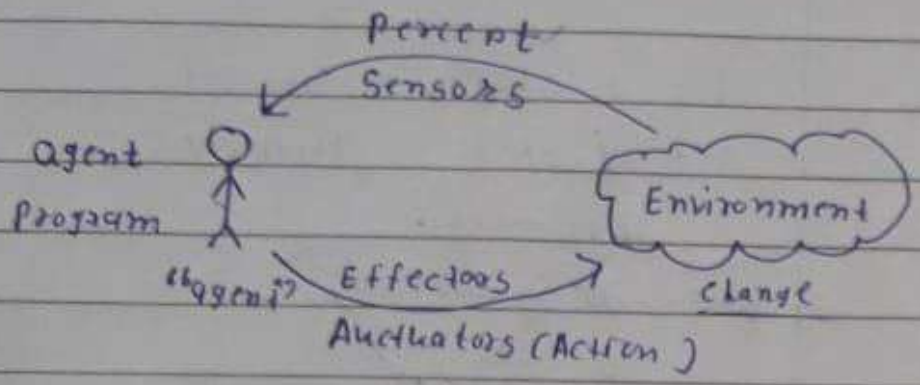
AI preparation:

★ Agents / Inteligent agents



- agent percept from environment of two type. ① current
② Historical
- based on the perception "agent" perform action through "agent program".
- So means on the basics of perception the output is known as effectors to change the environment.
- The whole flow

Agent → Percept → Decision → Action

From Environment

↳ to Environment

★ Goal of Agent :

- High Performance , - Optimised Result , - Rational Action

★ P → Performance measure
E → Environment
A → Actuators
S → Sensors

★ PEAS Description for Automated CAR ...

- Agent Type : Automated car
- Performance Measure : Minimizing fuel Consumption , Safe , fast , Accurate , Comfort , Minimum cost
- Environment : Road , other vehicles , traffic Sign & signal , ~~Pedist~~ Pedestrain , Road ~~cost~~ work
- Actuators : steering wheel , Acceleration , Break , indicator , Voice interaction with Person , Interaction Display
- Sensors : Camera , Speedometer , GPS , Range Sensor , SONAR , Keyboard & Microphone for user interaction.

★ Terms which are used in Task Environment of PEAS...

- agent = an agent can be anything which persist the environment through sensors and perform certain action based on the persistions to change the environment with the use of actuators.
- Sensors = an sensors can be defined which helps the agent to take inputs from the environment.
- agent program = an agent program is one Logic / program which process the input of the sensors through environment
- Actuators = an actuator is defined which perform some action based on the agent program's output and change the environment.
- Environment = An environment is everything in the world which surrounds the agent, but it is not a part of an agent which also described as a situation in which the agent is present.
- Effectors = Mechanism / devices through which the agent can act to the environment.
- Percept = The raw sensory data / information received by the agent through sensor from the environment.

Gajjar Harsh HARSH P GAJJAR

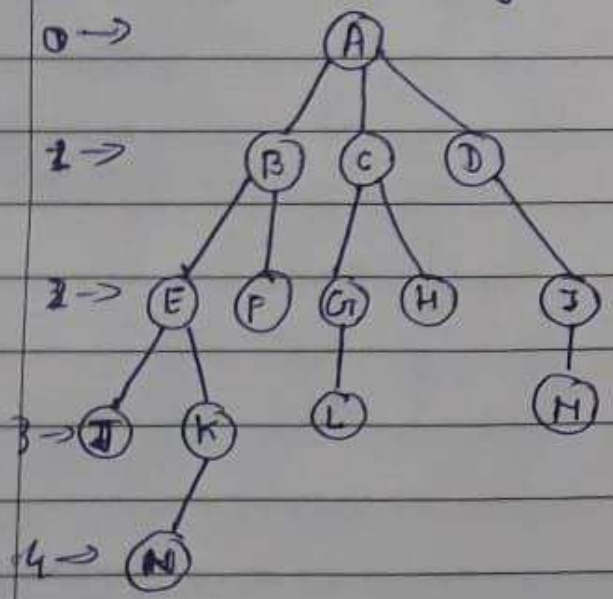
★ Types of Agent

① Simple Reflex agent - Fully observable
② Model based Reflex agent
③ Goal based agent - partially observable
④ Utility based agent - partially observable
⑤ Learning agent

★ Confusion Matrix :

TP	FN
FP	FN

★ BFS (Breadth First search) : uninformed search technique, blindfold search technique, Brutforce technique, ~~Shallow~~ level search technique.
- shallowest node, FIFO, Complete



A
BCD
CDEF
DEFGH
EFGHI
FGHIJK
GHIJK
IJKL

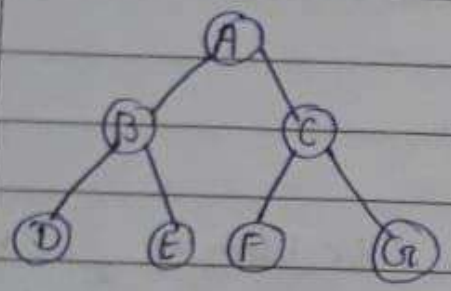
IJKL
JKLM
KLM
LMN
MN
N

B A S E
B A L L
G A M E (5)
3 7
2-8-A P R I L
1-4-F O O L S
3 9 - P R A N K
7 5

3 2
4 1
5 0

* DFS (Depth first search) : Uninformed search technique,

- Stack (LIFO) ; Deepest node, Incomplete, non-optimal solution



ACGFBED

A B ~~C~~

A B F ~~G~~

A B ~~F~~

A B

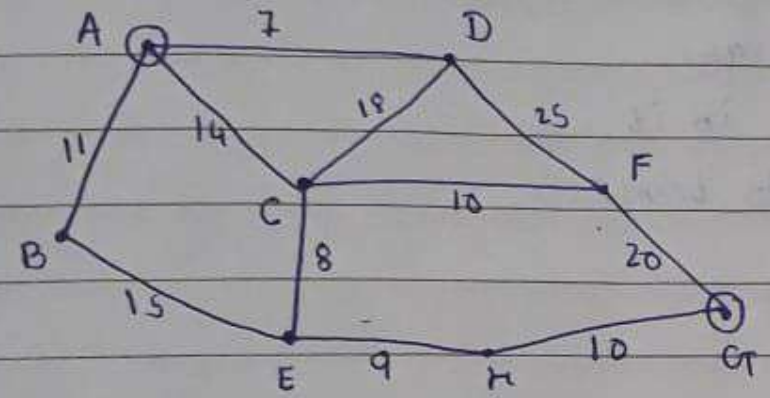
A D ~~E~~

A ~~D~~

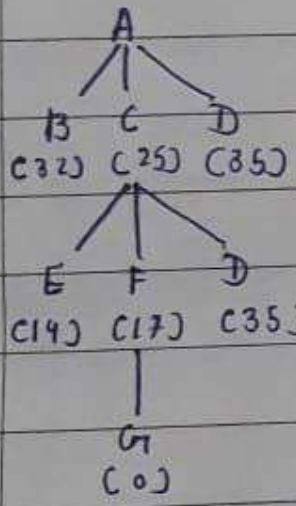
~~A~~

Best

★ ~~Beam~~ search algorithm [Informed, heuristic]



A → G = 40	E → G = 19
B → G = 32	F → G = 17
C → G = 25	H → G = 10
D → G = 35	G → G = 0



A → C → F → G

→ In the Beam search algorithm it is same as best search algorithm. You just only store n number of best node in open queue, but in best search algorithm we store all the value in open queue.

- Important Question Internal exam:

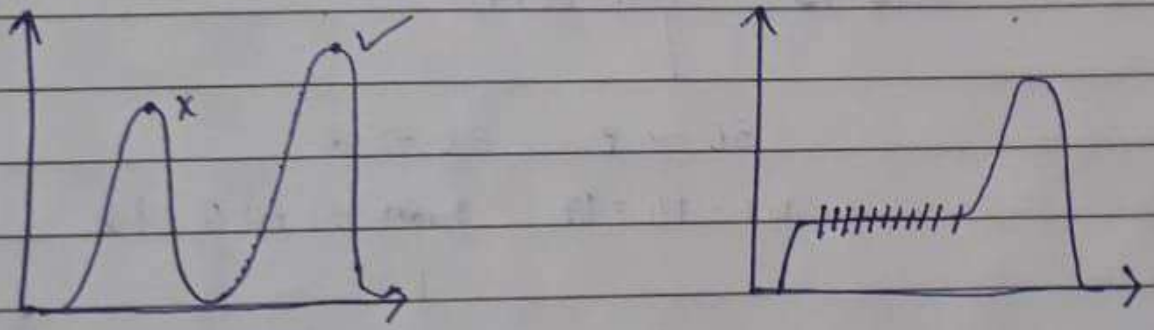
① PEAS representation of an Task environment. Explain with one example.
② Types of Agents in AI: Explain with diagram.
③ Explain A* algorithm and how to avoid loops in it.
④ Explain Hill climbing algorithm with all it's terms.
⑤ Explain Dijkstra's algorithm
⑥ Min-max algorithm & α-β algorithm

★ Hill climbing algorithm:

- Local search algorithm, Greedy approach, NO backtracking

• Problem in Hill climbing

① Local Maximum ② Plateau / flat Maximum ③ Ridge



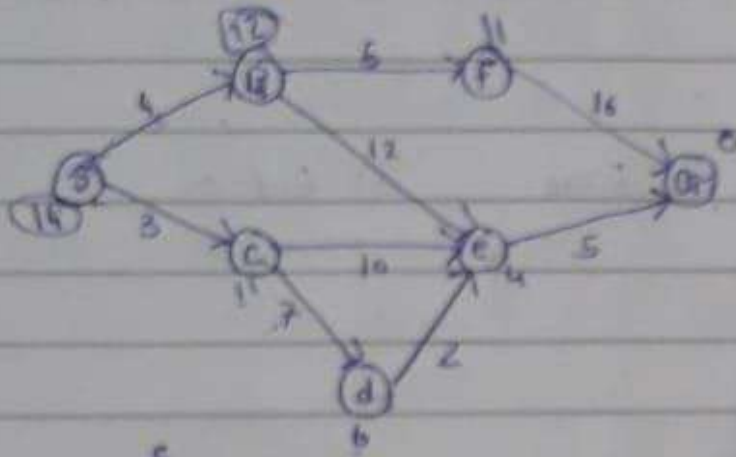
- If the β-value in beam search is 1 then it's known as a Hill climbing algorithm.

★ A^* algorithm :

- Informed search algorithm

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

where $g(n)$ = Actual cost



S → B : S → C

4+12 = 16 , 3+11 = 14

SC → e : SC → d

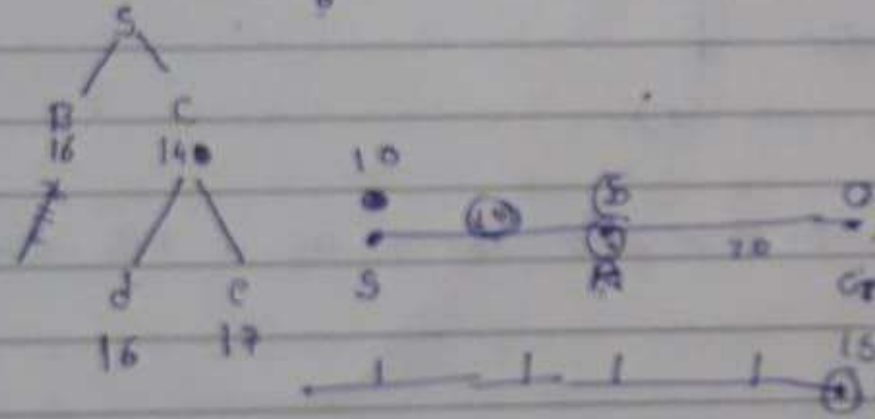
14+3+4 = 17 , 3+7+6 = 16

SCd → e

3+7+2+4 = 16

SCde → G

3+7+2+3+0 = 17



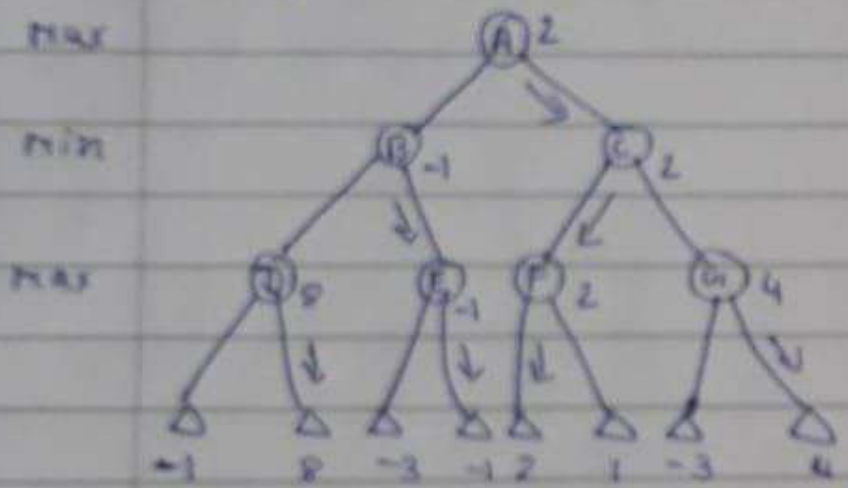
10 + 5 = 15

Final Path : S → C → d → e → G

- This will give the optimum solution for the problem.

★ Minimax algorithm :

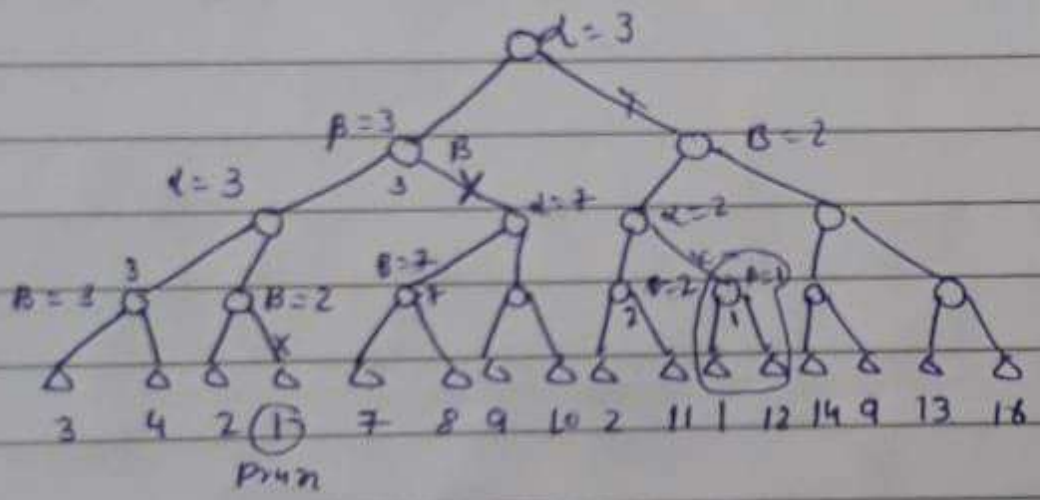
- Back tracking algorithm, Best move strategy used
- Max will try to maximize their utility and min will try to minimize their utility. (Best move) (worst move)



[Game Tree]

☆ α-β Pruning (α-β) algorithm :

- Advance version of minimax algorithm



★ A* algorithm :

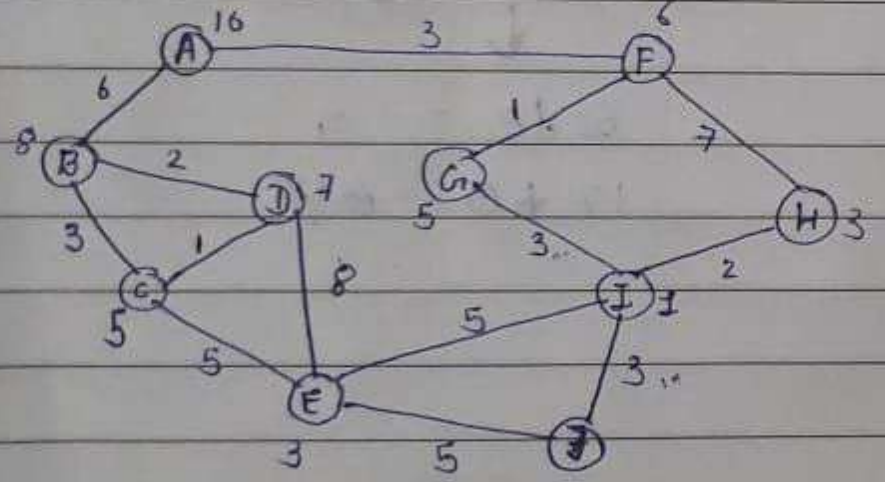
- Which is one of the best and popular techniques used for path finding and graph traversal. A lot of games are using this algorithm to find the best route efficiently. This is the best first search algorithm.
- A* algorithm extends the path that minimizes the following function...

$f(n) = g(n) + h(n)$ where n = last node on the path

$g(n)$ = cost of path from start to n

$h(n)$ = heuristic function that estimate cost from 'n' node to goal node.

- Example :



A → B | A → F

6 + 8 = 14 | 6 + 3 = 9

↓

AF → G | AF → H

4 + 5 = 9 | 10 + 3 = 13

↓

AFG → I → AFGI → J

7 + 1 = 8 | 10 + 0 = 10

Final path : A → F → G → I → J