can understand & respond to human language. She is programmed to engage in converesations, answer questions & follow verebal commands

2.) Machine learning:

While not truly learning like a human, Sophia can adapt & improve here responses over time based on patterens & data from here interactions.

3.) Computere vision:

Sophia can recognize faces

& interpret human facial expressions.

5.) Decision-making:

Sophia can make rational decisions and respond quickly and successfully to new situation.

Jarvis is an AI system that can control home system & perform basic tasks such as turning the light on or off, controlling room temparature etc

Features & signs of intelligence that Jarvis possess are:-

1.) Natural language processing:

Jarvis can understand & respond to spoken language.

2) Machine learning: Jarvis can learn from previous interactions

3.) Autonomous decision making:

Jarvis has a degree of autonomy, allowing him to take actions & make decisions based on pre-establish[ed] guidelines.

4.) Simulation of human-like behavior:

Jarvis can simulate hum[an] like reasoning & conversational skills.

5.) Real-time communication:

Jarvis enables real-ti[me] communication with the user

A.2

a.) Search refers to the process of finding a solution to a problem within a defined set of possible options or states.

Search problem is a foremal representation of a problem that requires finding a sequence of actions or steps leading from an initial state to a goal state.

Here's a simple example of how to formulate a search problem:

1.) Initial State (S0): This is the starting point of the problem.

2.) Actions / Operators (A):

↳ Represent the set of possible moves or operations that can be applied to change from one state to another.

3.) State Space (S):

↳ consists of all possible states that can be reached from the initial state using the defined actions.

4.) Goal state (G):

↳ represents the desired outcome of the search problem

5.) Path cost (C):

↳ cost function that assigns a cost to each action.

6.) Search Strategy:

↳ Depth-First Search.
↳ Breadth - " "
↳ A* etc

Examples of data structure that can be used to represent a state space in —

Explicit ways:

1.) Graph

2.) Array or Matrix

3.) Table

4.) Chessboard

5.) Puzzle ~~gid~~ grid

Implicit ways:

1.) Trees

2.) Constraint Satisfaction problem (CSP)

3.) Predicate logic.

4.) Hash table / dictionary.

5.) Bit vectors.

b.)

Initial State:

↳ The monkey on the ground

↳ Bananas suspended from the ~~ci~~ ceiling.

↳ Two crates on the ground

Actions / Successore functions:

↳ Hop on crate

↳ " off "

↳ Move a crate.

↳ Stack one crate on another.

↳ Walk from one spot to another.

↳ Grab bananas.

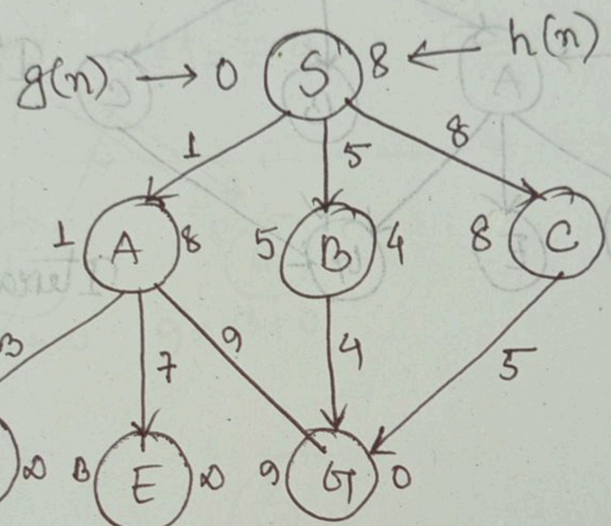
Goal test:

Monkey has bananas.

Cost function:

Number of actions.

c)

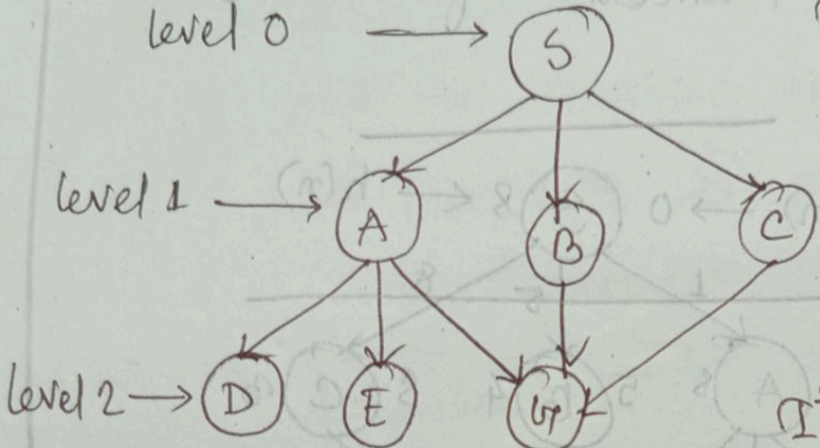


DFS:

Pre-order: Root → Left → Right

S → A → D → E → G

Iterative Deepening :

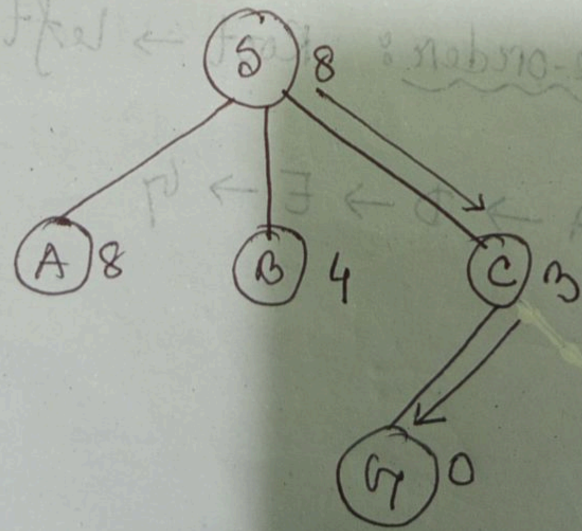


Iteration 0 : S

Iteration 1 : S → A →

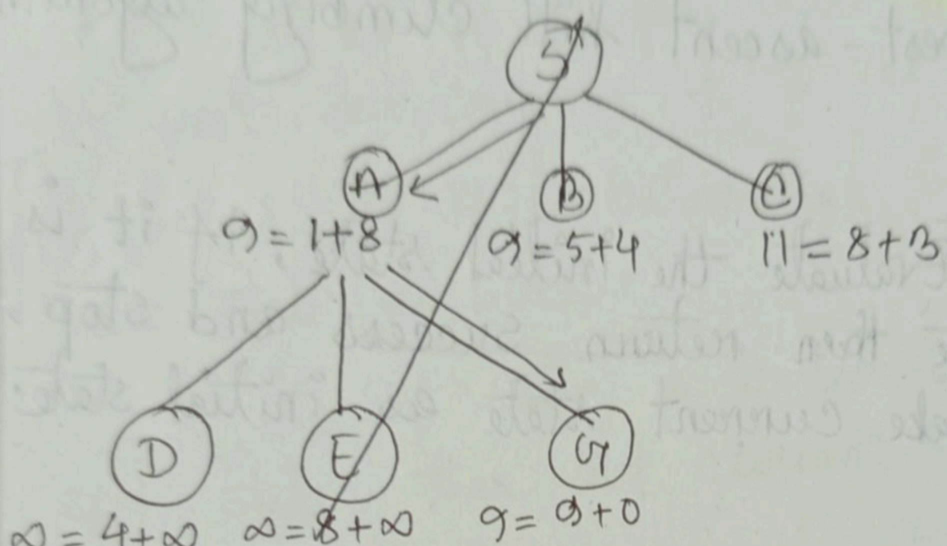
Iteration 2 : S → A → D → E
G → B → C

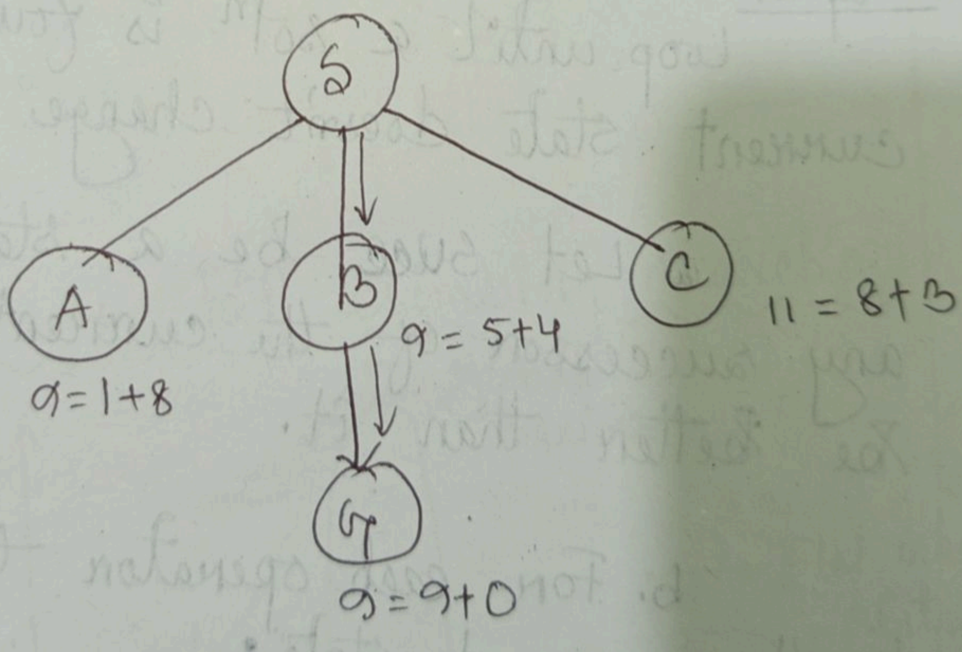
Greedy Search / Greedy Best-First Search :



A* Search:

(a)





A.3 (a)
Steepest-ascent hill climbing algorithm:

Step 1:
Evaluate the initial state, if it is goal state then return success and stop, else make current state as initial state.

Step 2:
Loop until a sol^n is found or the current state doesn't change

a. Let succ be a state such that any successor of the current state will be better than it.

b. For each operator that applies to the current state:

a) Apply the new operator & generate a new state.

b) Evaluate the new state.

c) If it is goal state, then return it & quit, else compare it to the succ

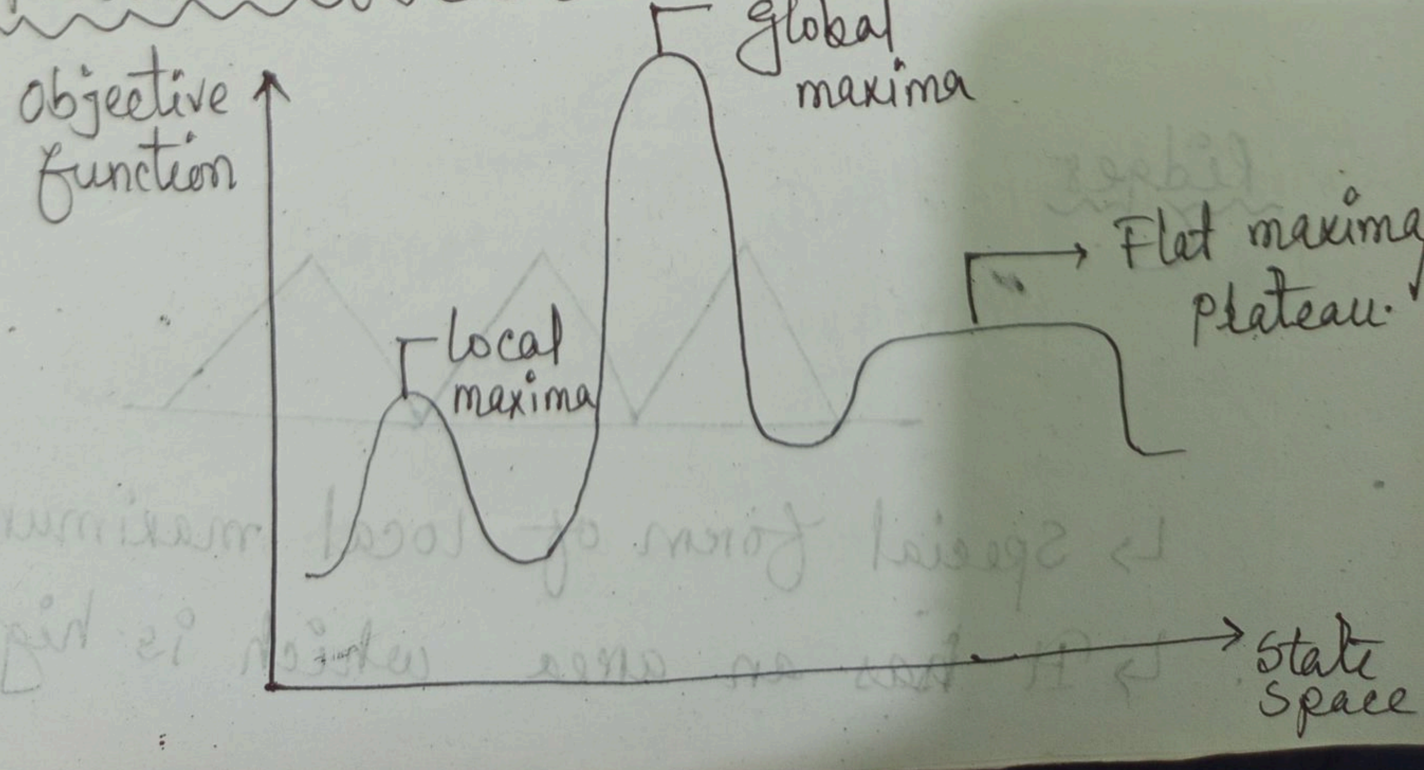
d) If it is better than succ, then set new state as succ

e) If succ is better than the current state, then set current state to succ

Step 3:

Exit.

Problems of hill climbing search:



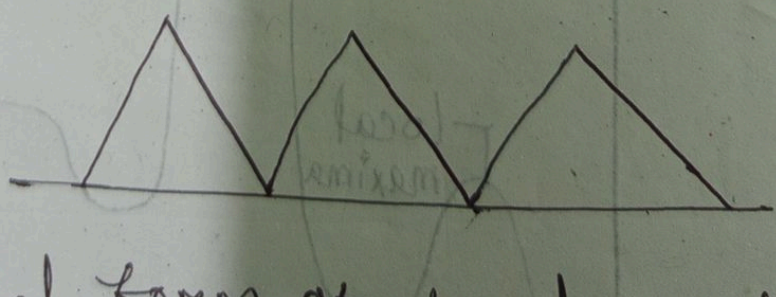
local maximum:

↳ a point in the search space where the objective function has a higher value than its immidiate neighbors but not the highest value in the entire search space.

Flat maximum/Plateau:

↳ flat space where all neighbor states of the current state have the same value.

Ridges:



↳ Special form of local maximum.

↳ It has an area which is higher

than its surrounding areas, but itself has a slope and can't be reached in a single move.

~~The~~ The problems of hill climbing search can be dealt with simulated annealing.

Simulated annealing is a method of solving problems where the goal is to find the best soln for a given set of parameters. In this method, the computer explores all possible soln until it finds one that meets the requirements.

b.) Difference betn Depth-First Limited Search & Iterative Deepening search:

DFLS	IDS
① Fixed depth limit.	① Depth limit increases with each iteration.
⑪ Not guaranteed to be complete.	⑪ Guaranteed to be complete.
③ Use less memory.	③ Use more memory.
④ doesn't explore the entire tree.	④ Explores the entire tree.

Advantages of DFLS over BFS and DFS:

1.) Memory efficieny:

↳ more memory efficient than BFS and DFS.

2.) Faster discovery of Shallow solutions:

↳ more quicker than BFS.

3.) Reduced node expansion:

↳ Expand fewer nodes compared to BFS when the solution is shallow.

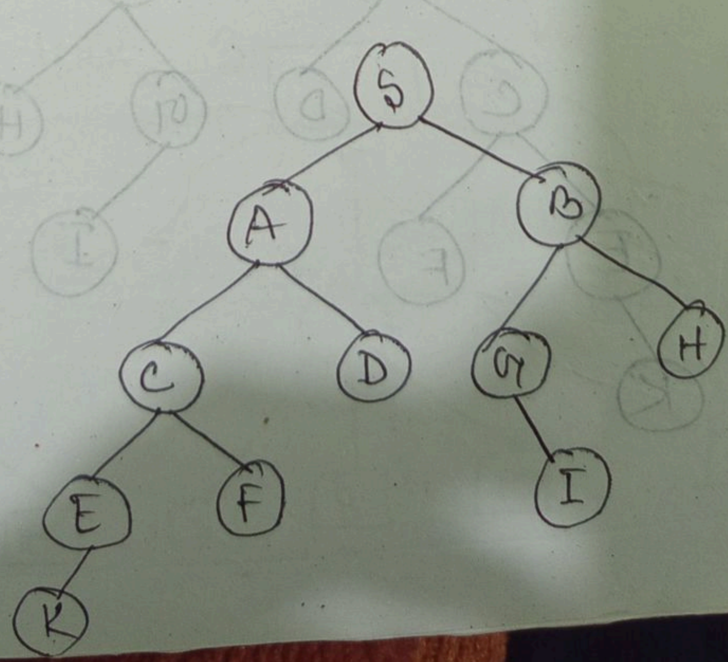
4.) Early pruning of unpromising paths:

↳ quickly identify and prune unpromising paths by stopping the exploration when the depth limit is reached.

5.) Simplified implementation:

↳ easier to implement than BFS.

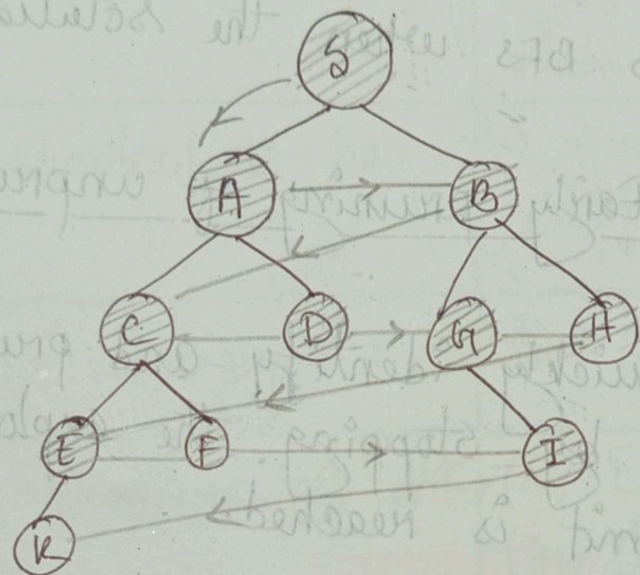
Example:



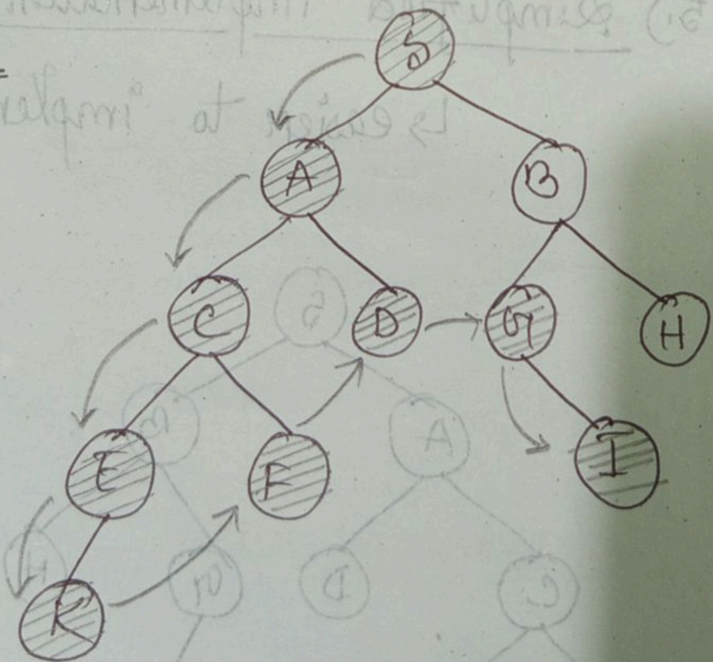
Goal = I

BFS

~~S → A → B~~

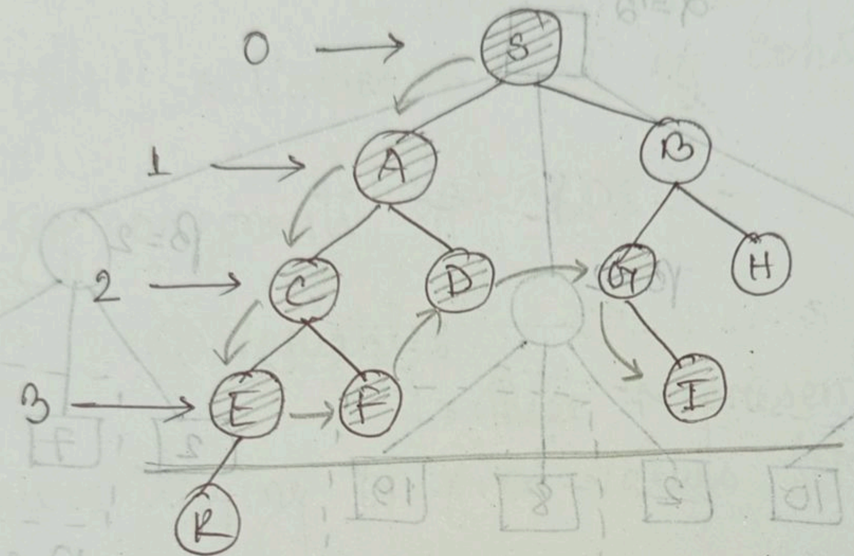


DFS



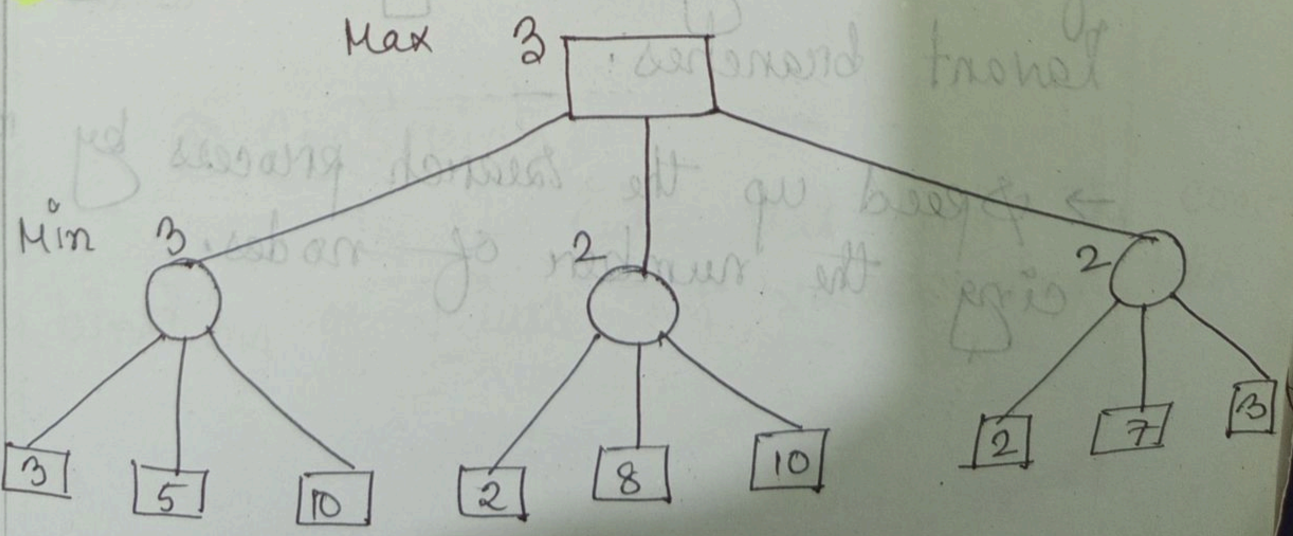
DFLS:

$l = 3$

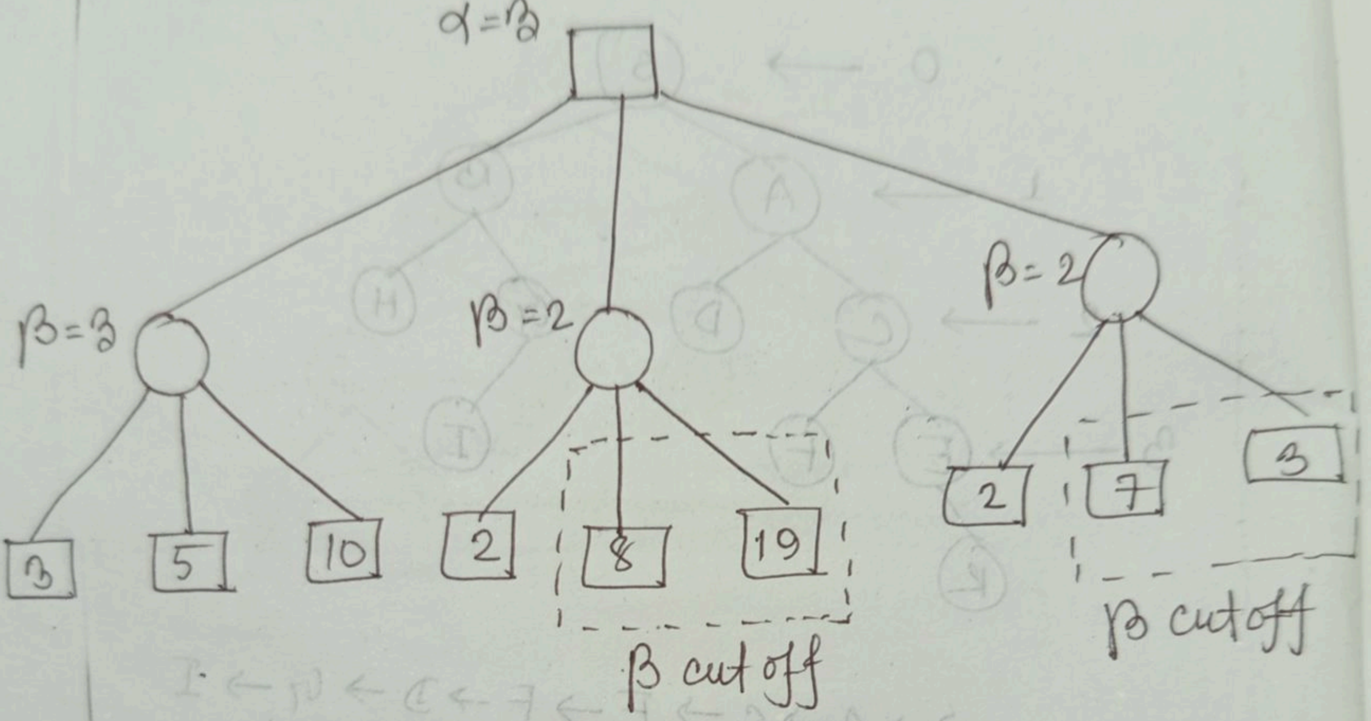


$S \rightarrow A \rightarrow C \rightarrow E \rightarrow F \rightarrow D \rightarrow G \rightarrow I$

(C) Minimax algorithm



Alpha-beta prunning:



Alpha-beta pruning optimize the game tree of minimax algorithm by eliminating irrelavant branches.

→ Speed up the search process by reducing the number of nodes.

2) Constraint satisfaction problem (CSP):

↳ Involves finding a solution that satisfies a set of constraints.

Key components of CSP:

① Variable:

↳ These represent the elements we're trying to find values for

② Domains:

↳ Each variable has a domain, which defines the possible values it can take.

③ Constraints:

↳ Rules that limit the combinations of values that variable can take.

The given problem is an example of CSP.

In the given problem:

variables : The set of regions in the map.

domains : {color1, color2, color3, color4}

~~Contr~~ constraints : No two adjacent region can have the same color

2021

1(a)

See 2019 A.1(b)

1(b) The capabilities ~~of~~ a machine must have to pass the turing test:

1.) Natural language processing: To enable it to communicate successfully in english.

2.) Knowledge Representation: To store what it knows and hears.

3.) Automated Reasoning: To use the stored info to answer questions & to draw new conclusions.

4.) Machine learning: To adapt to new circuinstances and to detect & extrapolate patterns

5.) **Computer visions** – To perceive objects

6.) **Robotics** – To manipulate objects.

1(c) Simple reflex agent:

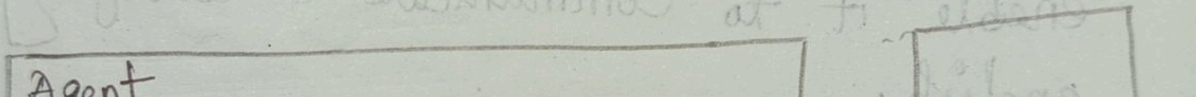


Fig: Simple Reflex agent.

→ Simple reflex agent acts based on straightforward "if this, then do that" rule.

→ It percept current situation from the environment through sensors.

→ It doesn't considere the history of past percepts.

→ The agent has a predefined set of condition-action rules. These rules map specific percepts to corresponding actions.

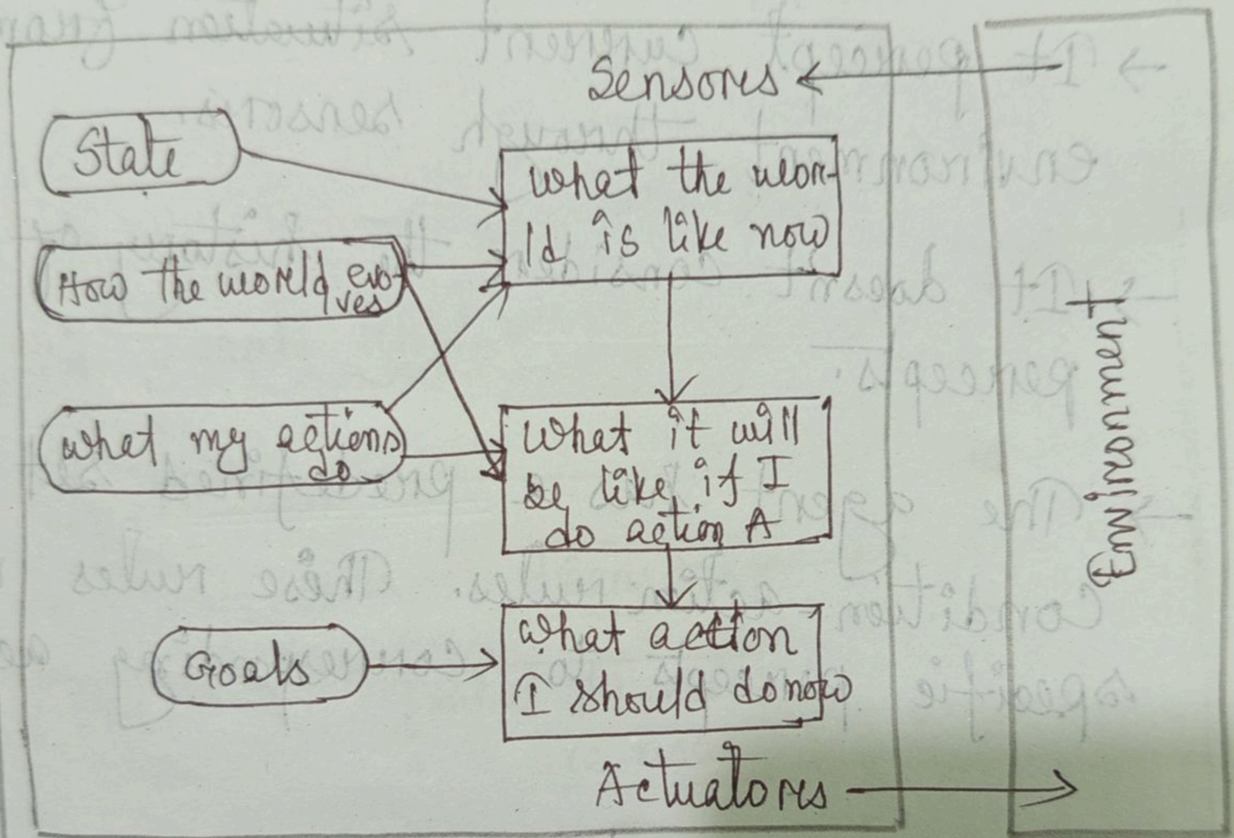
Goal based agent:

→ The agents starts with one or more predefined goals/objectives.

→ ability to plan.

→ Based on current situation and goals, the agent evaluates its plan

→ Carries out the actions as pere its plan.



1(d)

ⓘ PAGE Description:

- Percept:
Amazon Echo perceives voice commands & audio input forom users.

- Actions:
It can perforem actions such as playing music, answering questions, controlling smaret home devices, setting alarems and more

- Goals: To fulfill usere requests ore commands.

- Environment:
Operates in a domestic ore office environment, typically connected to the internet.

(ii) Environment Characterization:

- **Accessible:**
  ↳ Amazon echo can perceive user commands & interact with the environment.

- **Deterministic:**
  ↳ Responds to user commands in a predictable manner.

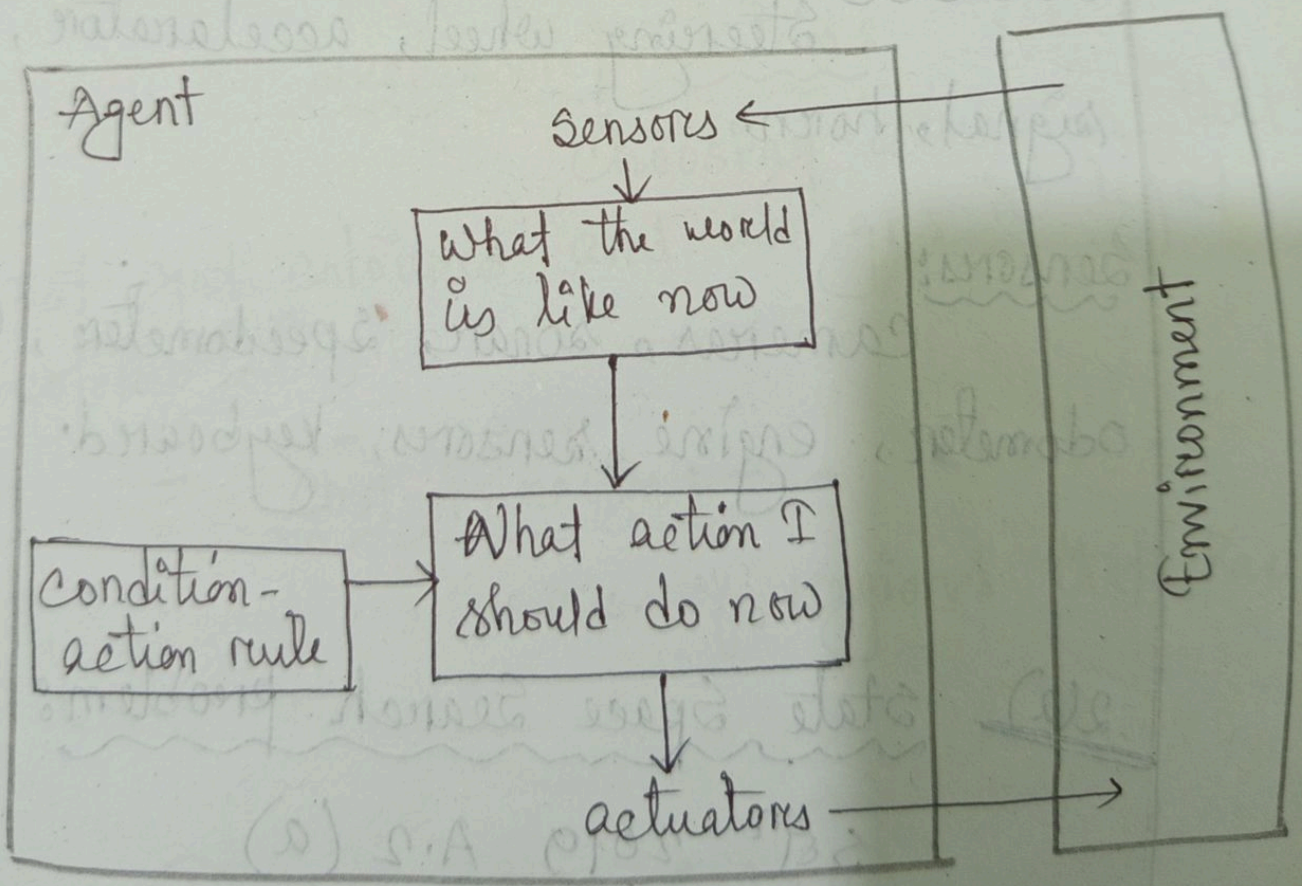
- **non-Episodic:**
  Actions & percepts are inter-related; each action is influenced by the previous state.

- **Static:**
  The environment is mostly static, it doesn't change rapidly by

itself.

- **Discrete:**
  actions are mostly discrete, as it responds to user commands.

1(e) Simple reflex agent architecture is best for automated taxi-driving.



Performance measure:

Safe, fast, legal, comfortable trip, maximize profits.

Environment:

Roads, other traffic, pedestrians, customers.

Actuators:

Steering wheel, accelarator, brake, signal, horn.

Sensors:

cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard.

2(a) State Space Search problem:

S σ^n 2019 A.2 (a)

2(b) (i) initial state: No colors are assigned in any region on the map.

Goal test: Determine if all regions have been colored such that no adjacent regions share the same color.

Successor Function:

Choosing a region that is not yet coloured and assigns a legal color to it.

Cost Function:

No. of regions that have been coloured

(ii) See 2019 A.2 (b)

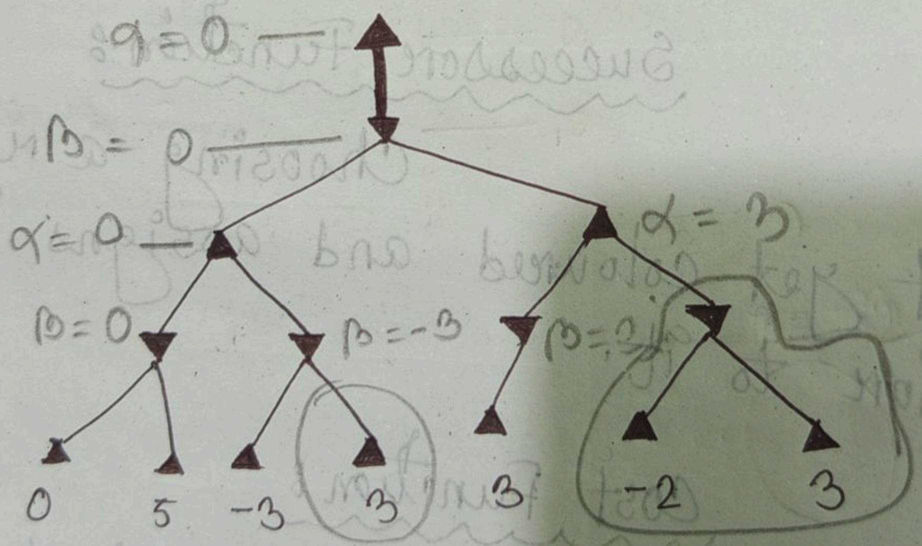
2(c) See 2019 A.2(c).

3(a) See 2019 A.3(a)

3(b) See 2019 A.3(b)

3(c) See 2019 A.3(d)

3(d)



4 (b)

Theorem:

A statement / proposition that has been proven to be true, based on a rigorous, logical argument.

Proof:

Step-by-step explanation that convinces everyone that the theorem is correct.

4 (c)

Valid → If all the answers of the given sentence ~~are~~ are "True".

Satisfiable → If some answers are 'True' and " " " 'False'

unsatisfiable → If all the answers are 'False'.

Given sentence:

$(P \vee Q) \wedge (P \vee \neg Q) \vee P$

P	Q	¬Q	P∨Q	P∨¬Q	(P∨Q)∧(P∨¬Q)	column 6 ∨ P
T	T	F	T	T	T	T
T	F	T	T	T	T	T
F	T	F	T	F	F	F
F	F	T	F	T	F	F

From the truth table, we can see that the sentence is satisfiable.

1 (c)

PAGE Description of Automated Taxi driving agent:

1) Percepts:
The taxi agent perceives information from its environment using various sensor. Environment includes positions of other vehicles, pedestrians, road conditions, traffic signals etc

2) Actions:
The taxi agent takes actions based on its percepts. These actions include accelarating, steering, breaking, changing lanes, obeying traffic signals etc

3) Goals:
Safely and efficiently traspor passengeres to their requested destinations,

minimizing travel time, providing a comfortable & secure ride etc

4) Environment:

Road Network, other vehicles, pedestrians, traffic conditions, weather, obstacles etc

Characterization of Environment:

1) Accessible:

The taxi agent can interact with environment & receives info from the surroundings.

2) Deterministic:

Road regulation, traffic rules largly determine the agent's action.

3) Non-episodic:

The agent's action ~~for~~ are influenced by its history & past percepts.

4.) Dynamic:

Continuously changes over time. vehicles move, pedestrians cross streets, traffic conditions fluctuate etc

5.) Continuous:

vehicle speeds, positions, traffic situations.

2(b) (1) Travelling Salesman:

Initial state:

Any city.

Goal test: Given a city, the ~~fuc.~~ Successor function generates all unvisited cities as possible next states.

Cost function:

Distance bet^n two connected cities.

(11) Missionaries and cannibals problem:

Initial State:

(3,3,1) - (Missionaries, cannibals, Boat on the initial side)

Goal test:

(0,0,1) - (All missionaries, cannibals and boat on the opposite side of the river)

Successor function:

1 Boat carries two missionaries must never be outnumbered by cannibals

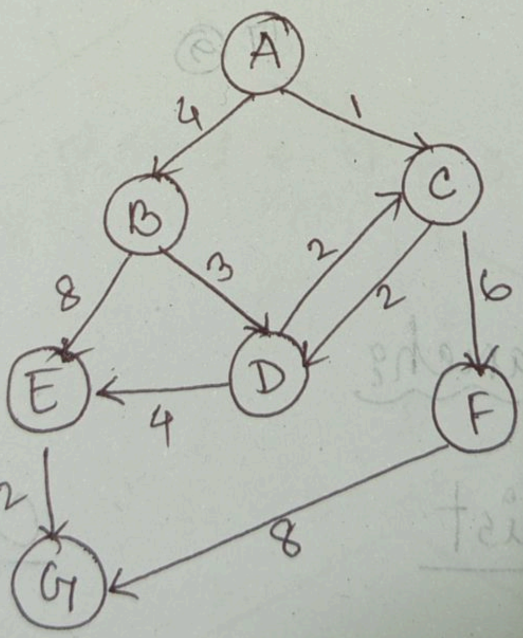
For example:

(3, 3, 1)
(1, 1, 0)
(2, 0, 0)

Cost function:

Uniform, each move has the same cost, e.g., 1 per move

2(C) (i)

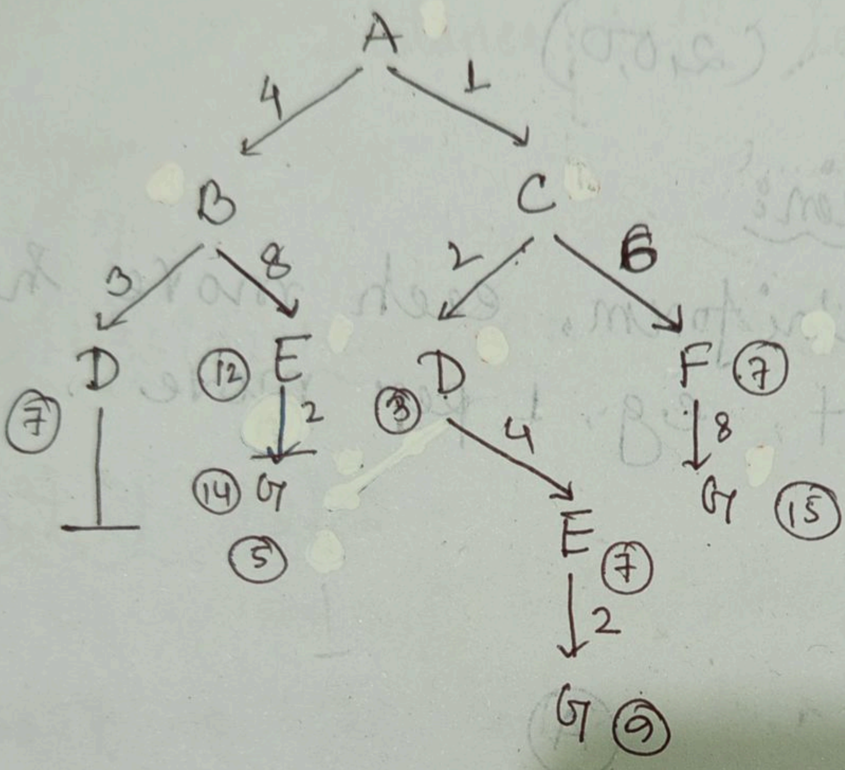


(11) Uniform cost search:

→ Extention of BFS.

visited

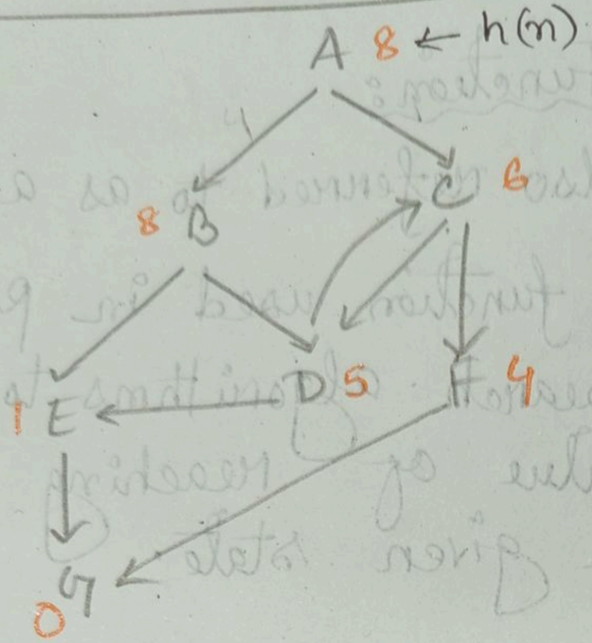
A, C, B, D, F, E, G.



A → C → D → E → G

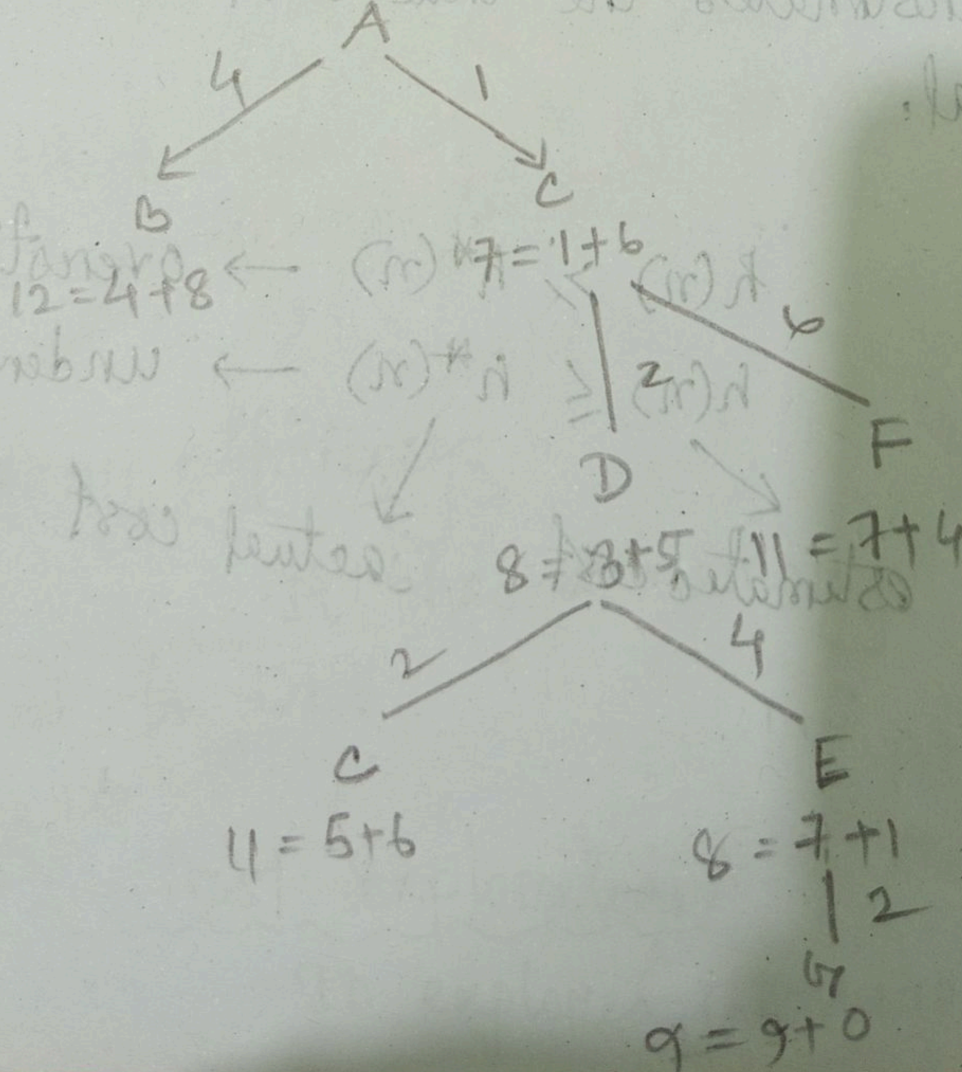
Greedy Search:

Open list	Close list
C, B	A
F, D, B	AC
G, D, B	ACF
	ACFG



A 8 ← h(n)
1
C 6
8 B
D 5
F 4
1 E
0 G

A* Search:



A
4
1
B
C
12 = 4+8
7 = 1+6
2
6
D
F
8 = 3+5
11 = 7+4
2
4
C
E
11 = 5+6
8 = 7+1
2
G
9 = 9+0

3(a) Heuristic Function:

↳ also referred to as a 'heuristic'.

↳ a function used in problem-solving ~~&~~ search algorithms to estimate the cost or value of reaching a goal state from a given state.

↳ An admissible heuristic never overestimates the true cost to reach the goal.

$$h(n) \geq h^*(n) \rightarrow \text{overestimates}$$

$$\underbrace{h(n)}_{\text{estimated cost}} \leq \underbrace{h^*(n)}_{\text{actual cost}} \rightarrow \text{underestimates}$$

3(b) Iterative Deepening (ID) combines the benefits of both BFS and DFS. It does this by repeatedly performing a series of depth-limited searches with increasing depth limits until the goal state is found.

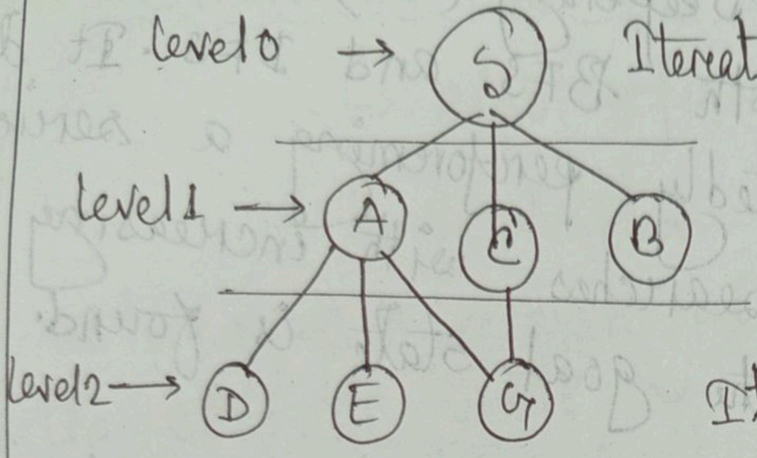
Here's how ID achieves these benefits:

1. Completeness:
ID is a complete search algorithm, which means it is guaranteed to find a solution if one exists.

2 Memory efficiency:
It has a memory footprint similare to DFS.

3. Depth-First Exploration:
ID explores the search space

in a depth first ~~po~~ manner



Iteration 0 : S

Iteration 1 : S → A → C → B

Iteration 2 : S → A → D → E → G → C → B

3(c) Breadth-First Search is an admissible search strategy when:

1.) The search problem involves finding the shortest path from a starting point to a goal.

2.) The problem domain has uniform or equal costs for moving betn states.

3(d) Minimax Algorithm:

↳ determines an optimal move for a player in a two-player game

↳ Recursive/ Backtracking algorithm.

↳ Two players play the game, one is called MAX and other is called MIN.

↳ Both players of the game are opponent of each other, where MAX will select the maximized value & MIN will select the minimized value

↳ Performs a depth-first search algorithm.

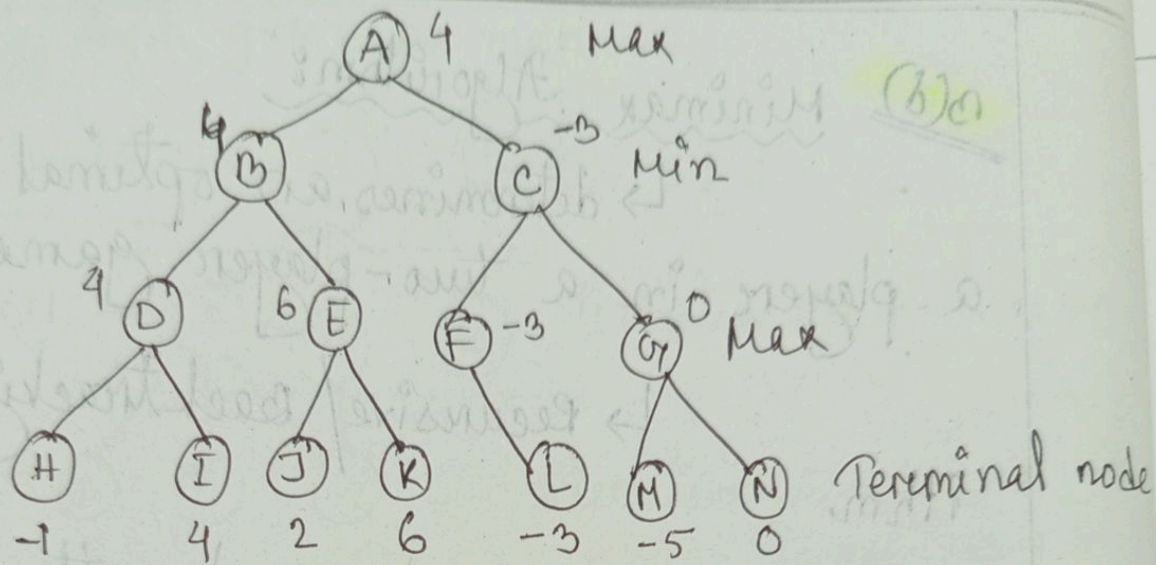


Fig: Minimax game tree.

4(a)

i) Marcus was a man

⇒ man (Marcus)

ii) Marcus was a pompeian.

⇒ Pompeian (Marcus)

iii) Marcus was born in 40 A.D.

⇒ born (Marcus, 40)

iv) All men are mortal.

$\Rightarrow \forall x : man(x) \rightarrow mortal(x)$

v) All pompeian died when the volcano erupted in 79 A.D.

$\Rightarrow erupted(volcano, 79) \wedge \forall x : [pompeian(x) \rightarrow died(x, 79)]$

vi) No mortal lives longer than 150 years.

$\Rightarrow \forall x : \forall t_1 : \forall t_2 : mortal(x) \wedge born(x, t_1) \wedge gt(t_2 - t_1, 150) \longrightarrow dead(x, t_2)$

vii) It is now 2023

$\Rightarrow now = 2023$

viii) Alive means not dead.

$\Rightarrow \forall x : \forall t : [alive(x, t) \rightarrow \neg dead(x, t)] \wedge [\neg dead(x, t) \rightarrow alive(x, t)]$

ix) If someone dies, then he is dead at all later times.

$\Rightarrow \forall x : \forall t_1 \ \& \ \forall t_2 : died(x, t_1) \wedge gt(t_2, t_1) \rightarrow dead(x, t_2)$

4(b)

$\neg alive(Marcus, now)$

$\uparrow$ (viii substitution)

$dead(Marcus, now)$

$\uparrow$ (ix substitution)

$died(Marcus, t_1) \wedge gt(now, t_1)$

$\uparrow$ (v substitution)

$Pompeian(Marcus) \wedge gt(now, 79)$

$\uparrow$ (ii)

$gt(now, 79)$

$\uparrow$ (viii, substitute equals)

$gt(2023, 79)$

$\uparrow$

nil.

4(c) Axioms in clause form:

1.) man (Marcus)

2.) pompeian (Marcus)

3.) born (Marcus, 40)

4.) $\neg$man (x_1) $\vee$ mortal (x_1)

5.) $\neg$ pompeian (x_2) $\vee$ died (x_2, 79)

6.) erupted (volcano, 79)

7.) $\neg$ mortal (x_3) $\vee$ $\neg$born (x_3, t_1) $\vee$ $\neg$ gt (t_2-t_1, 150) $\vee$ dead (x_3, t_2)

8.) now = 2023

9.(a) $\neg$ alive (x_4, t_3) $\vee$ $\neg$dead (x_4, t_3)

9(b) dead (x_5, t_4) $\vee$ alive (x_5, t_4)

10.) $\neg$ died (x_6, t_5) $\vee$ $\neg$gt (t_6, t_5) $\vee$ dead (x_6, t_6)

4 (d) Prove : $\neg$ alive (Marcus, now)

alive (Marcus, now)   9(a)

Marcus/x_4, now/t_3

$\neg$ dead (Marcus, now)   10

Marcus/x_6, now/t_6

5    $\neg$ dead (Marcus, now) $\vee$ $\neg$ gt (now, t_5)

Marcus/x_2, 79/t_5

$\neg$ pompeian (Marcus) $\vee$ $\neg$ gt (now, 79)

Substitute equals.

$\neg$ pompeian (Marcus) $\vee$ $\neg$ gt (2023, 79)

reduce

$\neg$ pompeian (Marcus)   2

$\square$

8(a) ~~By~~ Bayesian Network:

↳ A probabilistic graphical model of how different variables are connected and influence each other, taking into account uncertainty.

↳ It's a way to model and understand complex systems that involve probability.

Syntax of Bayesian network:

1.) Nodes:
↳ Represents random variables events or conditions in a problem domain.

2) Edges: Directed edges connect node to show the probabilistic dependencies or

casual relationships bet^n them.

(2)8

3.) Conditional probability Tables:

↳ Each node has an associated CPT that specifies the probability distribution of that node given its parent nodes in the network.

Semantics of Bayesian network:

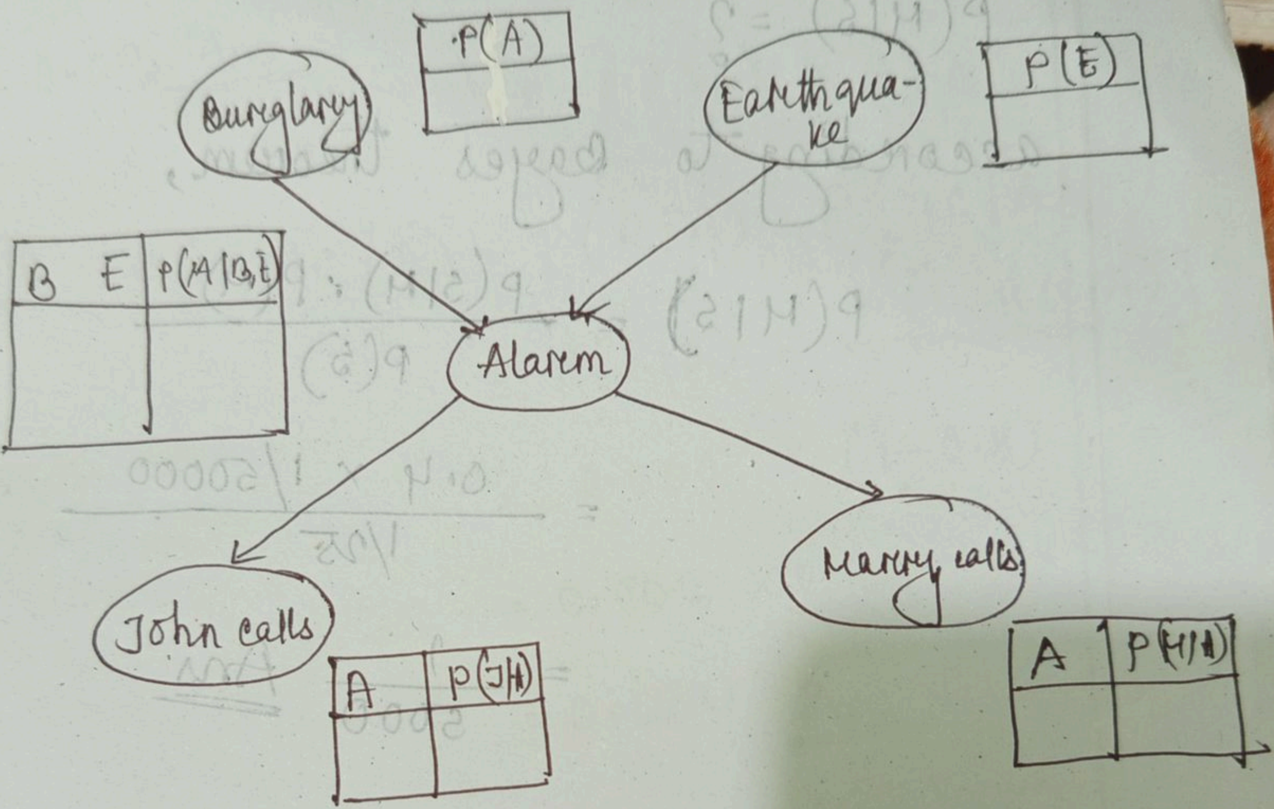
1.) Conditional independence:

↳ Bayesian networks capture conditional independence relationships among the variable.

2.) Joint Probability distribution:

↳ Specifies the probabilities of all possible ~~states~~ combinations of

States for the variables.



8(b) let,

M : patient has meningitis

S : " " stiff-neck

Given, $p(M) = 1/50000$

$p(S) = 1/25$

$P(S|M) = 40\% = 0.4$

$P(M|S) = ?$

according to Bayes theorem,

$$P(M|S) = \frac{P(S|M) \cdot P(M)}{P(S)}$$

$$= \frac{0.4 \times 1/50000}{1/25}$$

$$= \frac{1}{5000} \text{ Ans}$$

8 (C) (II) $P(A) = P(A|B,E) \cdot P(B) \cdot P(E) +$
$P(A|\neg B,\neg E) \cdot P(\neg B) \cdot P(\neg E) +$
$P(A|\neg B,E) \cdot P(\neg B) \cdot P(E) +$
$P(A|B,\neg E) \cdot P(B) \cdot P(\neg E)$.

$= (0.99 \times 0.02 \times 0.001) + (0.01 \times 0.98 \times 0.999)$

$+ (0.9 \times 0.98 \times 0.001) + (0.9 \times 0.02 \times 0.999)$

$= 0.03$ Ans

(iii) $P(R, \neg C \mid A) = P(R \mid A) \cdot P(\neg C \mid A)$

$= P(R \mid A) \times P[1 - P(C \mid A)]$

$= 0.002 \times (1 - 0.8)$

$= 0.002 \times 0.2$

$= 0.004$ Ans

(iv) $P(A \mid R, \neg C) = \dfrac{P(R, \neg C \mid A)\, P(A)}{P(R, \neg C)}$

$= \dfrac{P(R, \neg C \mid A) \cdot P(A)}{P(R) \cdot P(\neg C)}$

$= \dfrac{P(R, \neg C \mid A) \cdot P(A)}{P[P(R \mid A) \cdot P(A) + P(R \mid \neg A) \cdot P(\neg A)] * [P(\neg C \mid A) \cdot P(\ldots) + P(\neg C \mid \neg A) \times P(\neg A)]}$

$$= \frac{0.004 \times 0.03}{\left[(0.002 \times 0.03) + 0.999 \times (1-0.03)\right] \times \left[(1-0.8) \times 0.03 + (1-0.1) \times (1-0.03)\right]}$$

$$= 0.000014.$$

2021

8(a) We need to specify 5 probabilities:

p(Burglary)
p(Alarm 1 | Burglary)
p(Alarm 1 | ¬ Burglary)
p(Alarm 2 | Burglary)
p(Alarm 2 | ¬ Burglary)

A joint probability table would need,

$2^3 - 1 = 7$ probabilities.

8(b) (i) Expressing the joint distribution P(B, E, A, J, M) in terms of conditional probability:

$$P(B, E, A, J, M) = P(B) \cdot P(E) \cdot P(A \mid B, E) \cdot P(J \mid A) \cdot P(M \mid A)$$

(ii) $P(J \wedge M \wedge A \wedge \neg B \wedge \neg E)$

$= P(J|A) \cdot P(M|A) \cdot P(A|\neg B \wedge \neg E) \cdot P(\neg B) \cdot P(\neg E)$

$= 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998$

$= 0.00062$ Ans

(iii) $P(J \wedge M \wedge A \wedge B \wedge \neg E)$

$= P(J|A) \cdot P(M|A) \cdot P(A|B \wedge \neg E) \cdot P(B) \cdot P(\neg E)$

$= 0.9 \times 0.7 \times 0.94 \times 0.001 \times 0.998$

$= 0.00059$ Ans

- What is AI? Task domains of AI?
- What is AI Technique?
- What is Intelligent agent? Describe diff. agents with necessary block diagram.
- Different approaches of AI.
- Problem characteristics.
- Chat GPT, Generative AI, Open AI
- Search Space (Table → Tree) problem.
- Explain Alpha-Beta pruning algo.
- Minimax search algo.

- What is goal-stack planning?

  (i) UNSTACK (A,B)
  (ii) STACK (A,B)
  (iii) PICKUP (A)
  (iv) PUTDOWN (A)

  } Specification of these operators/actions.

- water jug problem.
- simple block world problem & Sussman anamoly problem
- Conflict resolution, how good conflict resolution can be used to reduce back-tracking.
- rule based expert system problem ও prev ques ~~theke~~ থেকে।
- Hypothesis, Inductive learning method.
- Bayesian Network, syntax and semantics of " ".
- Bayes's Theorem.
- menigitis & stiff-neck example ও।
- Burglary & earthquake " ".

[4] Short notes:

(i) waiting for quiescence
(ii) Iterative Deeping.
(iii) Futility-cutoff.
(iv) Fuzzy logic
(v) Uncertainty.

[5] Describe PEAS and explore wumpus world problem

[5] What is resolution?

[5] Describe Minimax procedure

[5] Forward and Backward Reasoning. What Factors that determine whether it is better to reason forward or backward.

[5] Knowledge Representation & mapping, role of KR.

- Steepest-ascent hill climbing algo, Problems of hill climbing search & soln.

- Algo for conversion to clause form with example

- Marcus problem ex,

- Expert system, expert system shell, major component of expert system.

- What is learning? Diff. types of learning method
  - → Supervised
  - → unsupervised
  - → Reinforcement

- NLP, Steps through which NLP is conducte

- approaches to solve conflict resolution.

- ANN, ANN vs biological network,

similarities & dissimilarities.

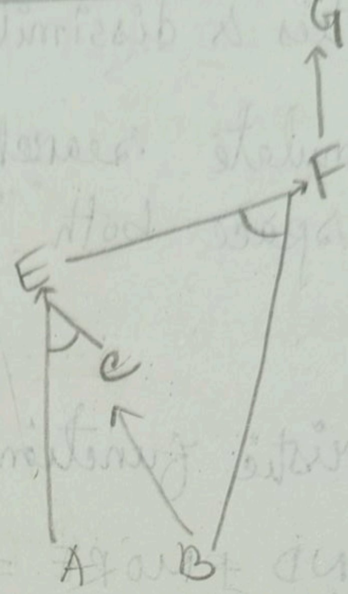
Formulate search problem, Represent a state space both in explicit & implicit ways

Heuristic function.

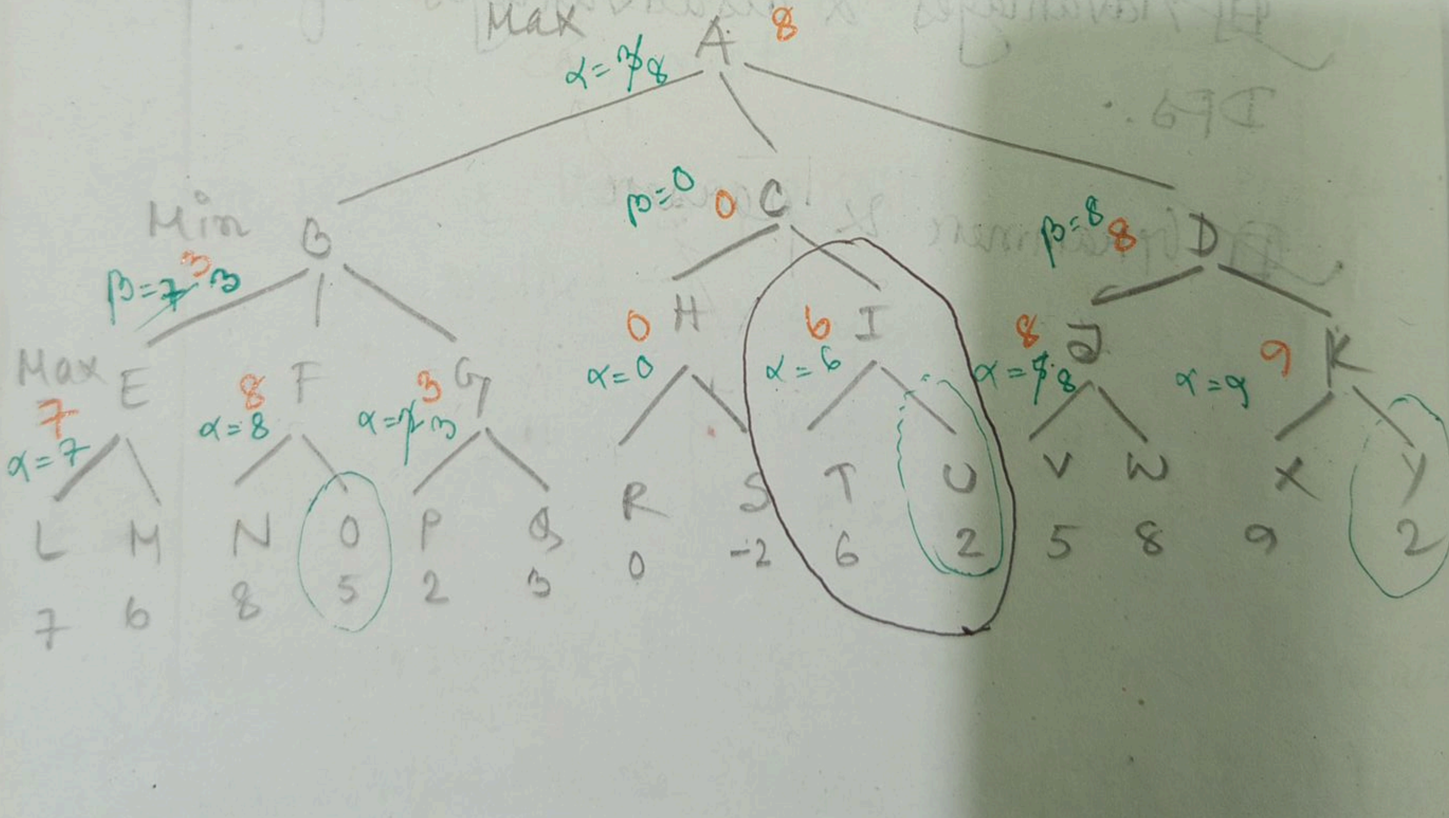
SEND + MORE = MONEY

Advantages & disadvantages of BFS & DFS.

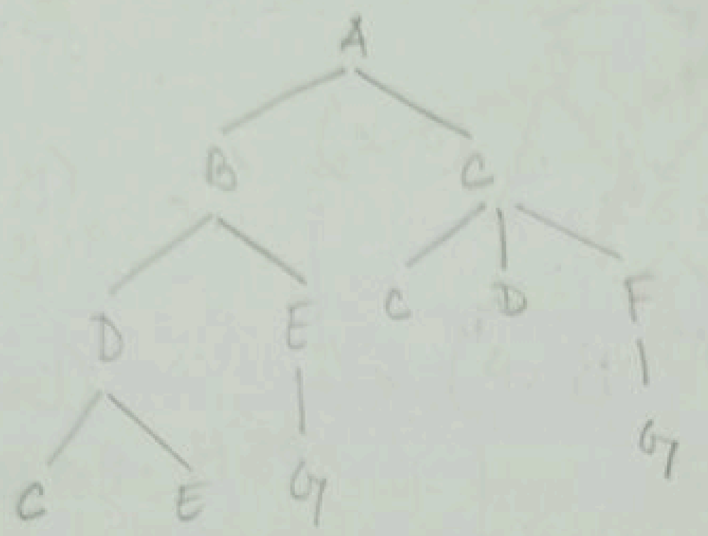
Grammer & parser



G
F
E
C
A
B



Max
A
8
α = 3 8
Min
B
β = 7 3
C
β = 0
0
D
β = 8
8
Max
E
7
α = 7
F
8
α = 8
G
3
α = 3
H
0
α = 0
I
6
α = 6
J
8
α = 5 8
K
9
α = 9
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
7
6
8
5
2
3
0
-2
6
2
5
8
9
2



Task domain of AI

① Mundane task
② Foremal task
③ Experet 4

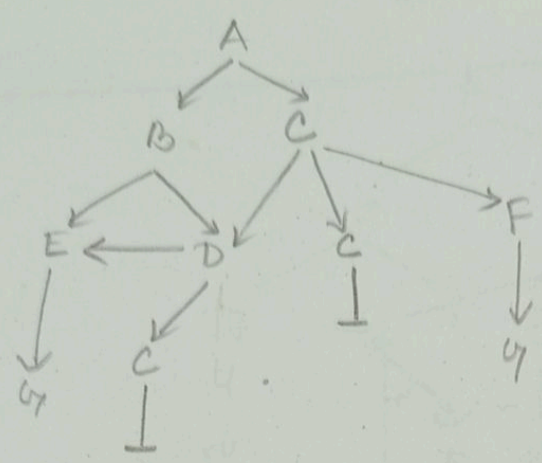
Water jug problem:

① state : (x, y)

$x = 0, 1, 2, 3, \text{or } 4$ $y = 0, 1, 2, 3$

② start state :

$(0, 0)$



Goal state:

$(2, y)$ for any n

State space:

1.) $(x, y) \longrightarrow (4, y)$
if $x < 4$

2) $(x, y) \longrightarrow (x, 3)$
if $y < 3$

3.) $(x, y) \longrightarrow (\cancel{x} - d, y)$
if $x > 0$

4.) $(x, y) \longrightarrow (x, y - d)$
if $y > 0$

5.) $(x,y) \rightarrow (0,y)$
if $x>0$

6.) $(x,y) \rightarrow (x,0)$
if $y>0$

7.) $(x,y) \rightarrow (4, y-(4-x))$
$x+y \geq 4,\ y>0$

8.) $(x,y) \rightarrow (x-(3-y), 3)$
$x+y \geq 3,\ x>0$

9.) $(x,y) \rightarrow (x+y, 0)$
$x+y \leq 4,\ y>0$

10.) $(x,y) \rightarrow (0, x+y)$
$x+y \leq 3,\ x>0$

11.) $(0,2) \rightarrow (2,0)$

12.) $(2,y) \rightarrow (0,y)$

(0,0)
(0,3)
(3,0)
(3,3)
(4,2)
(0,2)
(2,0)

water jug problem:

① state:

(x,y) $\quad x = 0,1,2,3,4 \;;\; y = 0,1,2,3$

② start state:

$(0,0)$

③ goal state:

$(2,n)$ for any n

State space:

1.) $(x,y) \rightarrow (4,y)$
if $x<4$

2.) $(x,y) \rightarrow (x,3)$
if $y<3$

3.) $(x,y) \rightarrow (x-d,y)$
if $x>0$

4.) $(x,y) \rightarrow (x,y-d)$
if $y>0$

5.) $(x,y) \rightarrow (0,y)$
if $x>0$

6.) $(x,y) \rightarrow (x,0)$
if $y>0$

7.) $(x, y) \longrightarrow (4, y-(4-x))$

if $x+y \geq 4$, $y > 0$

8.) $(x, y) \longrightarrow (x-(3-y), 3)$

if $(x+y \geq 3$, $x > 0$

9.) $(x, y) \longrightarrow (x+y, 0)$

if $x+y \leq 4$, $y > 0$

10.) $(x, y) \longrightarrow (0, x+y)$

if $x+y \leq 3$, $x > 0$

11.) $(0, 2) \rightarrow (2, 0)$

12) $(2, y) \rightarrow (0, y)$

(5) They occupy more space	(5) They occupy less space
(6) Scan conversion is required.	(6) Scan conversion is not required
(7) Cost less.	(7) Cost more
(8) They work best when it comes to editing photos.	(8) They work best when it comes to draw, illustrate & design logos.

$$\begin{array}{rccccc} & & c_3 & c_2 & c_1 & \\ & & S & E & N & D \\ + & & M & O & R & E \\ \hline & M & O & N & E & Y \end{array}$$

Initial State

$M = 1$, $S = 8$ or 9

$N = E + 1$

$O = 0$

$c_2 = 1$

$N + R > 9$, $E \neq 9$

$E = 5$

$S = 9$, $N = 6$

$E + D = Y$

$\Rightarrow 5 + D = Y$

or, $E + D = 10 + Y$

$\Rightarrow 5 + D = 10 + Y$

$C_1 = 0$ $\quad$ $C_1 = 1$

$5+D = Y$
$N+R = 10+E$
$\Rightarrow 6+R = 10+5$
$R = 9$

Conflict

$5+D = 10+Y$
$\Rightarrow D = 5+Y$
$R = 8$

$D=7$ $\quad$ $D=8$

$Y = 2$ $\quad$ $Y = 3$

Conflict

$C_3 = 0$ $\quad$ $C_2 = 1$ $\quad$ $C_1 = 1$

S 9	E 5	N 6	D 7
M 1	O 0	R 8	E 5

M 1	O 0	N 6	E 5	Y 2

Role of ~~KR~~ KR:

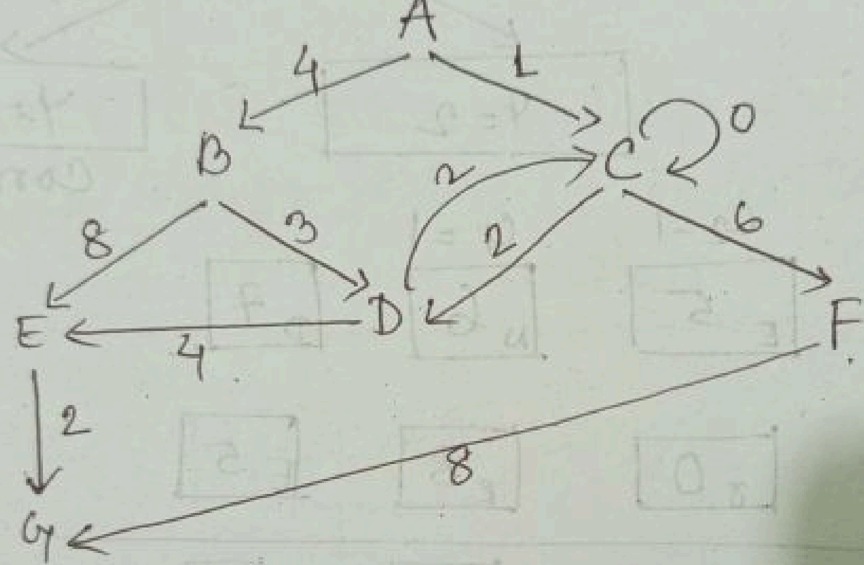
(i) A KR is a surrogate
(ii) " " " " set of othological commitments.
(iii) " " " " fragmentary theory of intelligent reasoning.

(iv) A KR is a medium of efficient ~~informa-~~ ~~tion~~. computation

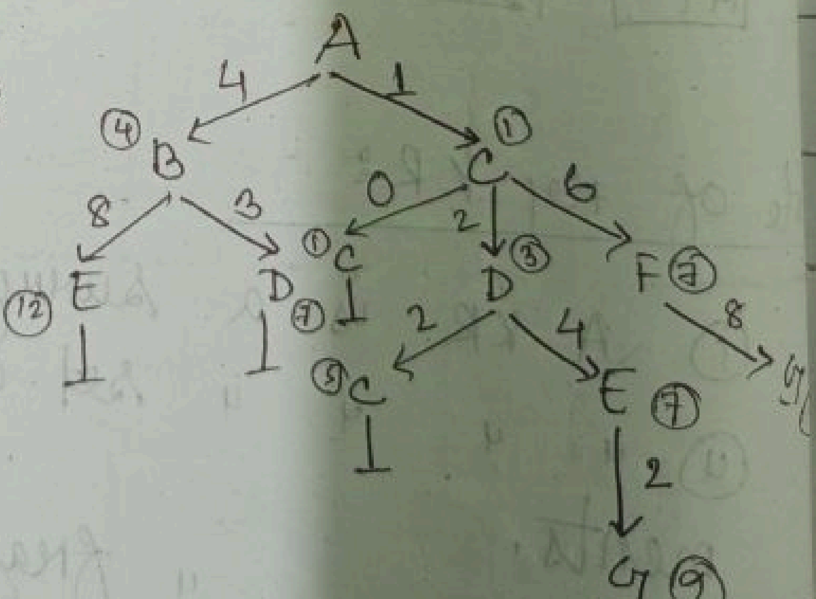
Human expression

(v) A

a)



b) Uniform search:



Priority queue:

~~A~~ ~~C~~ B

~~A~~ ~~C~~ ~~D~~ B F

~~A~~ ~~C~~ ~~D~~ ~~B~~ ~~F~~ ~~E~~ G

Visiting nodes:

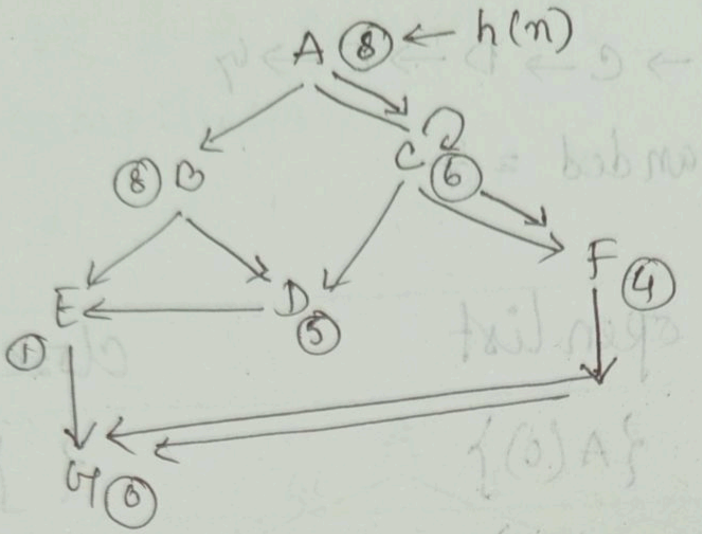
A C D B F E G

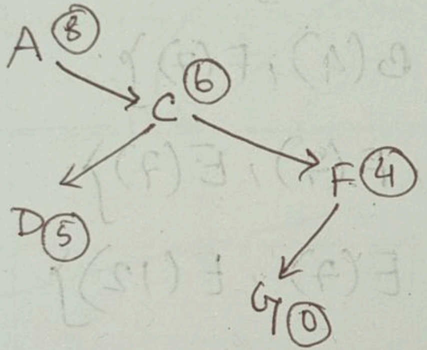
Total cost = 9
Solution path = A → C → D → E → G
Total node expanded = 7

Expanded node	open list	close list
	{A(0)}	{ }
A	{C(1), B(4)}	{A}
C	{D(3), B(4), F(7)}	{A, C}
D	{B(4), F(7), E(7)}	{A, C, D}
B	{F(7), E(7), E(12)}	{A, C, D, B}
F	{E(7), E(12), G(15)}	{A, C, D, B, F}
E	{G(9), E(12), G(15)}	{A, C, D, B, F, E}
G	{E(12), G(15)}	{A, C, D, B, F, E, G}

<u>Greedy search</u>

$f(n) = h(n)$





Total expanded node = 4

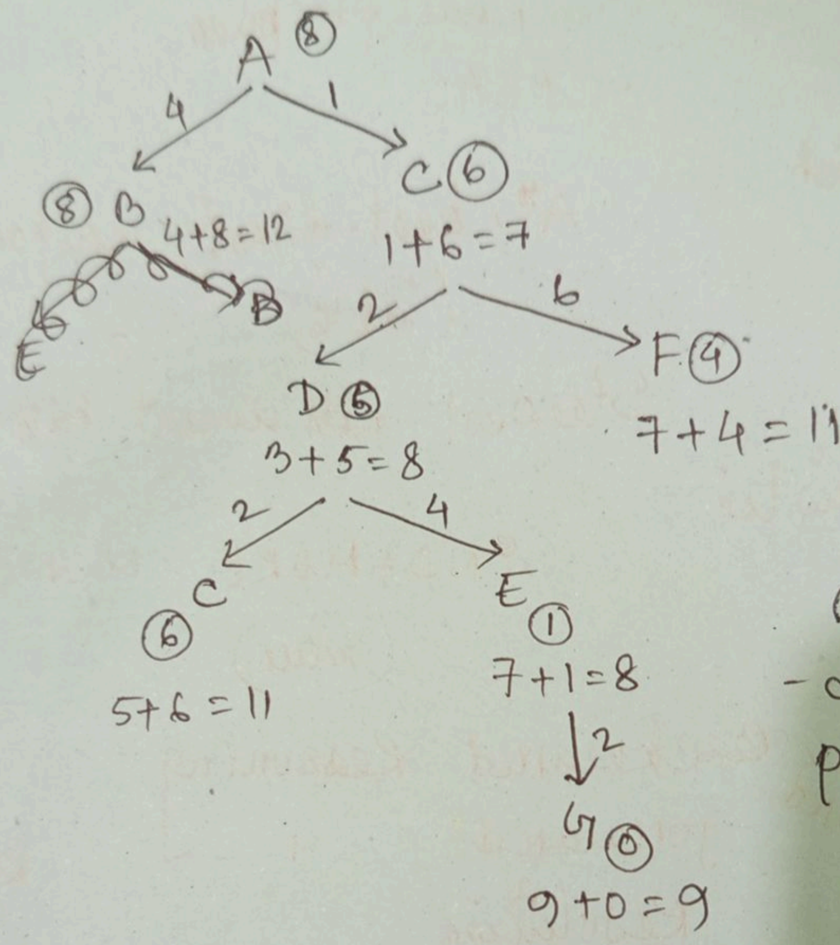
solution path = A → C → F → G

cost = 15

Expanded node	openlist	closelist
	{A(8)}	{ }
A	{ C(6) , B(8) }	{A}
C	{ F(4), D(5), B(8) }	{A, C}
F	{ G(0), D(5) , B(8) }	{A, C, F}
G	{ D(5) , B(8) }	{A, C, F, G}

A* search:

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



Total expanded node = 5

path = A → C → D → E → G

cost = 9

Expanded node	openlist	Close list
	{A(8)}	{ }
A	{C(7), B(12)}	{A}
C	{D(8), F(11), B(12)}	{A, C}
D	{E(8), F(11), C(11), B(12)}	{A, C, D}
E	{G(9), F(11), C(11), B(12)}	{A, C, D, E}
G	{F(11), C(11), B(12)}	{A, C, D, E, G}

Generative AI
ChatGPT → Short note
open AI

conflict Resolution

✓ wumpus ~~pro~~ problem
✓ PEAS

~~Rich~~

A*, best-first, uniform, greedy
(Table)

Steepest-~~hill~~ ascent hill ☆

local heuristic
global "

SEND + MORE = MONEY
(value)

KR, Role, Backward Reasoning
forward "
Resolution
WFF

Russel.
↑
Bayesian Network

$R_1 : R_2$
forward, backward
Russel

ANN
AN vs BN

pre condition, Action

α-β cut off
minimax, ID
Goal

Supervise
un "
Reinforce learning

Expert system → component, Shell, Definition

Inductive learning
fuzzy, uncertainty

NLP → Definition
→ Grammer, parser