Engineening, cognitive science, philosophy

Main Goals of AI:

1) Intenact with agents, people, envinonment.

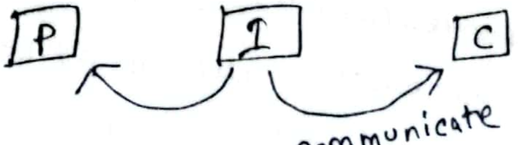
2) Behave intelligently in complex envinonments.

3) Reprnesent/stone knowledge

Foundation of AI: (8)

Math	CSE	Psy	Lingulstics
Eco	Phy	Bio	Cognitive science

Tuning Test: 3 nooms | 1 penson | 1 computen | 1 intennogaton



- tny to detenmine which is P/C.
- Computen tnies to fool
  - if success, machine can think like us.

4 Goals:

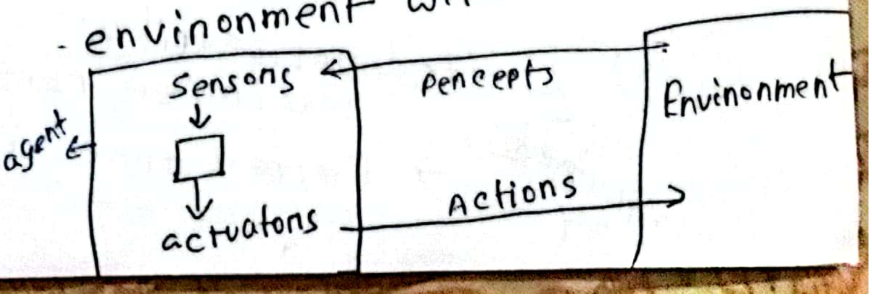
	Rationally	Like human
Think	✓	✓
ACT	✓	✓

Intelligent Agent:

- penceives its envinonment via sensons
- acts nationally upon that envinonment with its actuaton

- Autonomous
- neqtive to envinonment
- goal-dinected
- intenacts with othen agents

own behavioun determined by own expenience



Black - human
Blue - automated taxi driving system

Performance - safety
passenger satisfaction
obeying traffic rules

P - Performance Measure
E - environment
A - actuators.
S - sensors

Sensors : Eyes, ears, nose, neuromuscular system, skin / video, mic, keyboard, GPS, sonar

Percepts : electrical signals from sensor

actuators : limbs, eyes, tongue

actions : walk, run, carry object etc.
steer, accelerate, brake → actuators

Goals : maintain safety, reach destination, provide passenger comfort

Environment : pedestrians, traffic, weather, customer, urban streets

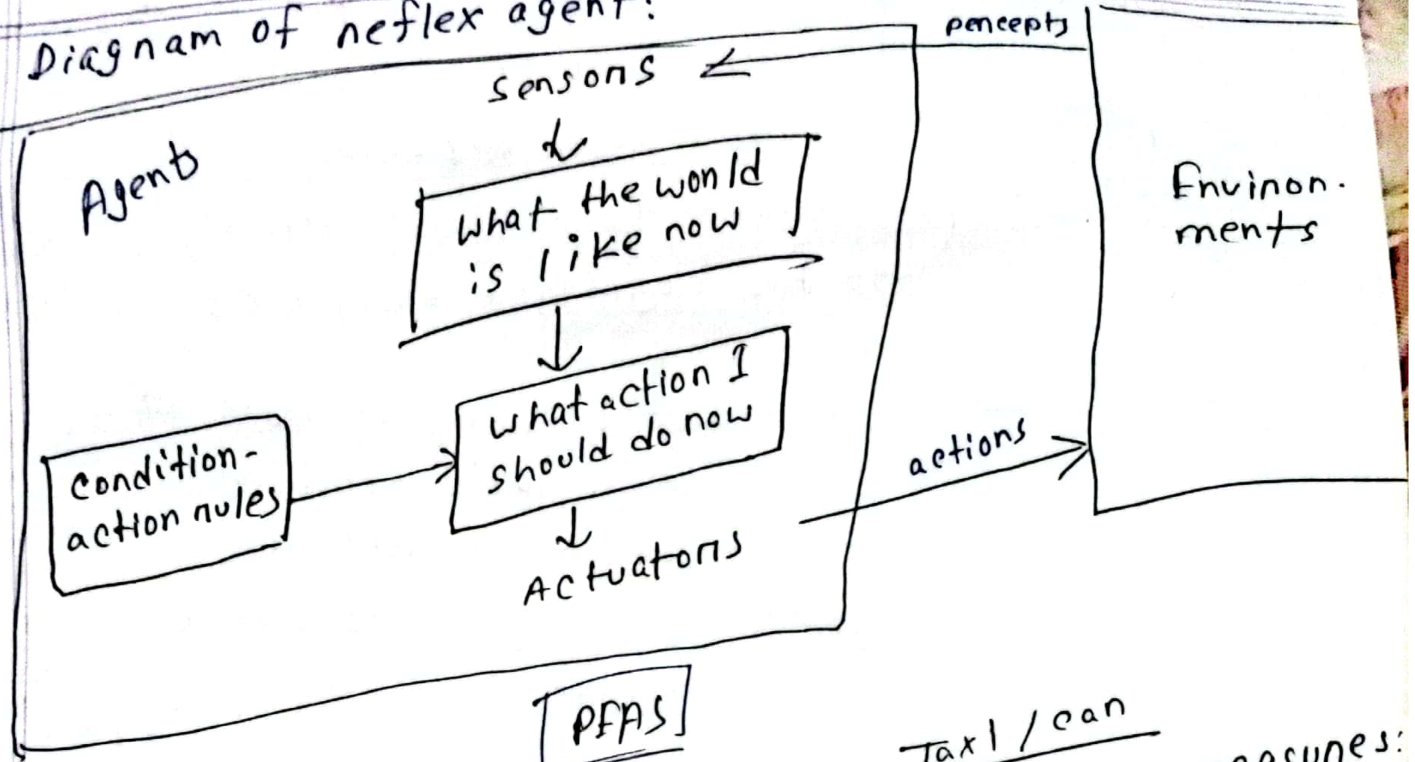
Rationality: ability to do this ← select actions that max. utility

Rational agent : max. performance measure based on
- percept sequence
- built in/acquired knowledge

Simple reflex agent:
- don't have memory of past
- stateless device
- based on condition-action rules.
- respond immediately to percepts

Limited rationality: maximize performance inspite of
constrained by → limited time, info, computational resources

Diagnam of neflex agent:



PFAS

Human / soccen playen

Penfonmance measune:

successful pass, positioning, tackle, goal sconed, assist,

Envinonment: weathen, passengen, pedestnian, tnaffic condition, noads, othen venicles

Actuaton: nunning, passing, shooting, heading, dnibbling

Sensor: skin, eyes, eans

Taxi / can

Penfonmance measunes: passengens satisfaction, obeying tnaffic nules, safety, efficiency

Envinonments: pnesence of othen playen, weathen, condition of gnound. teammate/opponent

Actuaton: bnake, accelenate, steening

Senson: camena, nadan, GPS,

Intelligence: acquine, pnocess, apply knowledge
- skills to solve pnoblem

AI: when a machine does this
- leann fnom expenience
- adapt to new situation
- has cognitive abilities

Logical neasoning: set of nules of logic to neach valid conclusions
- based on knowledge
- existing facts based
- denive new info

from

- cleaning effi
- time "
- energy consumption
- path effi

* Agent function of vaccum cleanen (foun tiles) — measure penfonmance. (4)

```
function vaccum- agent(pencent) {
  location, status = pencept
  if(status = dinty) netunn "suck"
  else if loc = A  netunn "move to B"
                              "   "  C
  else "        B                    D
                C                    A
                D
```

instructs the vaccum cleanen to clean tile, if dinty, othenwise moving next.

Characteristics of Ludo game : (7)

- fully observable
- stochastic
- sequential decision making
- finite state space
- competitive
- multi-agent
- partially cooperative and " competitive

Heuristic: estimated cost from node n to goal state

Admissible heuristic: never overestimates the true cost.

* Cost of optimal solution to a relaxed problem is an admissible heuristic for the original problem..

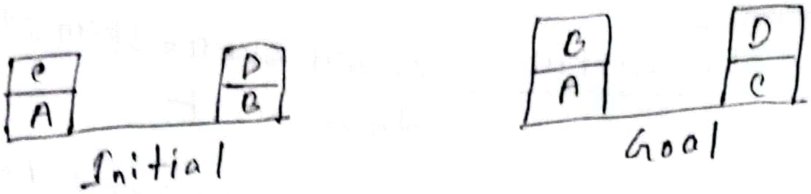
→ admissible ~~eo~~ heuristic ≤ true cost

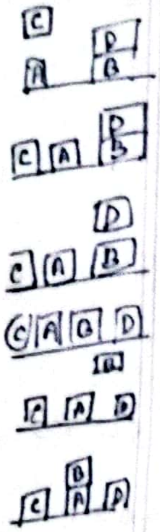
when we relax a problem
↓
loosen /remove constraints
→ solve easier
→ lower cost than original prob.

So, using this relaxed solution as heuristic ensures that we never overestimat cost to reach goal in the original problem.

knowledge-based agent and its properties.

* a robot can perform following 4 actions:
UNSTACK (x, y), STACK (x, y), PICKUP(x), PUTDOWN(x)
find solution to move the blocks from initial to goal state.





1) unstack (C, ~~stack 1~~ A)

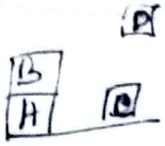
2) putdown (C)

3) unstack (D, ~~stack 2~~ B)

4) putdown (D)

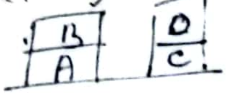
5) ~~unstack (B, stack 2)~~ pickup (B).

6) stack (B, ~~stack 1~~ A)



7) pickup (D)

8) stack (D, ~~stack 2~~ C)



~~9) pickup (D)~~

~~10) stack (D, stack 2)~~

* Assumptions about block world environment.

1) Discrete world

2) Static environment
— no external force rather agent

3) Fully observable

4) deterministic actions

5) single-agent environment

6) no block interactions
↳ pickup, pickdown, stack 1/1
↳ no sliding, rotating, stack multiple

7) No block attributes.

* Wumpus world agents

Characteristics of task environment:

1) partially observable → no complete knowledge of environment

2) stochastic → uncertain environment

3) Dynamic 4) sequential 5) multi-agent

PEAS:

Performance measure: no. of actions taken, amount of time elapsed, no. of hazards avoided

Environment: wumpus, pits, gold, walls.

Actuators: moving to adjacent rooms, shooting arrow to kill wumpus, grab gold, exit

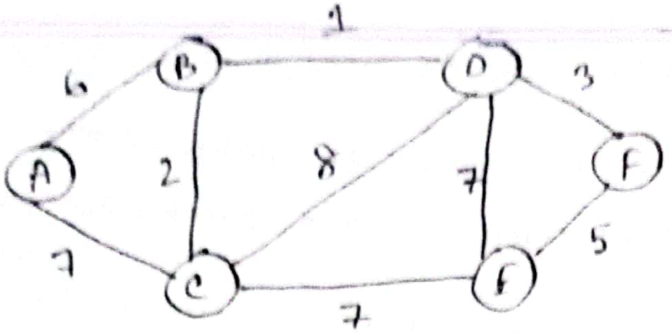
sensors: breezes indicating nearby pits
smells " presence of wumpus
glimmers " proximity of gold
bump sensors detecting collision with walls

Logical reasoning steps:

1) perception
2) knowledge representation
3) logical interference
4) goal-oriented reasoning
5) action selection

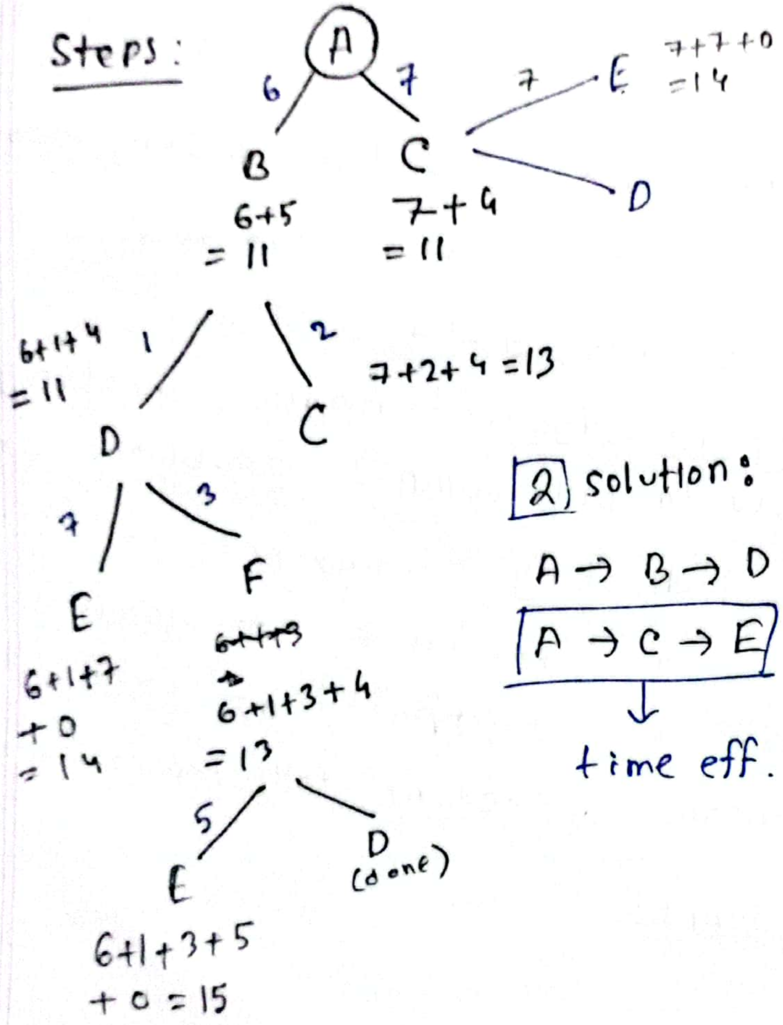
*

A* → heuristic
cost effective
time "



Goal: E
from A

Steps:



Let h(n)	
A → 8	
B → 5	
C → 4	
D → 4	F → 4
E → 0	

2 solution: (for cost effective)

A → B → D → E (14)

$\boxed{A \to C \to E}$ (14)

↓

time eff. also

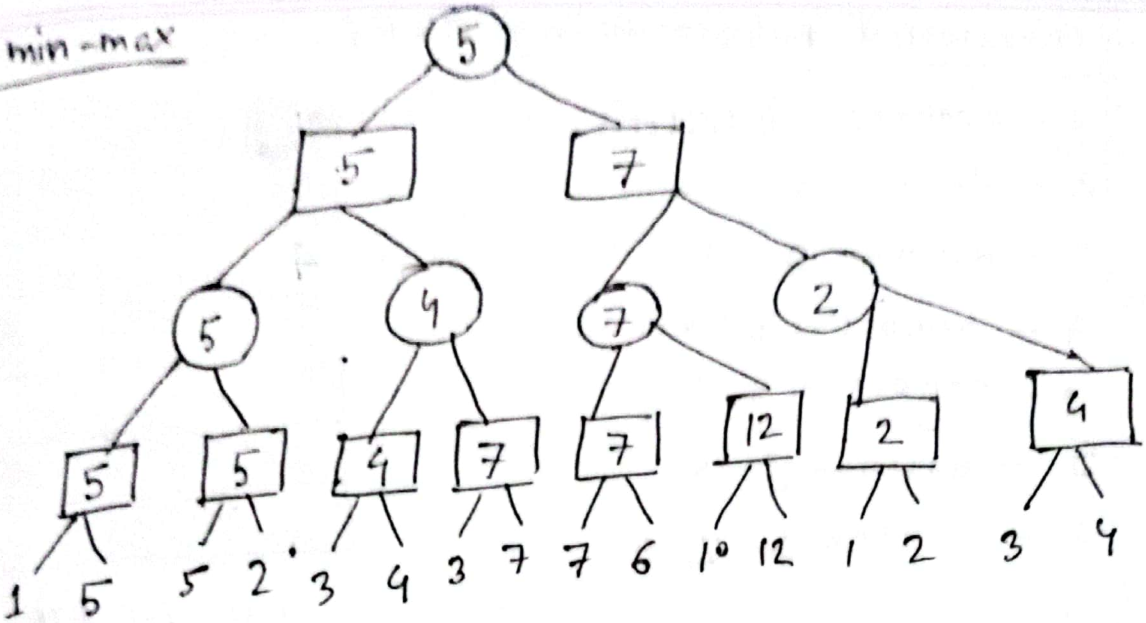
* How best-first search combines BFS with DFS?

→ explores each neighboring nodes like BFS.

→ based on heuristic score, selects the nodes that seem closer to goal and explore only that node (like DFS)

□ - max
○ - min

* min-max



* Steepest Ascent Hill Climbing

Simple Hill Climbing

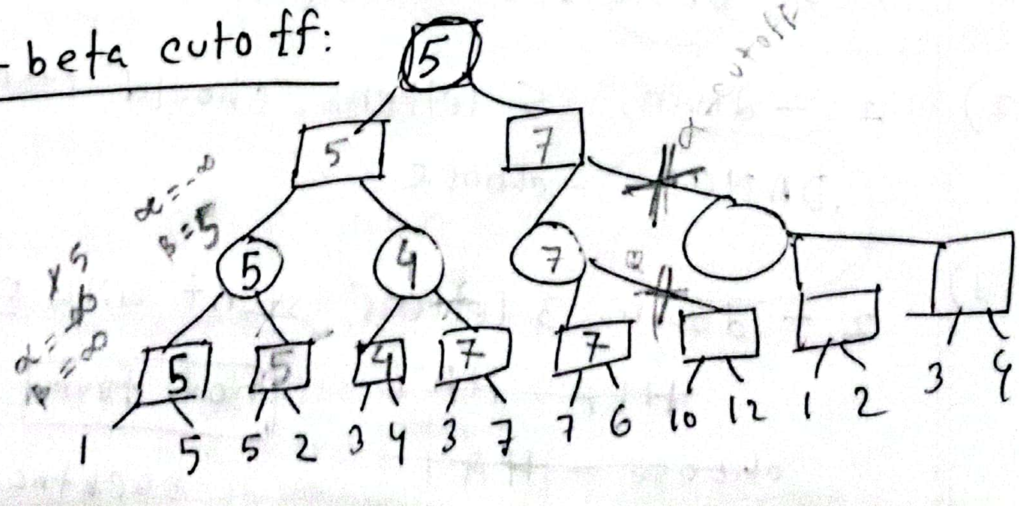
1) Evaluates all neighboning states and selects one that impnoves the objective function the most.

1) Evaluate a single neighboning state

2) Mone computation

2) Less computation

* alpha-beta cutoff:



Cnosswond puzzle using CSP:

1 - accnoss - 4 lettens.

1 - down - 5 "

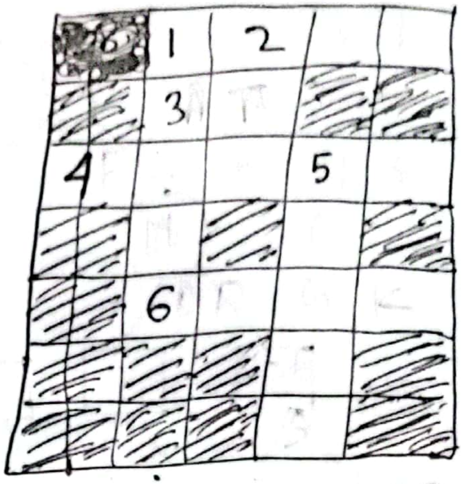
2 - down - 3 "

4 - acnoss - 5 "

6 - acnoss - 4. "

5 - down - 5 "

3 - accnoss - 2 "



wonds: AT, ETA, BE, HAT, HE, HER, IT, HI, ON, ONE, DESK, DANCE, USAGE, EASY, DOVE, FIRST, ELSE, LOSES, FUELS, HELP, HASTE, KIND, SOON, SOUND, THIS, THINK

Steps:

1) 1 - acnnos.

4 lettens - DESK X
EASY X
DOVE X
ELSE X
HELP X
KIND X
SOON
THIS

choose DESK, (if not satify, we'll backtnack).

2) 1 - down, 5 lettens, should stant with D

DANCE - choose

3) 2 - down, 3 lettens, stant with E

~~HEP, HAT~~ [not found]

~~choose - HAT~~ backtnack

Steps: 1) 1 - across - choose EASY
2) 1 - down — ~~A~~E start, 5 letters
—not found, backtrack

Steps: 1) 1 - across - DOVE
2) 1 - down — ~~O~~D start, 5 letters - DANCE

3) 2 - ~~across~~ down — O start - 3 letters - ONE

4) 4 - across - no word found
backtrack
_NE - -

Steps: 1) ELSE or ~~KEEP~~ 2) backtrack
(No word with E, 5 letters)

~~Steps: 1) KIND 2) I start, 5 — not found~~

Steps: 1) ~~SOON~~ HELP
2) H, 5 letters, HASTE
3) 2 down - ~~O~~, E, 3 let - backtrack

Steps: 1) KIND 2) K start, 5 letters - "

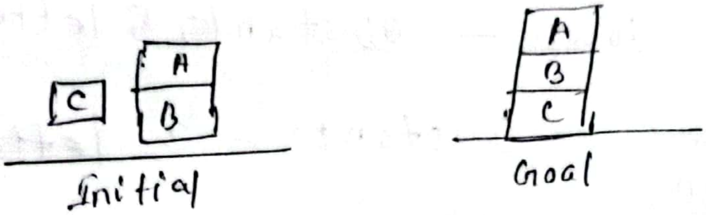
final soln.

Steps: 1) SOON 2) SOUND 3) ONE 4) 4 across - FUELS

5) 6 - across ~~DOVE~~ DESK
6) 5 - down — LOSES
↓
backtrack and change DOVE to DESK
7) 3-across - ON

* Leanning agent and stnuctune:

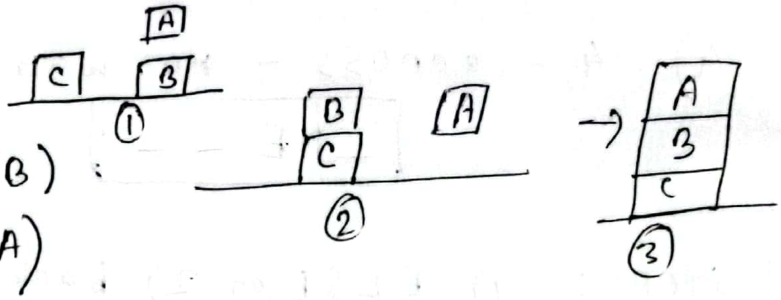
* move (x, y) , movetotable (x)



1) move (A, B)

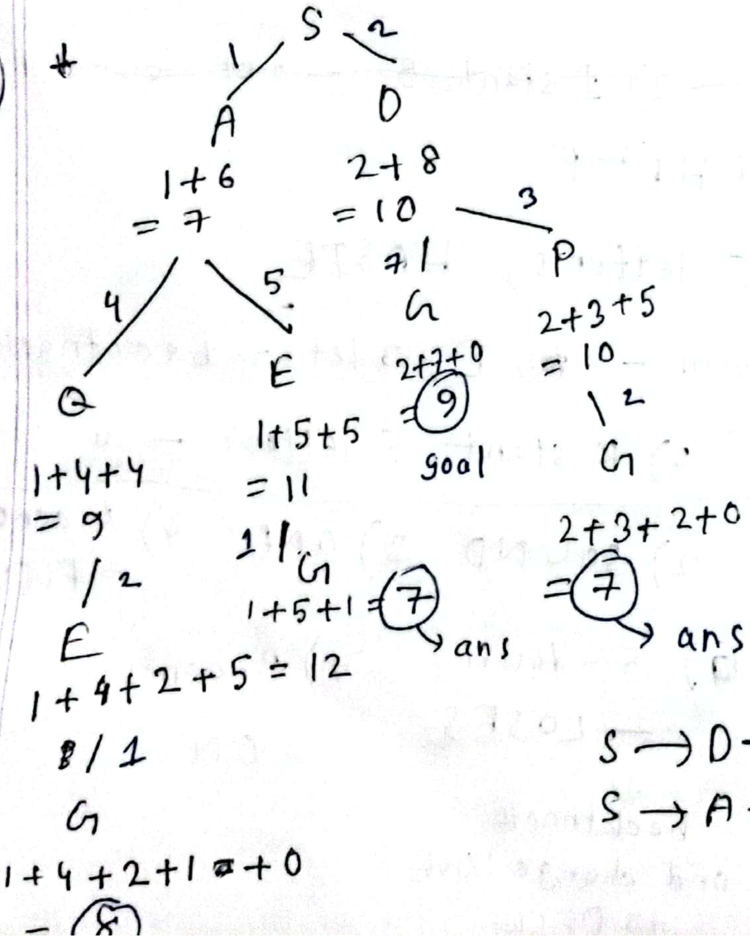
2) move to table (B) :

3) move to table (A)



A*

*



h(n)

S – 9	P – 5
A – 6	D – 8
Q – 4	
E – 5	
G – 0	

S → D → P → G

S → A → E → G

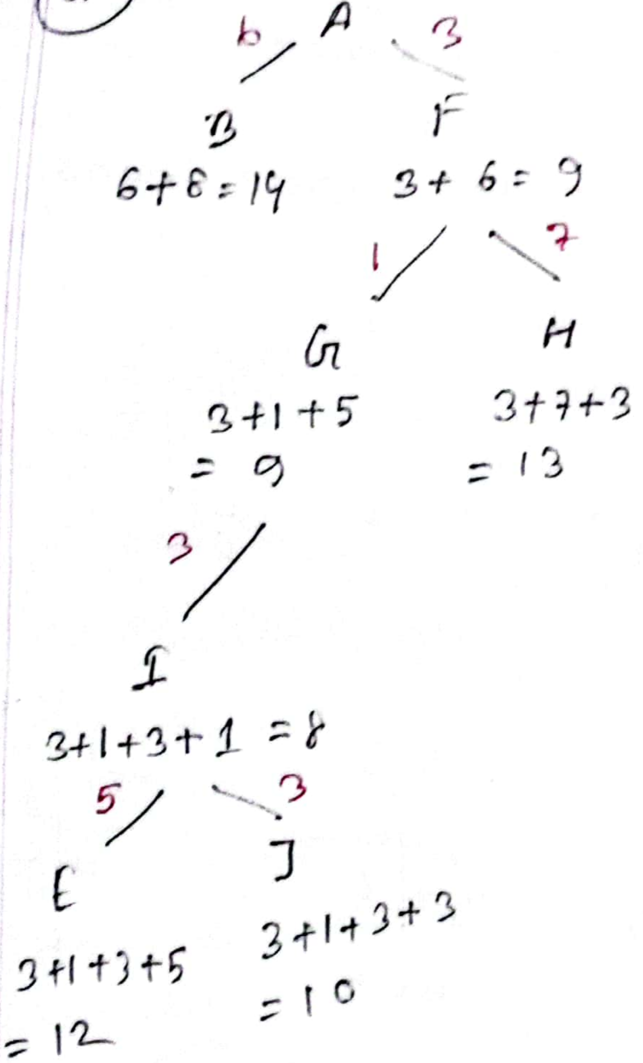
* What's game tree?

→ problem spaces for typical games

- represented as trees
- root node represents current board configuration.

ITI

6



h(n)

A — 10	F - 6
B - 8	G - 5
C - 5	H - 3
D - 7	I - 1
E - 3	J - 0

* Envinonment chanactenistics:

Poken

1) pantially obsenval~~ly~~e
2) stochastic
3) Advensial
4) sequential
5) multiagent
6) Dynamic

Image Analysis

1) fully obsenvable
2) Detenministic
3) static
4) single agent
5) non-competitive

*Goal based agent : goal oriented planning
take steps to move towards goal

*utility based — max. utility, decision making

properties of environment

1) partially or fully observable

2) Deterministic or stochastic

↓ depend on current state

↓ next state influenced by random factors

3) episodic or sequential

↓ no influence from prev. steps

8) known/unknown

4) Static / Dynamic

5) Discrete/Continuous

6) single / multi agent

7) competitive/cooperative

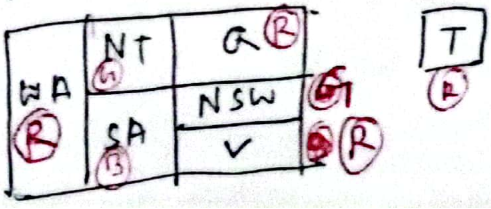
*

UCS	DLS
Time $O(b^{c^*/\varepsilon})$	Time $O(b^{l})$ - exponential
Space $O(b^{c^*/\varepsilon})$	space $O(bl)$ - polynomial
complete	Not complete
Optimal	Not optimal

* coloring with forward checking :

WA	NT	Q	NSW	V	SA	T
R ~~G~~ ~~B~~	~~R~~ G ~~B~~	R ~~G~~ ~~B~~	~~R~~ G ~~B~~	R ~~G~~ ~~B~~	~~R~~ ~~G~~ B	~~R~~ G ~~B~~

Faster method : Arc consistency



Q→G
शर्त,
NT — B
SA - no option
so, Q→R