Artificial Intelligence

AI is making machines that can think & do like humans can do.

1956 ⇒ Dart Month Workshop.

Thinking → Human Manner
Thinking → Rational Manner → Acting.

		T	A
Human	H	✓	✓
Robot	R	✓	✓

Turing Test :-

Human ← [Chatbox] → Machine

Intelligent Machine :- That can think like Human.

ANN ⇒ Artificial Neural Network

Binary Logic :- 0/1 T/F

Fuzzy Logic :- between 0 to 1 (conditional)

With same input each time, output will improve each time due to intelligent program.

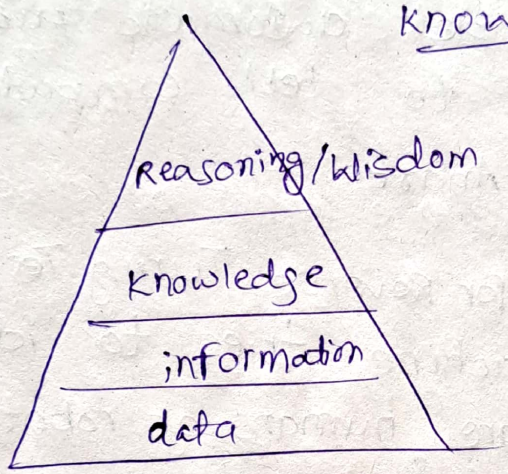
Scanned with OKEN Scanner

- Magic square / Tic-Tac-Toe / N-Queen.

- Turing Test :- A machine acting like Human.

(Text related)
- → NLP
- → Knowledge Representation. (★★)
- → Automated Reasoning (uses $\wedge, \vee, \oplus, \neg$, Truth Table)
- → machine Learning.

Knowledge Pyramid



DIKW

- Ontological tree is used for representation of knowledge.

x = Today is Thursday [True]

y = This notebook is laptop [False]

z = This sentence is True. [cannot be predicted]

[For reference :- Paradox] R

Machine domain :- Remove ambiguous statements
:- Keep only statement with Truth values

- Russell's Paradox!!

Total turing Test

- → Text, Voice, speech, video
- → + 4 skills
  - → NLP
  - → Knowledge Representation
  - → Automated Reasoning
  - → Machine Learning

CAPTCHA :- Complete automatic public Turing Test to tell computers & Humains Apart.

- → used for Reverse Turing Test where Machine tries to identify whether we are humans or robots

---

Summary :- AI is not only reverse Turing Test. It is doing tasks that human can do, but not mimicking them.

Scanned with OKEN Scanner

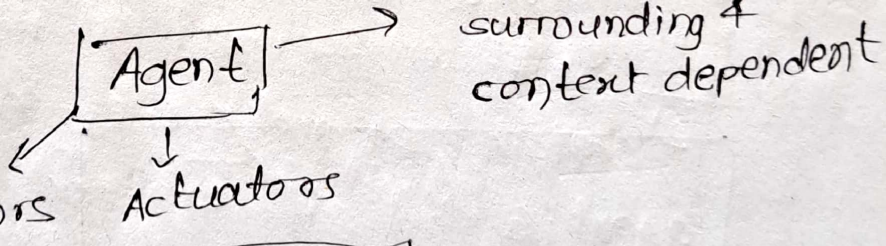
Acting Humanly

Thinking Humanly

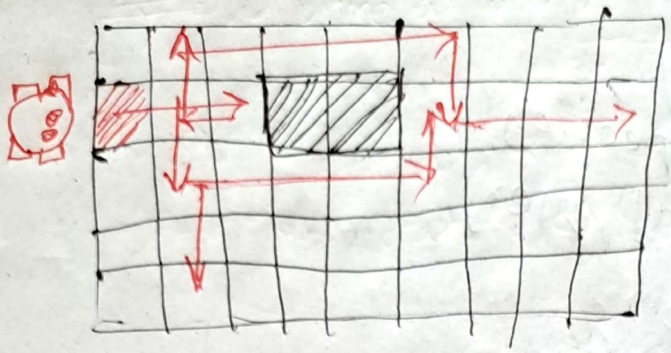
- self introspection
- psychological query
- brain mapping.

Hypothesis :- A program or machine that can perceive from its environment is called Agents. (Intelligent Agents - Matrix)

- Agent observes its environment. (Agent Smith).
- Agent is possibly a robot.
- Robot/Agent will also have computer vision.
- sensors / Actuators.



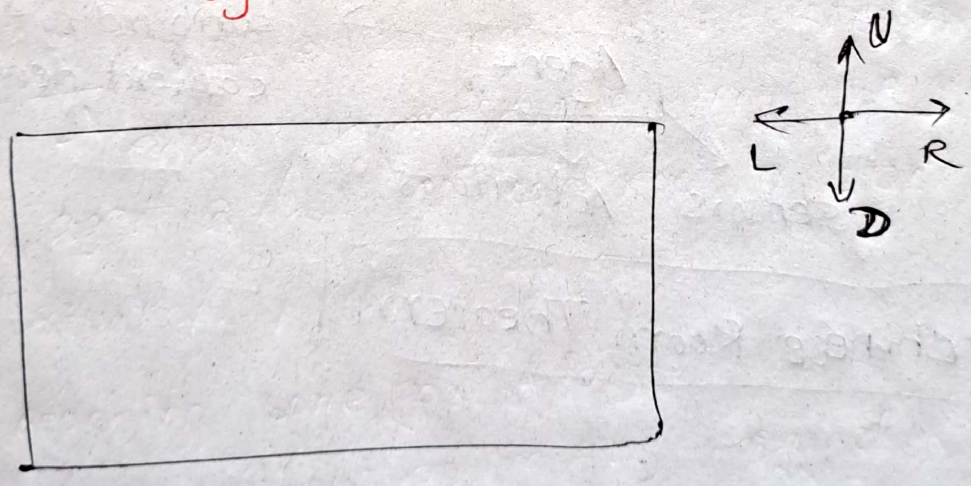
Chinese Room Theorem



- city block distance (manhatten dist)
- chessboard distance ~~(manhatten distance)~~
- Euclidean Distance
  - ★ we teach robot to move in steps
  - ★ hence distance must be discrete whole nos

- Agent will record its moves.
- sequence will be called as percept sequence

set of actions made by agent



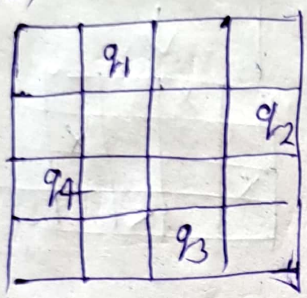
Percept sequence :- moves made by agent in a sequence

:- Percept sequence will be used to Learn

4-Queens problem

Water jug problem :- 3 li, 5 li, ⇄ 4 li

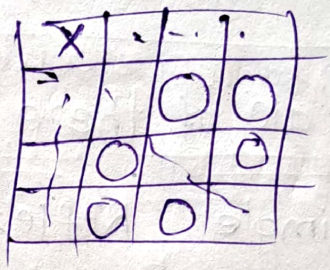
(0,0)



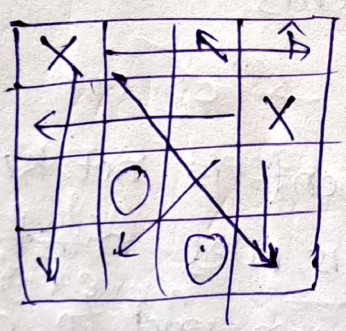
(3,3)

Solution

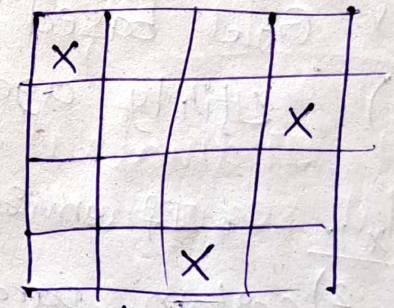
- give in form of steps
- place first queen at (0,0).
- mention rules to agent



- Place in any of the circles



→



↓

no space for 4th queen

↳ Rollback

Steps

place at (0,0)

find next position.

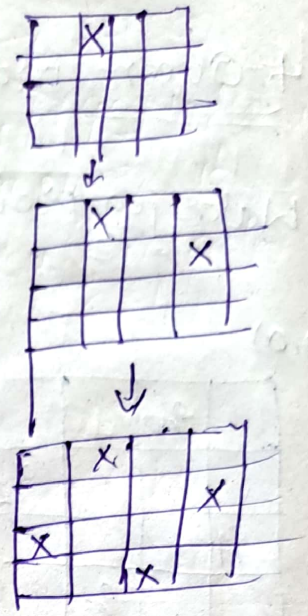
place at (1,3)

place at

⋮

Rollback to previous & previous steps if not possible.

Again
- place at (0,1)
- place at (1,3)
- place at (2,0)
- place at (3,2)



Agents are of these types:-

1) Simple Reflex Agents ex:- automatic car.
2) Model based ~~Agents~~ reflex agent.
3) Goal based agents.
4) Utility agents

ex:- • If environment is not known.
2) certain parameters, rules must be given to agent & that it can react to current situation.

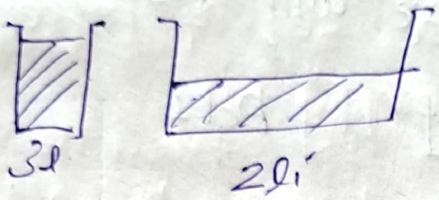
3) • Reaching some goal & target without optimizing parameters.

4) Goal + extra condition (optimisation)

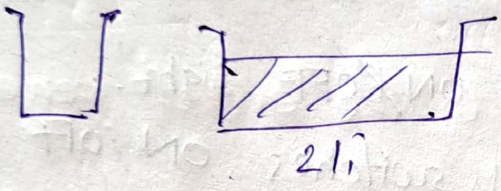
Scanned with OKEN Scanner

Water Jug problem (Sol 1)

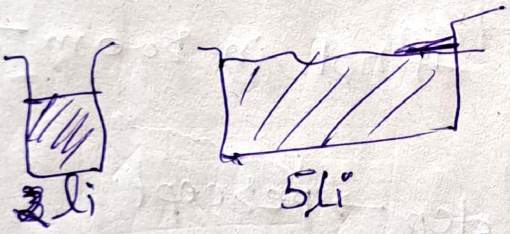
- Fill 5li vessel.
- Pour 5li in 3li



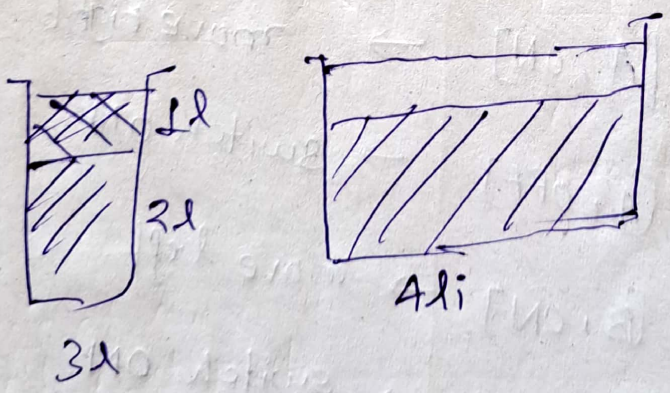
- Empty 3l.



- Pour in 3l & Fully fill 5li



- Pour 5li in 3li



Sol 2

	5l	3l
Step1	0	0
Step2	0	3
Step3	3	0
Step4	5	0
Step5	2	3
Step6	2	0
Step7	0	2
Step8	5	2
Step9	$\boxed{4}$	3

Scanned with OKEN Scanner

Percept sequence of an Agent → sensor → actuators

- As agent does its steps, & remember it.
- Percept sequence will result in action

[ ] Act 1

experience ↙ [ ] Act 2.

---

ex:- Agent to switch ON/OFF light.

- An agent which switches ON/OFF light.

A	B

If we are in room → ON

If not in room → OFF

Individual step → percept

Total sequence → percept sequence

Percept	Action
[A, ON]	→ move right
[A, OFF]	→ switch ON
[B, ON]	→ move left.
[B, OFF]	→ switch ON.

∴ percept sequences:

[A, ON], [B, OFF], [B, ON], [A, ON], [B, ON], etc.

*Exam question

Scanned with OKEN Scanner

- Read about envisonment
- Find valid actions using sufficient hypothesis

Agent = Architecture + Program

- AI: modern approach Deepak adani

Books.

- Peter Russel & Norwick :- AI : ✓
- Rich & Kevin night (1991)
- AI for dummies. (Head First)

After unboxing agents

- It stores a step a first step in its table.
- we can use any data structure for table.

~~Real life example~~

Real life Examples :-

Function { Table Driven Ag (percept) returns an action.

persistent : percepts - a seq. of percepts, initially empty.

table - a table of actions, indexed by percept sequence

append percept to the end of percepts.

action ← Look Up (percepts, table)

return action

}

End

Scanned with OKEN Scanner

Camel & banana problem

Person
Camel

$\longrightarrow$ 1000 km $\longrightarrow$ Dest.

3000 banana

1 banana / km

- If person moves with camel, after 1 km, 1 banana req.
- Camel at most 1000 banana.

Fun Simple Ref Ag (percept).

persistent rules - a set of if-then rules

State ← Interpret Input (percept)

rule ← Rule match. (state, rules)

action ← rule. Action.

return action

End

Q] Agents learn methods——?

reinforcement learning (RL)

ans- rewards & punishments are awarded.

Problem Solving	State Space Search
Goal based Agent	P Sol. Stop Goal state.

Scanned with OKEN Scanner

Einstein - we cannot solve problem with the same thinking that created them.

- State space search questions will be solved using State Space search Tree.
- These methods are known as uninformed search
- We try connecting initial state to final state

Problem solving will have 4 things:-

1) Define the problem
2) Analyze the problem.
3) Isolate & represent the knowledge.
4) Choose the best Tech.

- ~~define~~ problem ~~as statement~~

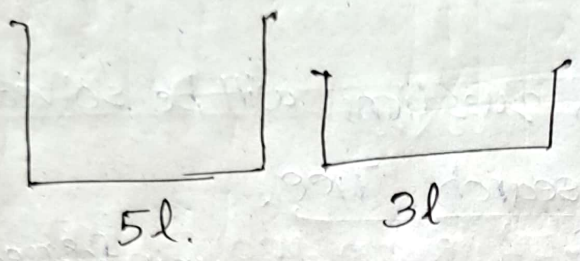
~~analyze~~

5 components to define problem

- initial state
- description of actions.
- transition model.
- Goal Test
- Path cost estimation.

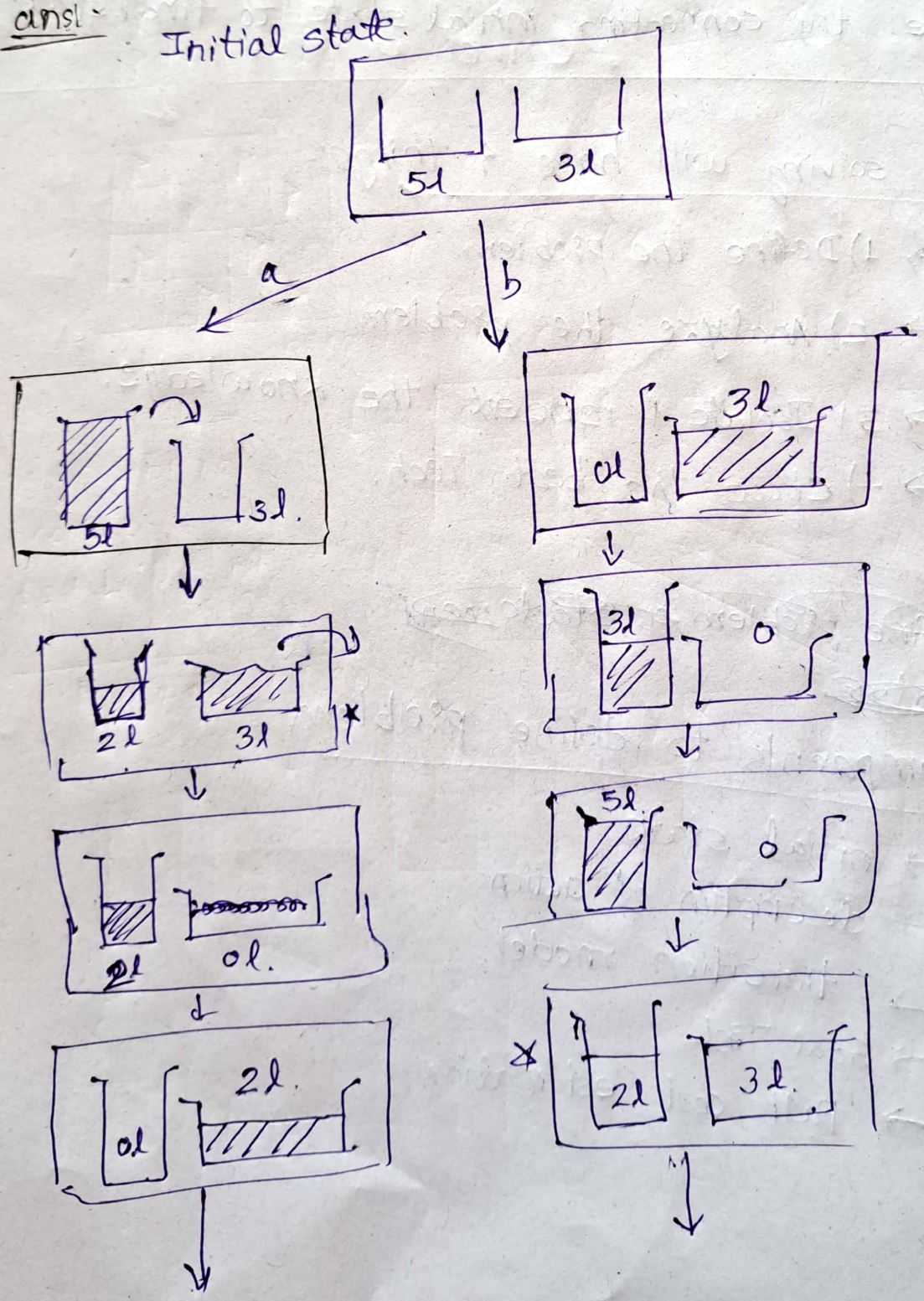
Scanned with OKEN Scanner

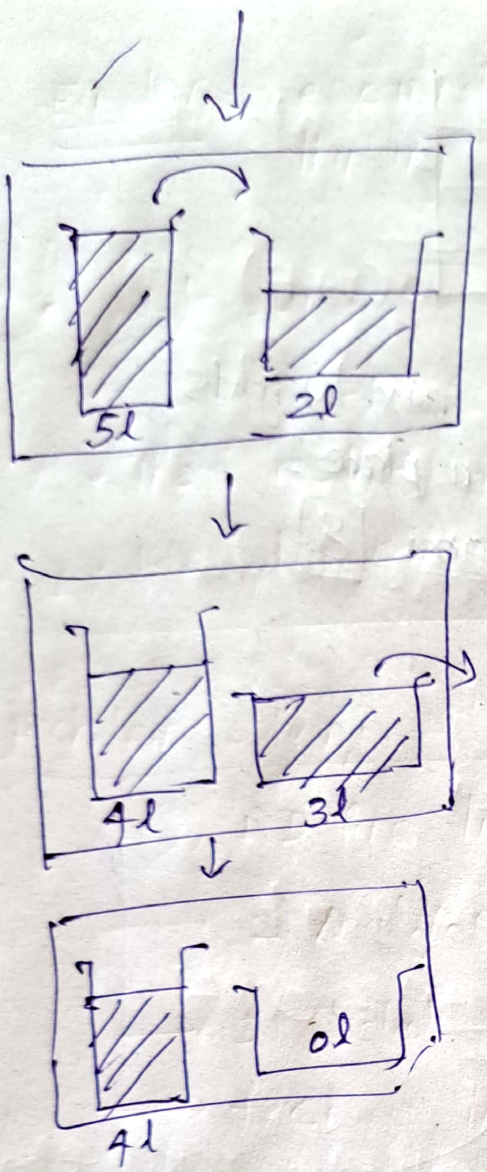
ex:- Water-jug problem



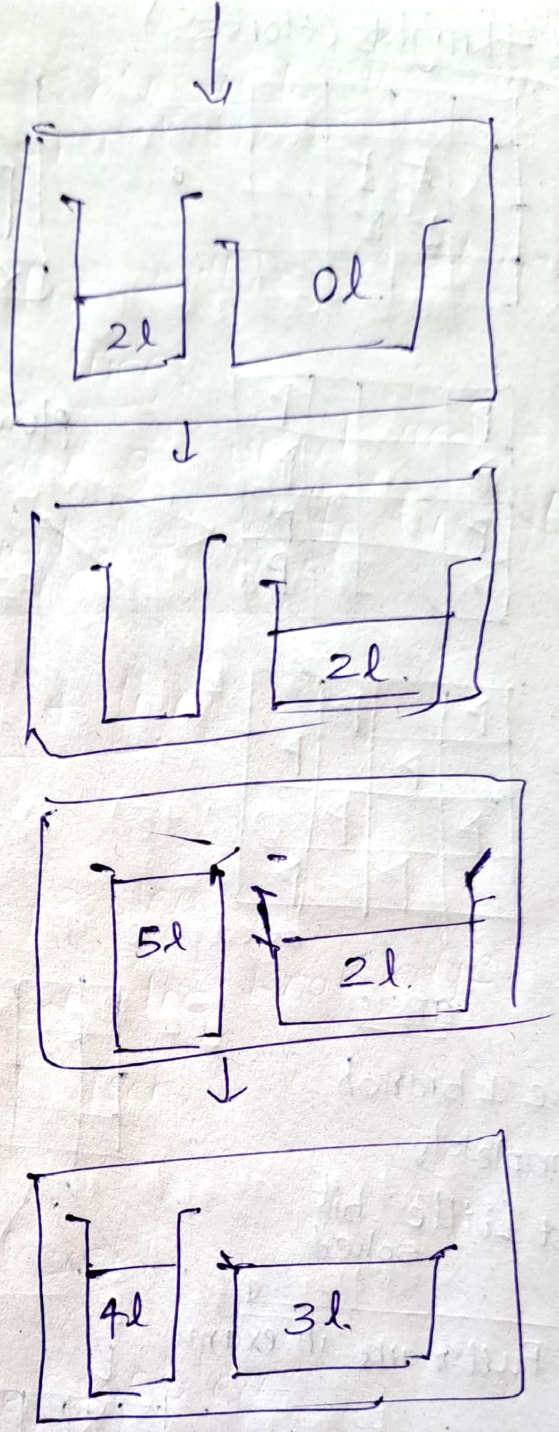
Assumptions
- infinite water
- u can waste water
- No measurement marks on container.

ans:- Initial state.





7 steps.

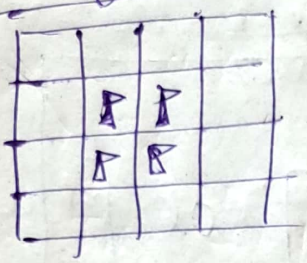


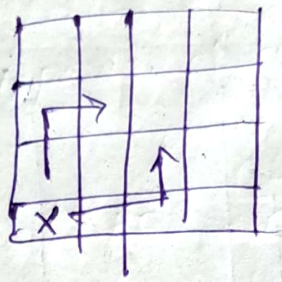
8 steps.

path a > path b (less steps)
(less water wastage)

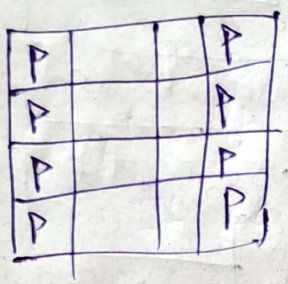
Scanned with OKEN Scanner

4X 4knights (Horses)



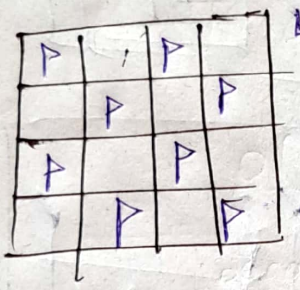


or



How many max. Knights can u place?

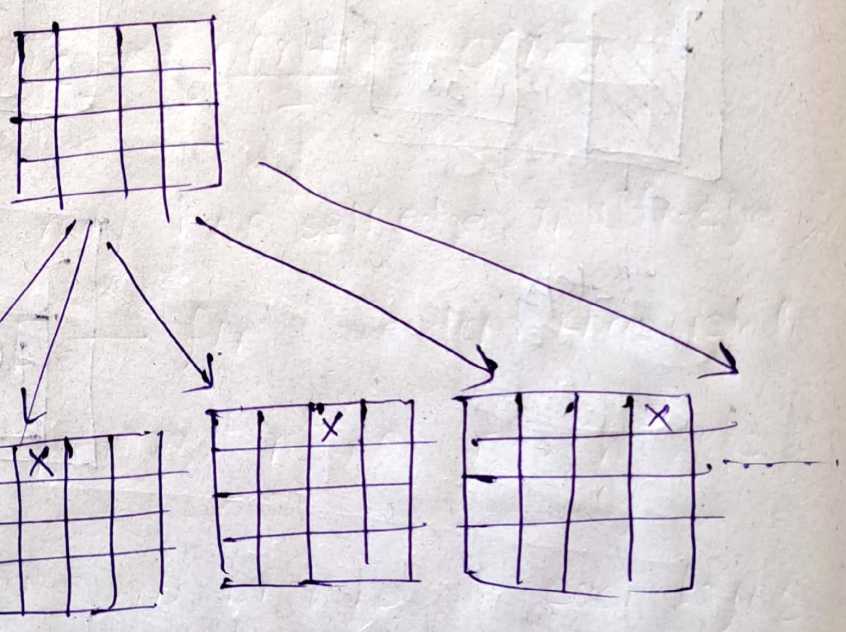
ans:- $\boxed{8}$

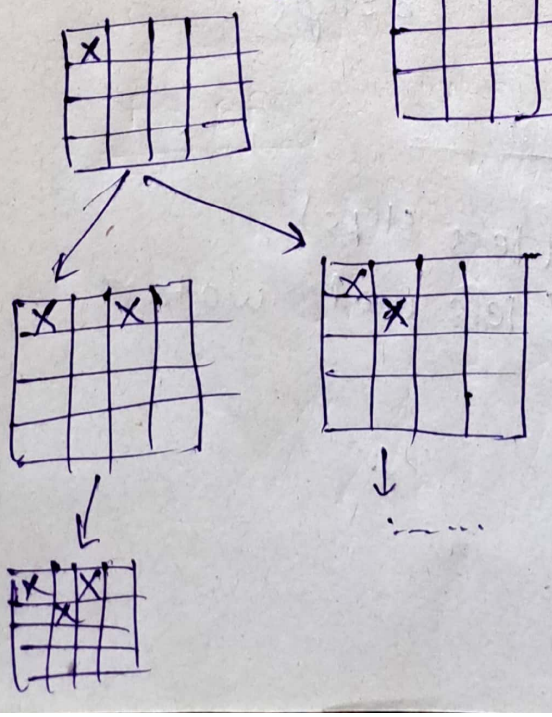


ans:-

- solve 1 branch completely
- Rest Little bit solved

Full marks in exam





Q] Camel Banana problem :-

3000 km, 1 banana per km.

Russel-Norwick Book

Formulate - Search - Execute Model :-

- These models are tested on completeness & optimality (Time & space).

(majority Time complexity)

```
Search ( )
{
    OPEN ← {START}
    while OPEN != Empty.
    do Pick a node 'n' ∈ OPEN
        OPEN ← OPEN \ {n}.
    if Goal Test (n) = True Return n.
    else OPEN ← OPEN ∪ Move Gen (n)

return FAIL.
}
```

* OPEN is bag data structure (like set).

Move Gen :- means generate child nodes.

Scanned with OKEN Scanner

Sir will ask Time complexity of any algorithm in Exam.

ex:- Time complexity of BFS:- $O(V+E)$

Q) Graph with n nodes $G(V,E)$.
If we apply BFS, what will be Time complexity.

ans:-

$O(V+E)$ or $O(n^2)$

because in complete graph,

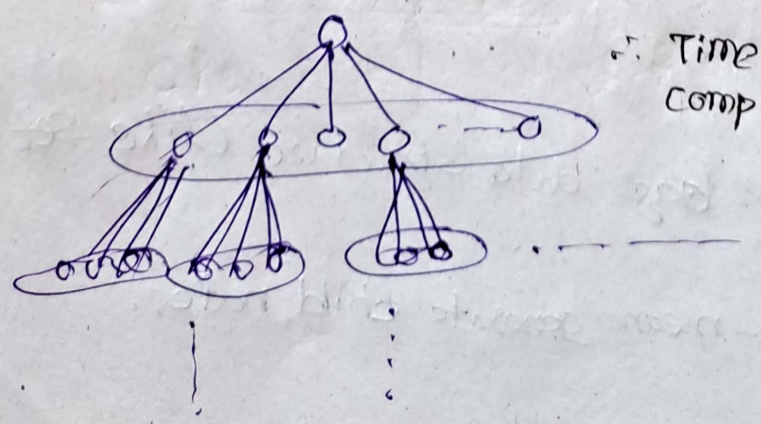
$$E = {}^nC_2 = \frac{n(n+1)}{2} = O(n^2)$$

But a graph may have self edges, multi-edges

Hence, E — $f(n)$ we don't know relation

$E \approx n^2$ or n^3, or n^e, or $n \log n$, etc?

For BFS, Take 'b' branches at each layer and 'd' depth of tree.



$\therefore$ Time complexity $= O(b^d)$

Q] Is BFS complete?

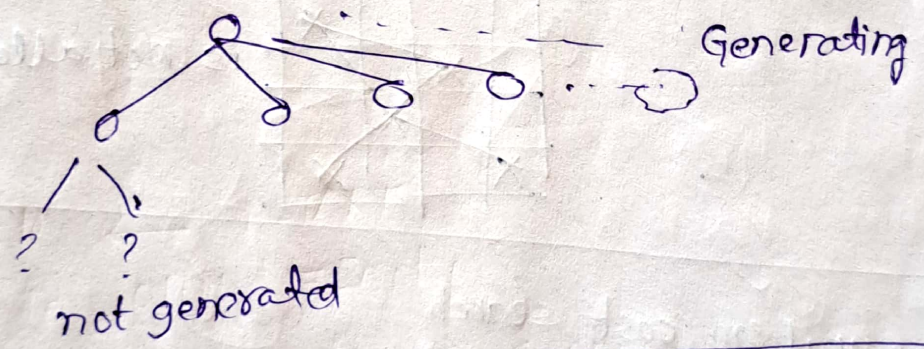
ans:- Yes, it will give whole solutions if it exists

∵ Algorithm returns if goal state Reached.

- If goal does not exist, it explores all paths.

but if b or d is infinite.
Then solution will never be found.

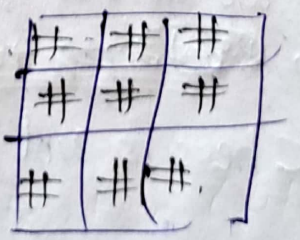
Note:- ~~We will not generate all~~
BFS theoritically generates each node & moves down.



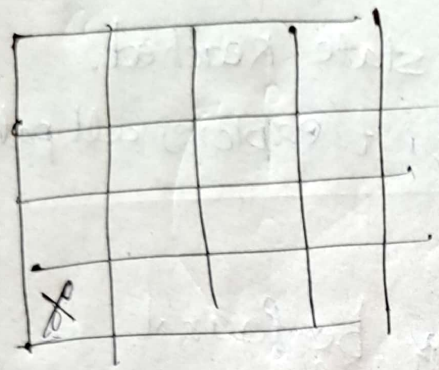
- We find a instant solution, that is not generalised. Humans do like this.
- But machines generalise it for any 'n'.

search $(\text{Tic Tac Toe})^2$

Reverse Tic Toe Toe.



Knight's move / Tour



ver 1

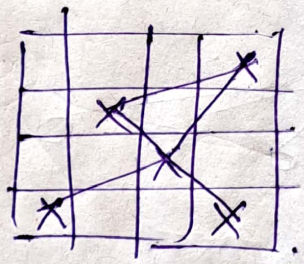
ver 2

- cover whole grid, with Knight's step.

cannot be solved for , 2x2, 3x3.

ver 1 • Knight has to visit whole board, without repeating any cell.

ver 2 The lines should not intersect.



is not allowed

- uniform cost search
- Depth first search
- Depth bounded / Limited search
- Best fit search.

Facts :- we do not know solution.

~~Problems in chess~~ :-

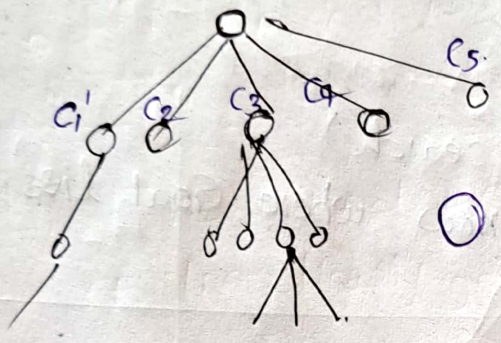
Chess is not solved problem.

Uniform cost search :-

state space search

BFS :- $O(b^d)$

Dijkstra's Algo :- with cost optimized comparison

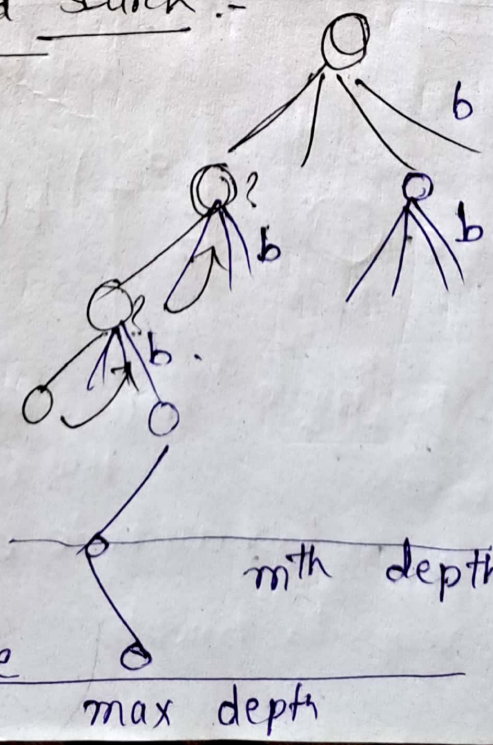


$$O\left(b^{\left(1+\lfloor c^*/\epsilon \rfloor\right)}\right)$$

- BFS is specific case of uniform cost search
- If cost of each search is Equal, Dijkstra's Algo becomes $O(b^{d+1})$

c^* - optimum cost.

Goal Test based search :-



b^2 branches

Time complexity:

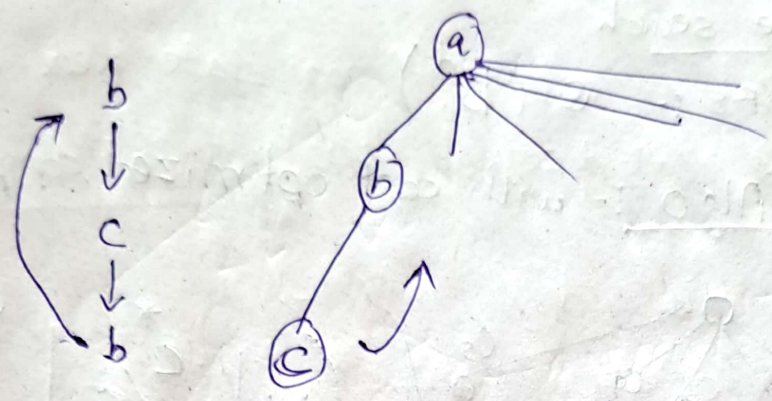
$O(b^m)$

worst case

m^{th} depth :- b^m branches.

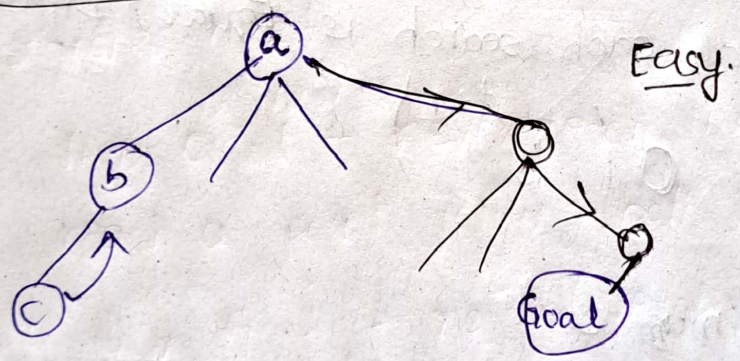
worst case max depth :- b^d branches.

DFS can have infinite loop problem



- uninformative (blind search)
  we do not know where Goal lies in Tree

Informed Search (Heuristics Search) :-



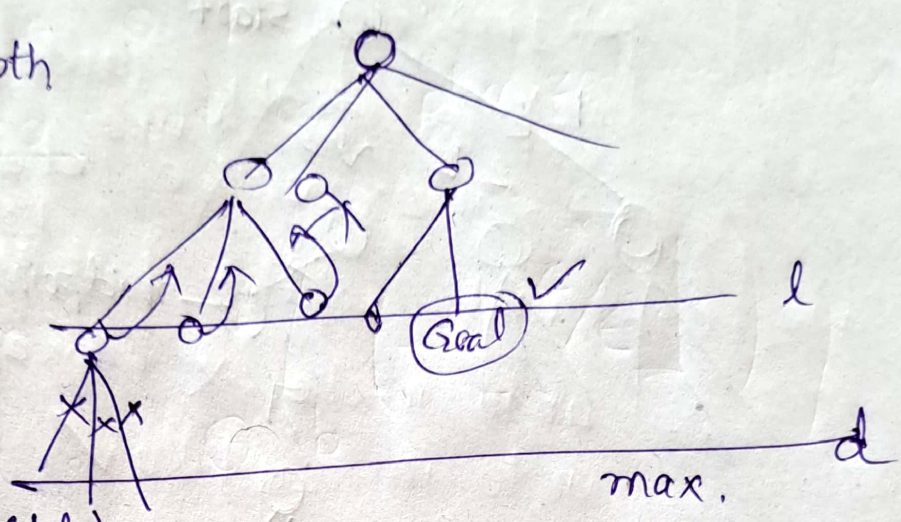
Easy.

ex:- Google maps changes Route based on Traffic

Depth Limited search :-

If goal is at depth 'l'.

- Go till 'l' depth
- back track after it.



- Time complexity :- $O(b^l)$
- Space complexity :- $O(bl)$

If $l=d$ or $l=\infty$, then DLS = DFS.

Note :- Space complexity in DFS :- $O(bm)$

Iterative Deepening :-

- Apply Depth Limited search sequentially

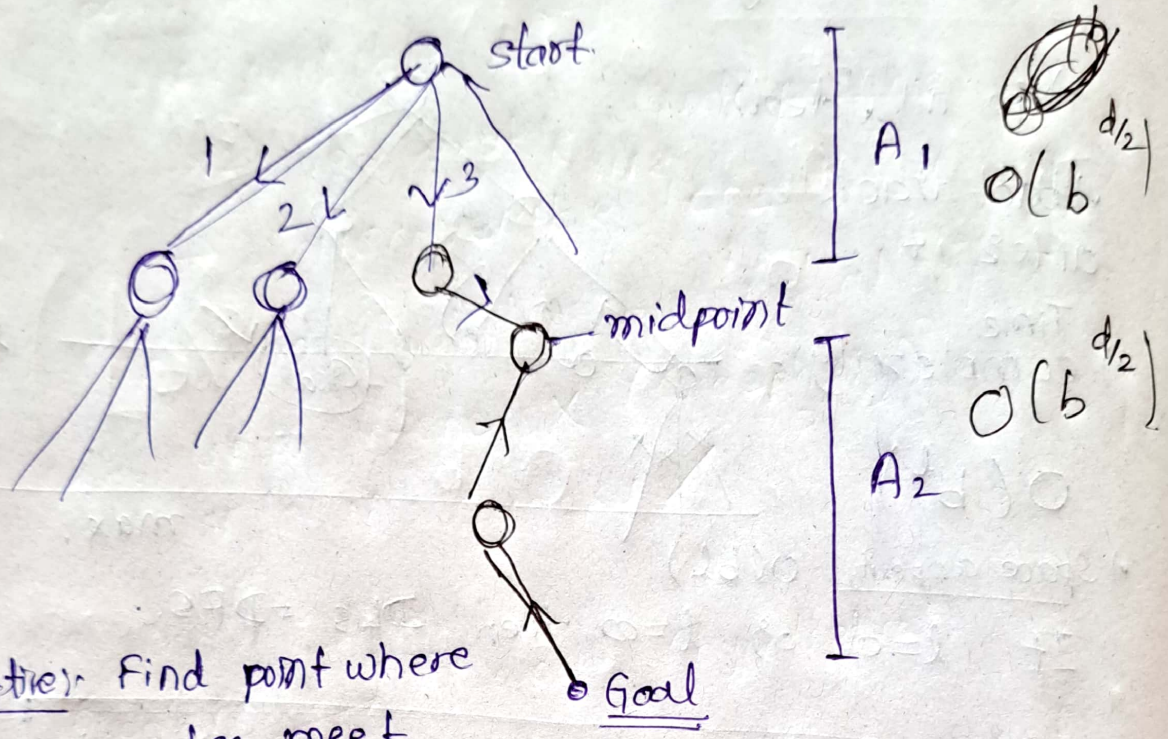
Let $d=0$, find all nodes.
$d=1$, explore nodes, Look for Goal.
$d=2$, explore nodes, Look for Goal
⋮
If Goal found, backtrack all nodes

This is restricted form of DFS & BFS.
⋮
If $d=d$ ⇒ Depth first search.

Time complexity :- $O(b^d)$

Scanned with OKEN Scanner

Bi-directional search (BDS)



objective: Find point where both searches meet

- Start from beginning & never reach the Goal, but reach a point joining both searches
- Start from Goal & never reach beginning but reach the point joining both searches

Time complexity will be halved

$$\text{Time complexity} = O(b^{d/2}) + O(b^{d/2})$$

problem:- How to move from Goal to its predecessors. H.W (saturday Lab - monday Timetable)

You will have to visit all nodes once to remember & map the parents

	complete	optimal
BFS:-	✓ traverse all nodes & find Goal	✓ Yes, if goal is at less depth, we will have more to more depth for other goals. We will find first goal & return
Uniform cost search :-	✓	✓
DFS :-	✗ • if Loop trap. never find Goal	✗ • If goal not found, what about optimisation.
DLS :-	✗	✓
BDS :-	✓	✓
Iterative deepening	✓	✓

Russel & Norwick

Scanned with OKEN Scanner

Can solution steps be undone?

✓ (ignore solution)

3 possibilities

1) ignorable
2) recoverable (undo possible)
3) irrecoverable (chess)

we can undo solution, but not always

ex: undo in chess ✗ (Gameplaying.)
undo in Sudoku ✓

Problems → Toy world (hypothetical)
→ Real world (Robotic movement)

1	4	7	2	5	8	3	6	9
2	5	8	3	6	9	1	4	7
3	6	9	1	4	7	2	5	8
4	7	1	5	8	2	6	9	3
5	8	2	6	9	3	4	7	1
6	9	3	4	7	1	5	8	2
7	1	4	8	2	5	9	3	6
8	2	5	9	3	6	7	1	4
9	3	6	7	1	4	8	2	5

rotations

3 ~~not~~ unit ahead.

Scanned with OKEN Scanner

Exam How magic square can solve Tic-Toe?

ans:- Heuristics:-

certain problems can be transformed from one domain to another.

- Problem with Tree algo for Tic-Tac-Toe:-

  combinatorial Explosion.

Heuristic functions will help in this.

~~magic square~~

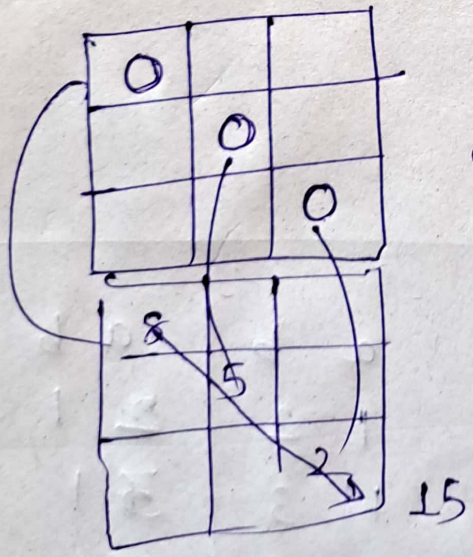
1	2	3
4	5	6
7	8	9

00	01	02
10	11	12
20	21	22

location grid

Transformed magic square :-

8	3	4
1	5	9
6	7	2



any winning position will give sum 15

ex:-

O (8)	O (3)	O (4 = 15)
	X	
	X	X

a) First move - decide nothing

b) second move - some winning position.

c) 3rd move - winning position becomes clear.

ie if (a+b+c) = 15 ✓ win.

Scanned with OKEN Scanner

if $\left(\begin{array}{c} 15-(a+b) \leq 0 \\ \text{or} \\ 15-(a+b) > 9 \end{array} \right)$

ex:-

O		X
X		O

$8+9=17$

$15-17=-2$

$1+4=5$

$15-5=10>9$.

a, b are not in straight line.

8	3	4
1	5	9
6	7	2

ex:-

else a, b in straight line

O	O	
X	X	

$6+7=13$

$15-13=2$

$8+3=11$

$15-11=4$.

Not straight line positions:-

a	b	a b
9	8	8, 9
9	7	7, 9
9	6	6, 9
8	7	7, 8

a	b	a	b
1	2	2	1
1	3	3	1
1	4	4	1
2	3	3	2

Scanned with OKEN Scanner

Lets play Game

1	2	3	4	5

O :- 8, 6, [ ]

X :- 5, ,

O		
	X	
O		

If both players use same strategy, They will also block other one.

8+6+1 = 15.

X will block O.

O		
X	X	
O		

O will block X

1+5+9 = 15

O		
X	X	O
O		

X will try another combination.

O	X	
X	X	O
O		

O will block X.

O	X	
X	X	O
O	O	

Scanned with OKEN Scanner

O	X	
X	X	O
O	O	X

X will block O.

O	X	O
X	X	O
O	O	X

1 place remaining,
no willing position
place O.

Dual. ★ we will use magic square as Heuristic func.

Scanned with OKEN Scanner

Heuristic - extra info will be given.

Heuristic function will be given to us.

Best first search

↓

A^*

↓ utilises this.

$f(n) = g(n) + h(n)$ → Heuristic function

↳ cost fun

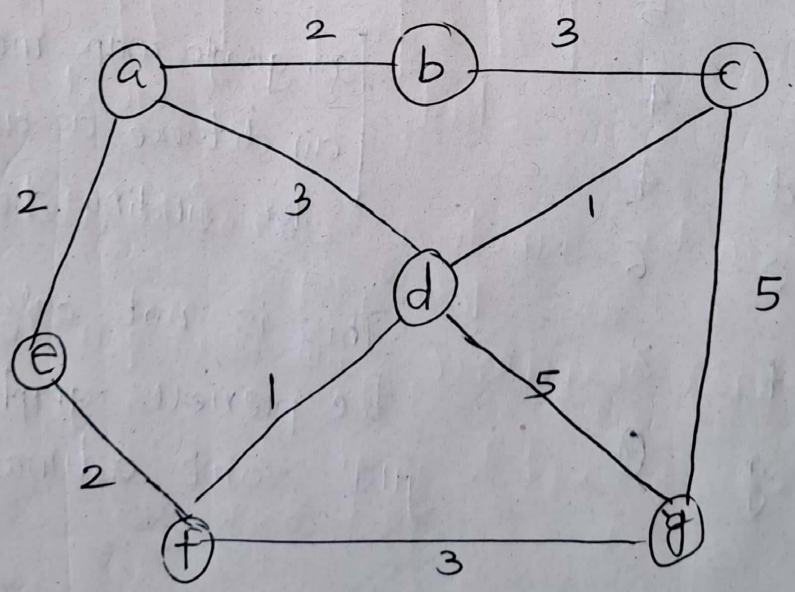
- Heuristic is better than blind search

- They are trying to estimate 'distance' of goal.

Heuristic:- an evaluation function used to estimate the cost of reach goal state.

Find shortest path :-

source: a
goal: g.



- Dijkstra will give only 1 path.
- Heuristics is not specified.

Scanned with OKEN Scanner

$g(n)$

$g(e) = 2$

$g(a) = ?$

$g(b) = 2$

$g(d = 5.$

cost function

What is heuristic func here?

We had not provided,

we will provide now:-

estimated distance of goal from these node

[this is not exact distance from goal,

else $h(a)$ will be the answer.]

a	7
b	5
c	4
d	4
e	6
f	3
g	0

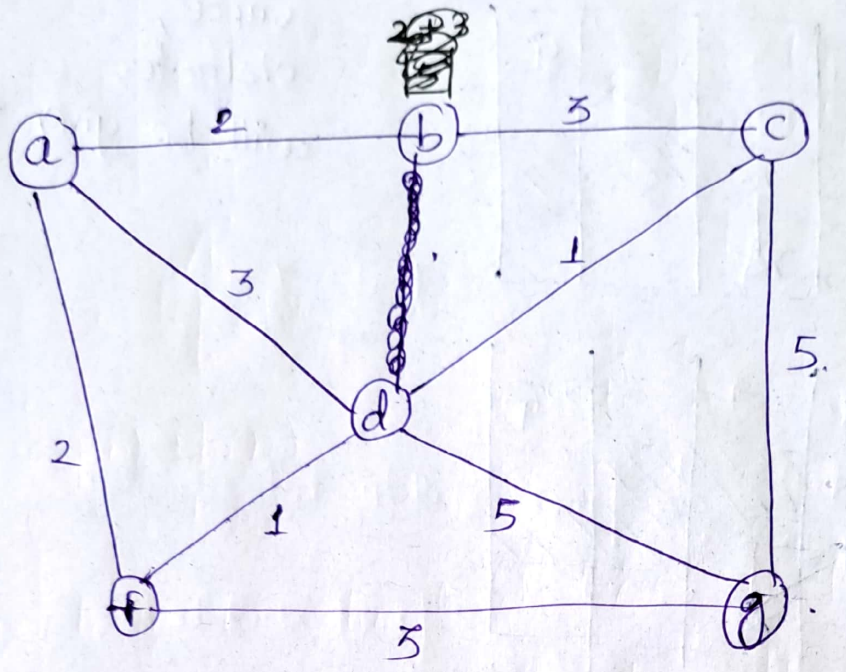
This can be treated as air distance / Euclidean dist, etc.

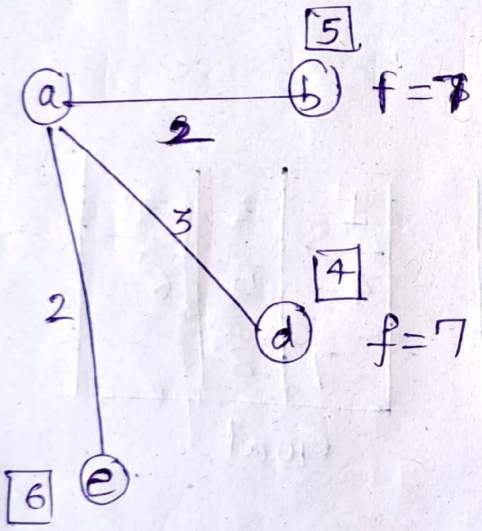
[so, google maps may use air distance as an heuristic for finding shortest path.]

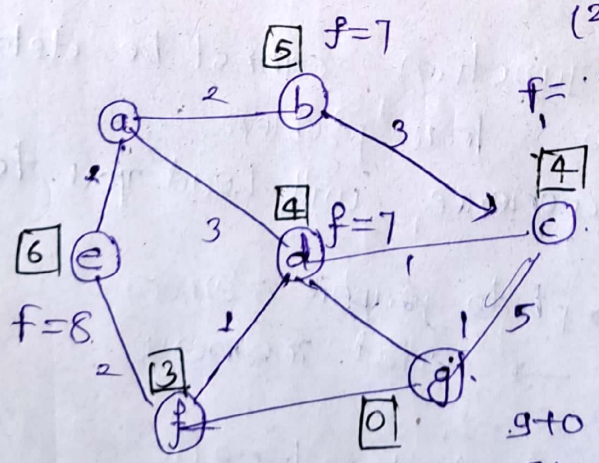
This is not calculated from the previous graph, it is just some estimate.

Scanned with OKEN Scanner

we will use $f(n) = g(n) + h(n)$







$f = g + h = 7$

$(2+2)$ 3

$(2+3)$ 4

$f = g + h = 9$ ✗

$f = g + h = 8$ ✓

$(3+1)$ (4)

$9+0 = 9$ ✗

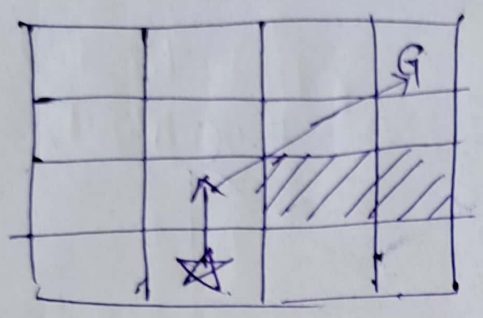
$8+0 = 8$ ✗

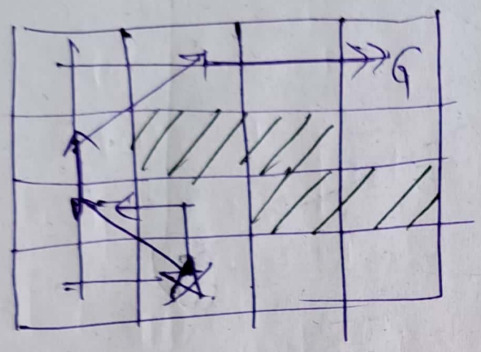
$7+0 = 7$ ✓

A* will be asked in exam

Maze problem

grid,
source,
obstacles
will be given to you.





Heuristics For 8-puzzle

2	1	4
3		6
7	5	8

start

1	2	3
4	5	6
7	8	

Goal.

- Heuristic function cannot be determined from problem.
- only Experience will Lead you to it.

- Humans → photo graphic memory
- Computer → serial memory.

$h(n)$ → no. of displaced cells.

Scanned with OKEN Scanner

~~Sudoku~~ Square roots :-

$\sqrt{2}$

pen-paper:-

	x y
x	$2.\overline{00}\ \overline{00}\ \overline{00}$
	x^2
y	00
	y^2
	00 ...

In computer programs, we use binary-search (or) bisections.

- use quadratic equation:-

$$ax^2+bx+c=0 \qquad x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}.$$

not like this

$x^2-2=0$ | $a=1$, $b=0$, $c=-2$

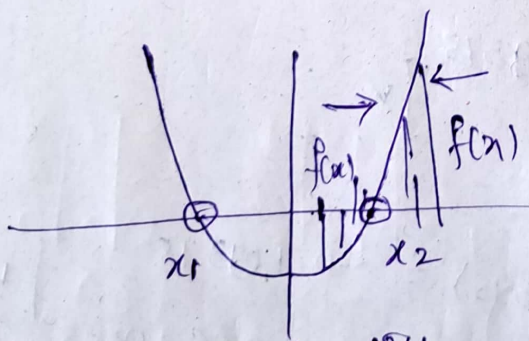
$x^2=2$

$$\therefore x=\frac{\pm\sqrt{-4\cdot1\cdot(-2)}}{2}$$

$$x=\frac{\pm\sqrt{8}}{2}$$

$$\boxed{x=\frac{\pm2\sqrt{2}}{2}=\pm\sqrt{2}}$$

draw graph of $x^2-2=0$.

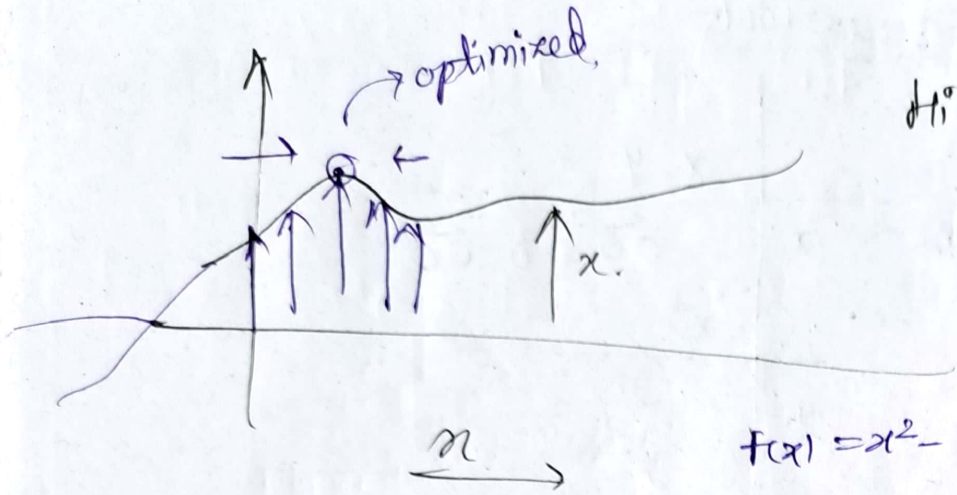


x_1, x_2 are roots.

move untill $f(x)=0$.

Newton-Raphson method.

we calculate till certain accuracy.



Hill climbing algo.

$f(x) = x^2 - 2$

$f(0) = -2$

$f(1) = -1$] root will be between them

$f(2) = +2$

$f(1) = -1$] root will be here

$\therefore f(1.5) = +0.25$

$f(2) = +2$

$f(1) = -1$] ✗

$f(1.25) = -0.4375$] ✓

$f(1.5) = +0.25$

$f(1.25) = -0.4375$ ✗

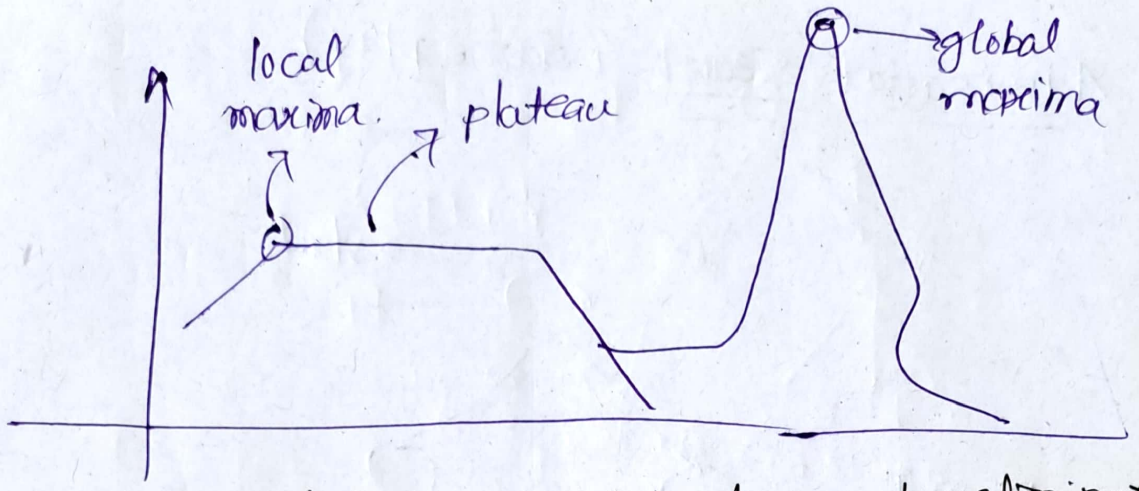
$f(1.375) = -0.109375$] ✓

$f(1.5) = +0.25$

$f(1.375) = -0.109375$] ✓

$f(1.4375) = +0.06640625$

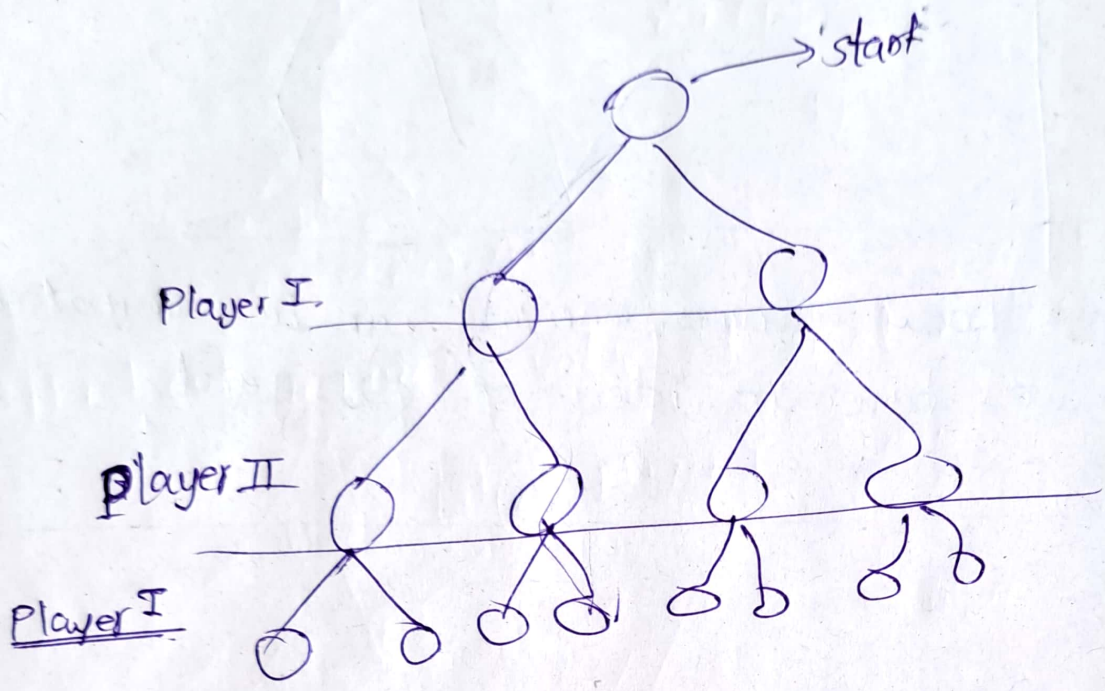
$f(1.5) = +0.25$] ✗



Local maxima, minimal can be eliminated if we have an idea of goal.

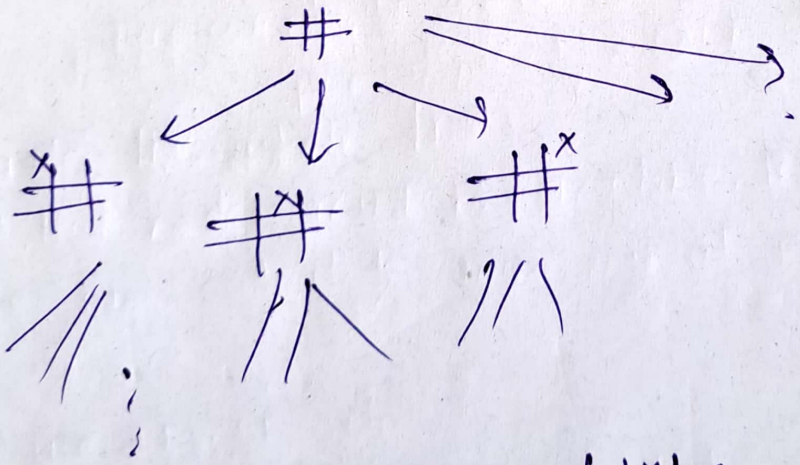
Scanned with OKEN Scanner

Adversarial Search (Game play).



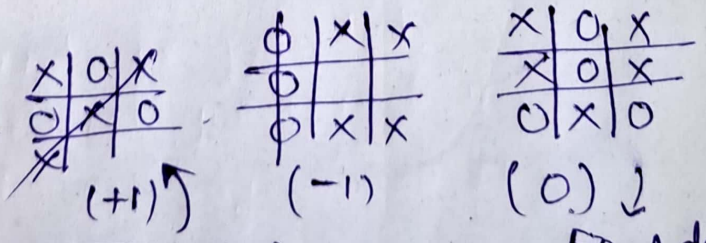
Win	Lose	Draw
1	0	0.5 , 0.5

- We are not finding 1st goal.
- we are finding goal with maximum reward.



Finally 3 cases.

Limited :-



Final depth

max. reward at min no. of steps

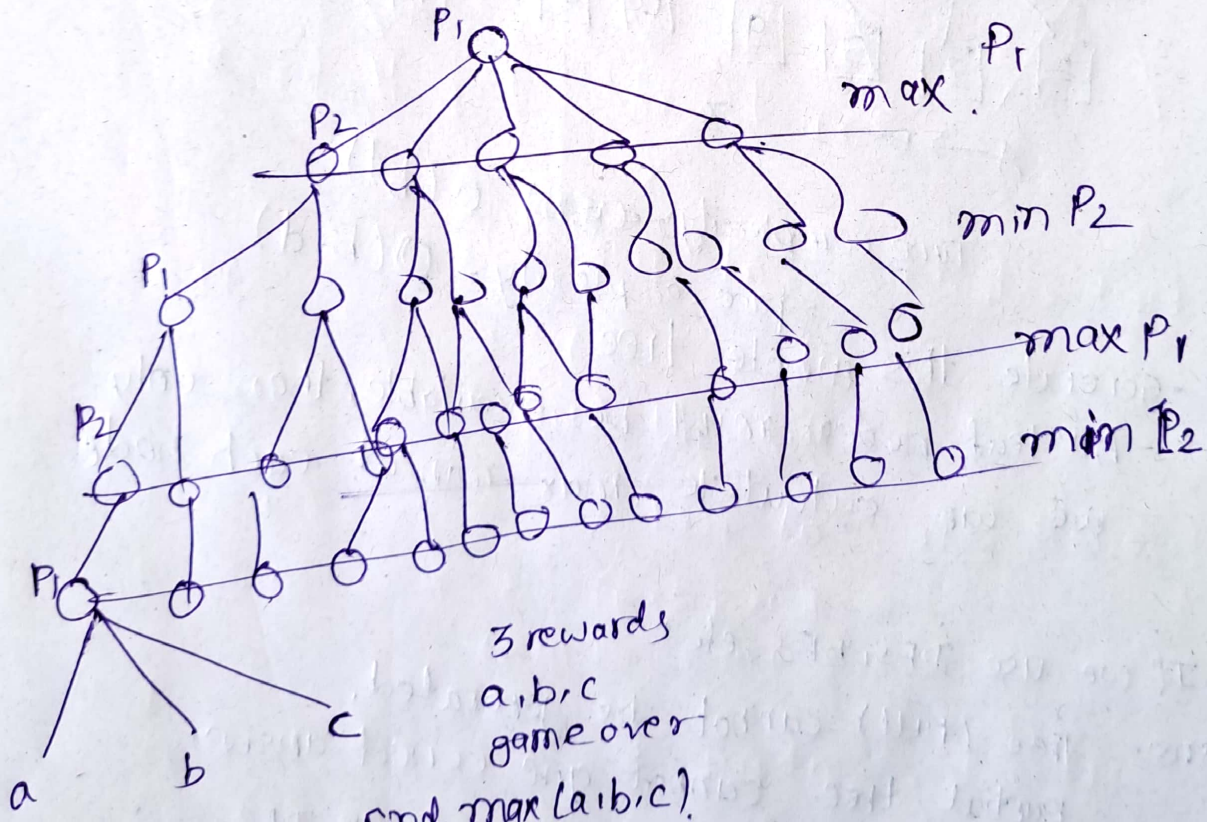
* These game trees will be solved using mini-max algo

zero-sum game tree

	P_1	P_2	sum
win	1	-1	0
Tie	1/2	-1/2	0

some games, may not have same no. of branches (ex- chess, at a point you cannot move anywhere (sometimes you can move in all directions, etc).

Try to maximize moves of Player I.
& minimize moves of Player II.

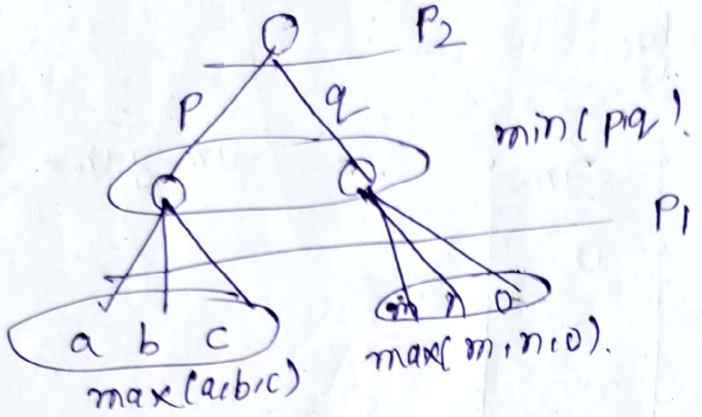


3 rewards
a, b, c
game over

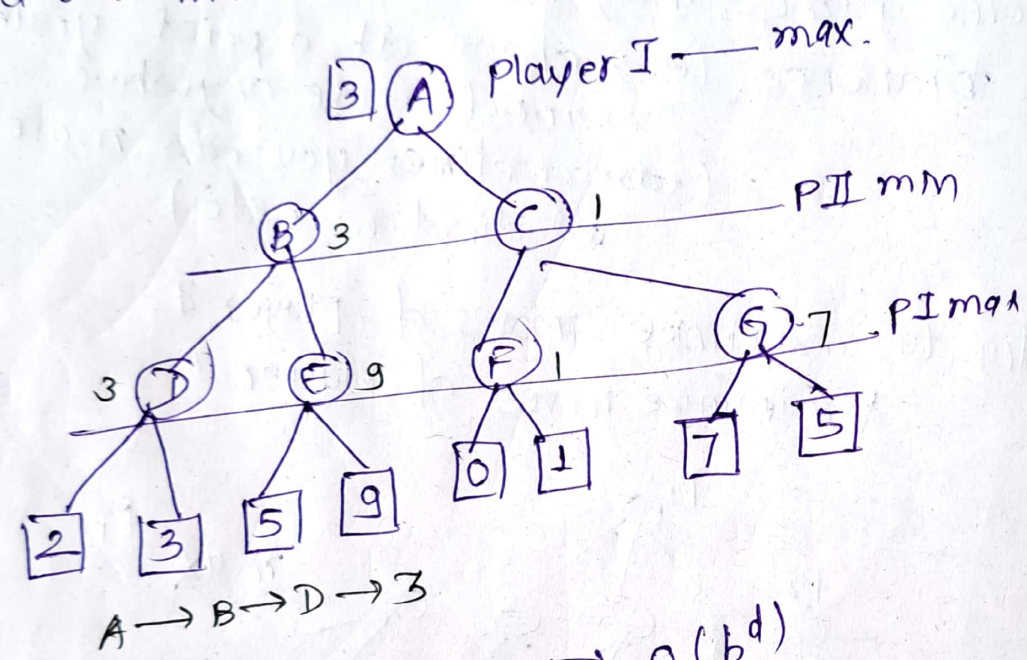
Find max (a, b, c).

If only 1 ~~path~~ branch → 1 reward game over

If more paths → max. of the rewards, game over.



Q] Find out maximum value at Goal node, ?



Time complexity :- $O(b^d)$

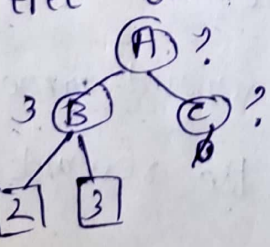
Space complexity :- $O(b \cdot d)$

- Generate this whole tree
- All leaf nodes must be available, then only we can calculate max min at each node.

If we use DFS, BFS, etc?

ans:- Tree (full) cannot be generated.

Partial tree cannot give correct answer.



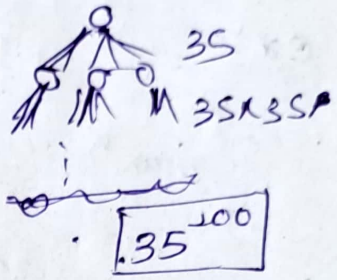
say used DFS.

but in games like chess where each player plays 50 moves in average.

∴ (100 moves, player + enemy)

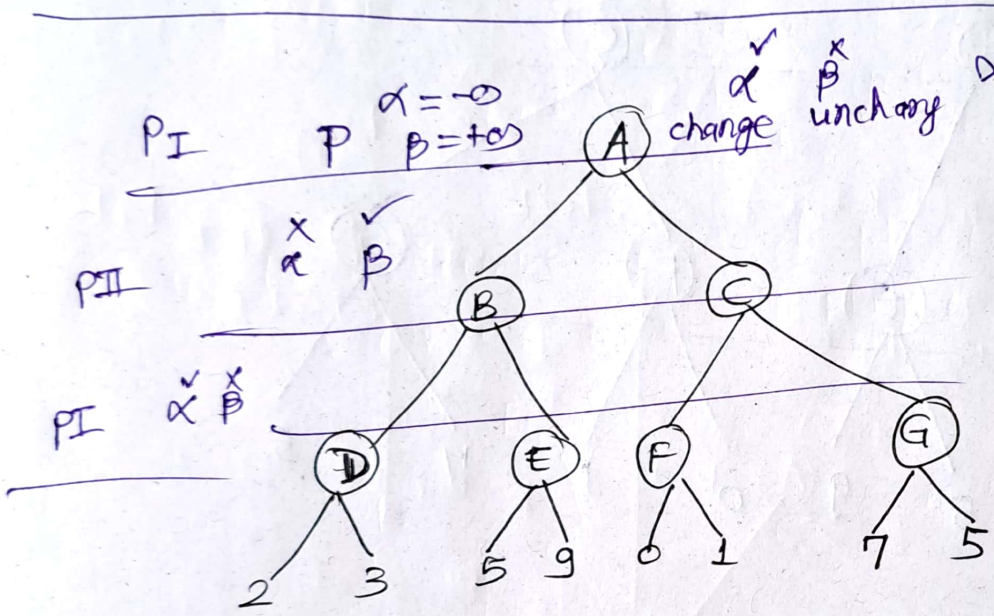
and 35 branches per move (in average)

∴ 35^{100} nodes.



such tree cannot be generated & stored.

We use α-β pruning in such case.



α-β pruning cut the tree so that some nodes are not visited

Let initially α = −∞
β = +∞

P_1 - α - max

P_2 - β - min

goal

stage 1) all α-β same values will be passed as it is, till Leaf nodes.

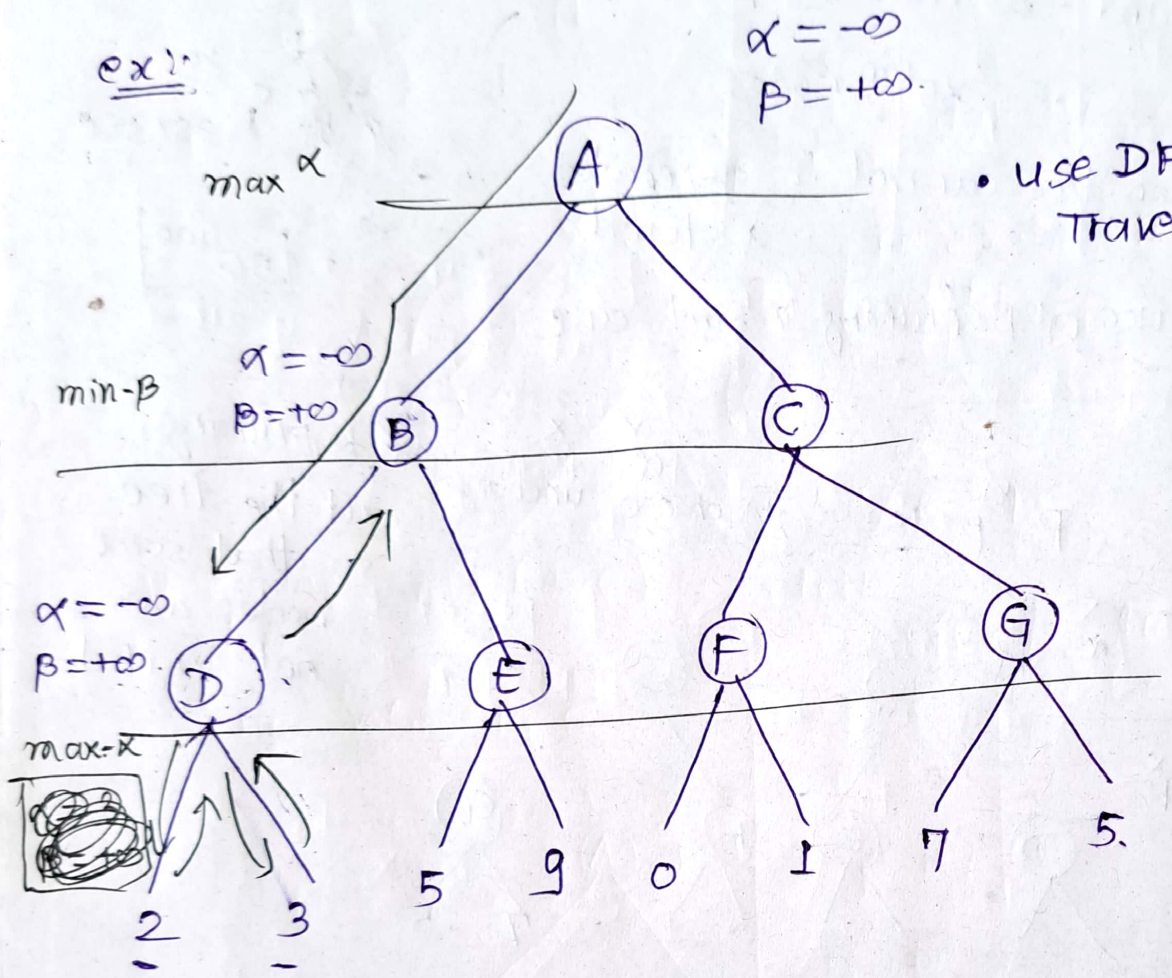
stage 2) calculate α, β for each node, while coming back.

If some node follows α ≥ β, prune its child nodes.

Scanned with OKEN Scanner

node value of a Node = max or min of child reward values.

ex:



• use DFS while Traversing.

First calculation, at point 2

$\alpha = \max(2, -\infty) = 2$

β = unchanged (+∞)

second calc, at point 3

$\alpha = \max(2, 3) = 3$

β = unchanged (+∞)

3rd calc, at point D.

$\alpha = \max(2, 3) = 3$

β = unchanged (+∞)

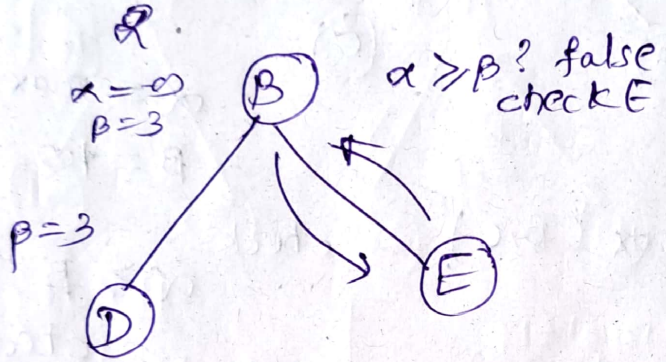
Scanned with OKEN Scanner

4th calc, min, β, at point B.

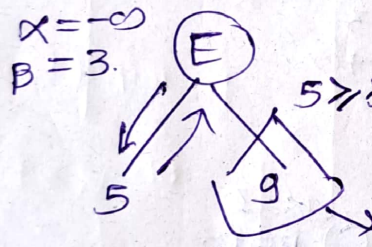
~~unchanged (3)~~

Previous values	data from child D:-
$\alpha = -\infty$	$\beta = \min(3, +\infty) = 3$
$\beta = +\infty$	~~unchanged~~ (

Finally $\boxed{\alpha = -\infty, \ \beta = 3}$



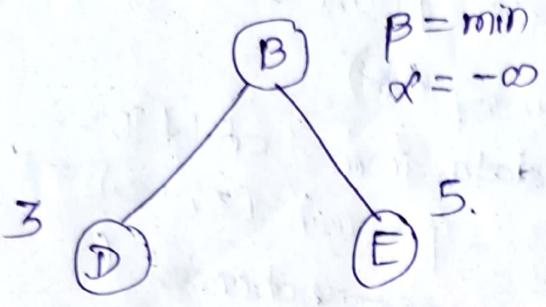
at E, α will change.



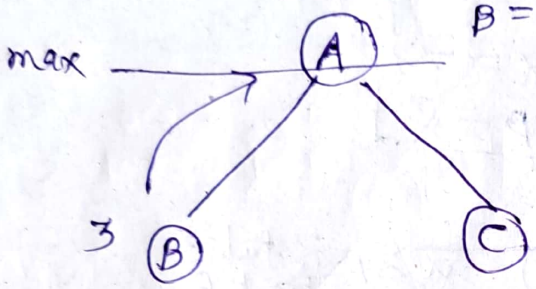
$\alpha = \max(-\infty, 5)$
$= 5.$

why? min of 3 and (5, 9) will be 3 itself
Hence, we cut those branches.

∴ $\boxed{\alpha = 5, \ \beta = 3}$ (E) ~~leaf~~ node with 1 reward value



$\beta = \min(5, 3) = 3.$
$\alpha = -\infty$



$\alpha = -\infty$ initially
$\beta = +\infty$

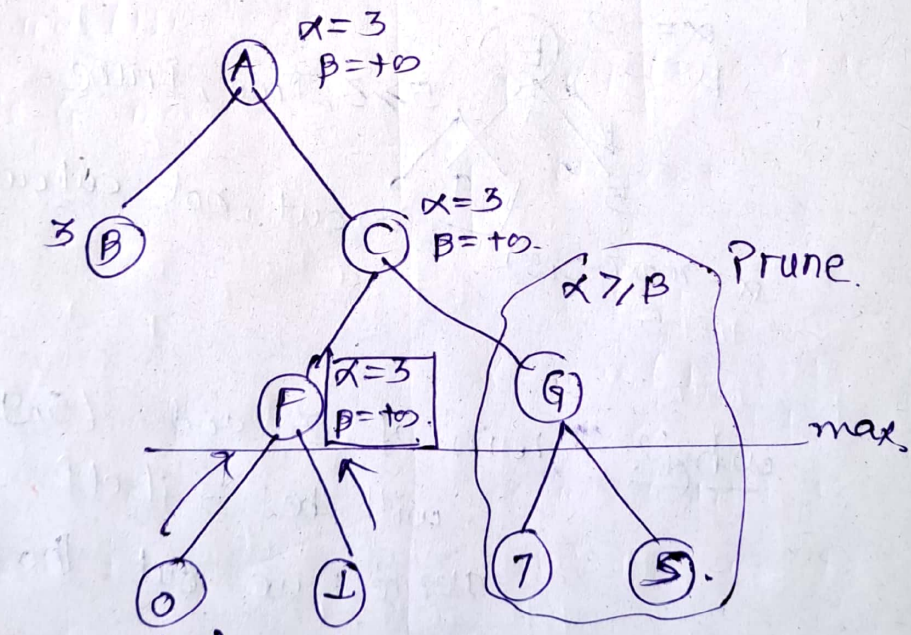
$\boxed{\text{Node value of } B = 3}$ ~~from B~~

$\therefore \alpha = \max(3, -\infty) = 3$
$\beta = +\infty$

Node value of A $= \max(B, C)$.

check $\alpha \geq \beta$? ~~True~~ False
check C.

<u>Pass values to C ?</u>



$\alpha = 3$
$\beta = +\infty$

$\alpha = 3$
$\beta = +\infty.$

$\boxed{\alpha = 3}$
$\beta = +\infty.$

$\alpha \geq \beta$

$\alpha = (3, 0)$
max $\boxed{\alpha = 3}$.

$\alpha = \max(1, 3)$
$\alpha = 3$

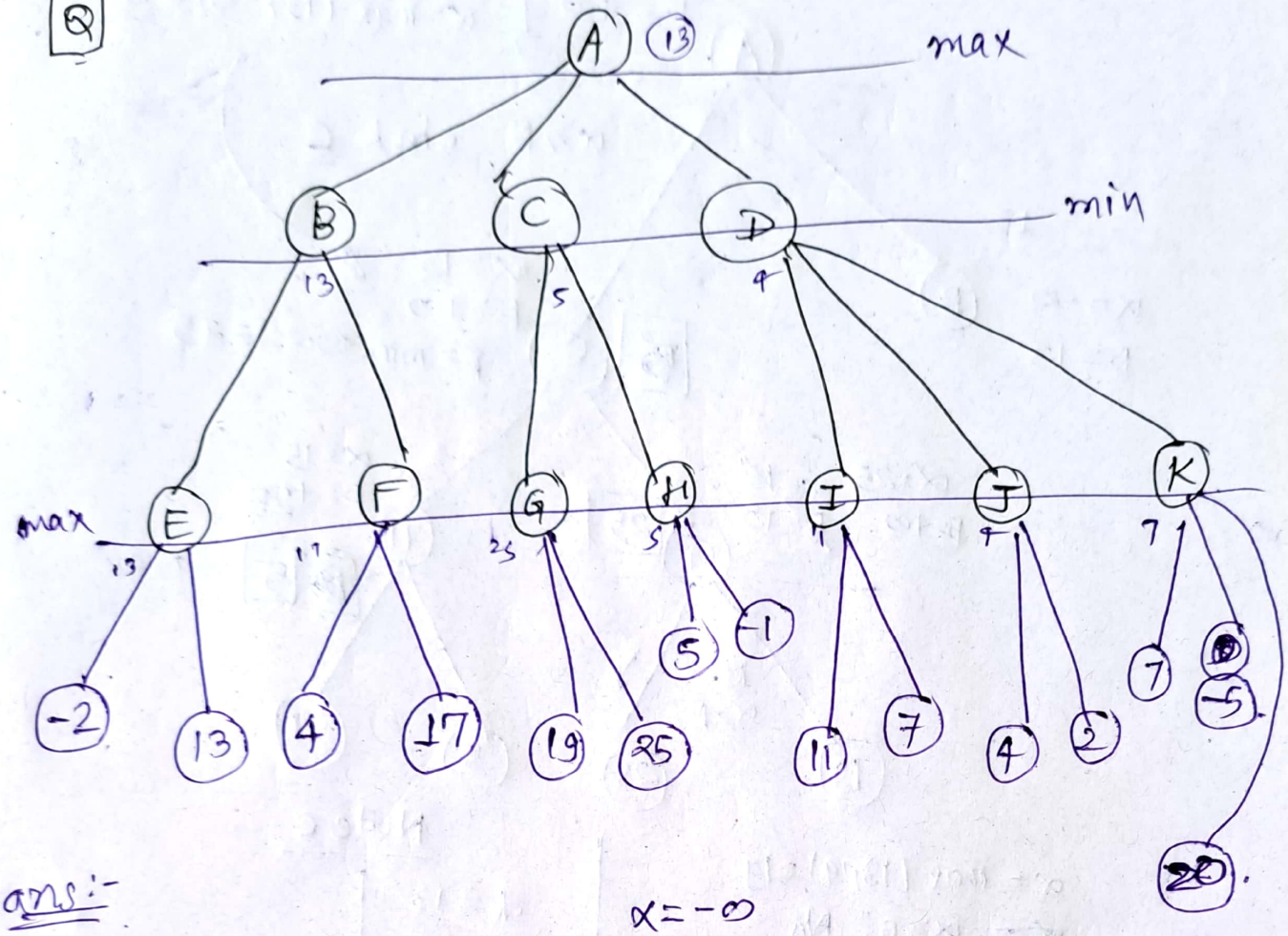
pruned.

<u>Finally</u> Node value of $C = \min(\text{node } F, G)$
$= \min(3)$
$= 3$

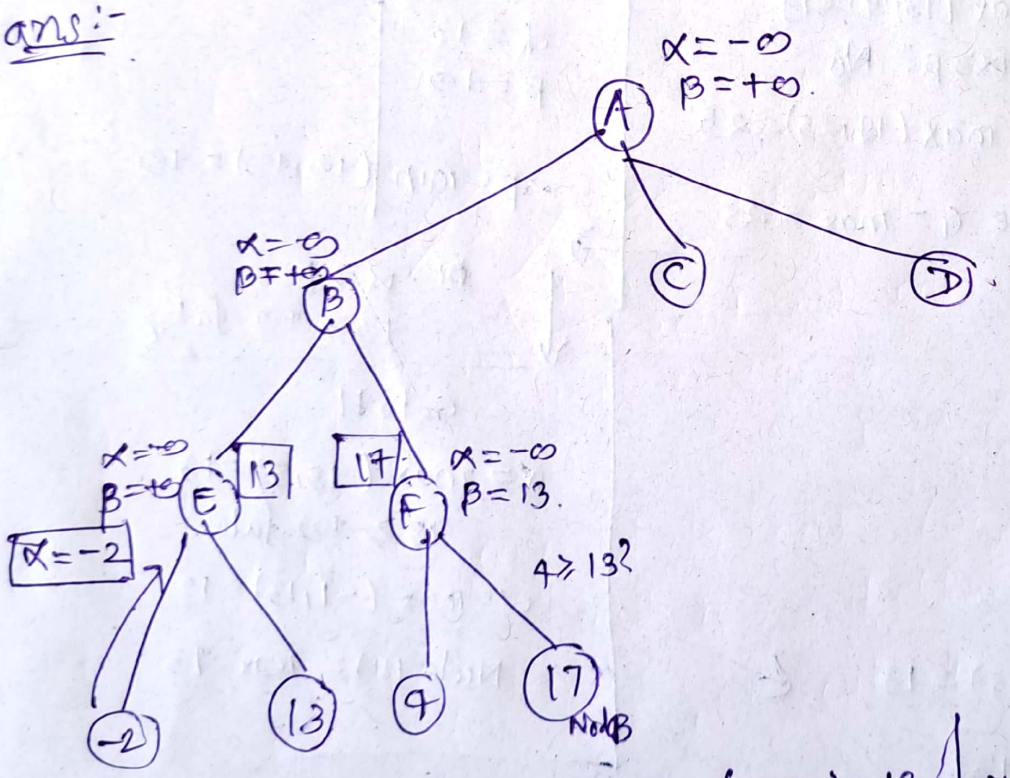
$\therefore$ Node value of $A = \max(B, C) = \max(3, 3) = \boxed{3}.$

Scanned with OKEN Scanner

Q



ans:-



$\alpha = \max(-\infty, 2) = -2$

$\alpha = \max(-2, 13) = 13$

node(E) = 13

$\beta = \min(13, +\infty) = 13$

$\alpha = -\infty$

check $\alpha \geq \beta$

$-\infty \geq 13$

Node F = 17

$\alpha = \max(-\infty, 4) = 4$

$\alpha = \max(4, 17) = 17$

Node B

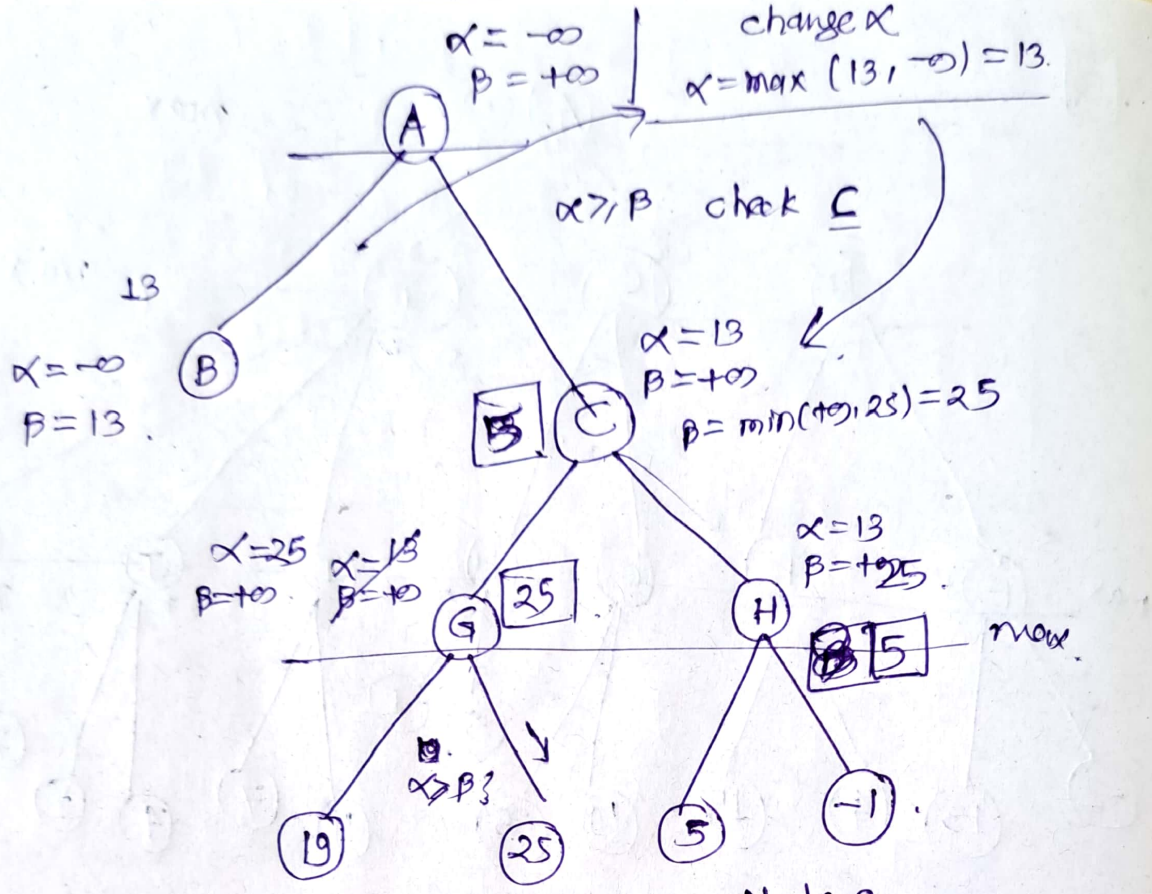
Node B = min (17, 13)

= 13

$\alpha = -\infty$

$\beta = 13$

Scanned with OKEN Scanner



α = max (13, 19) = 19

α ≥ β? No

α = max (19, 25) = 25.

Node G = max = 25.

Node C

α = 13

β = +∞.

∴ β = min (+∞, +∞) = +∞.

α ≥ β

13 ≥ +∞ false

Goto H

α = max (13, 5) = 13.

13 ≥ +∞ false

α = max (−1, 13) = 13

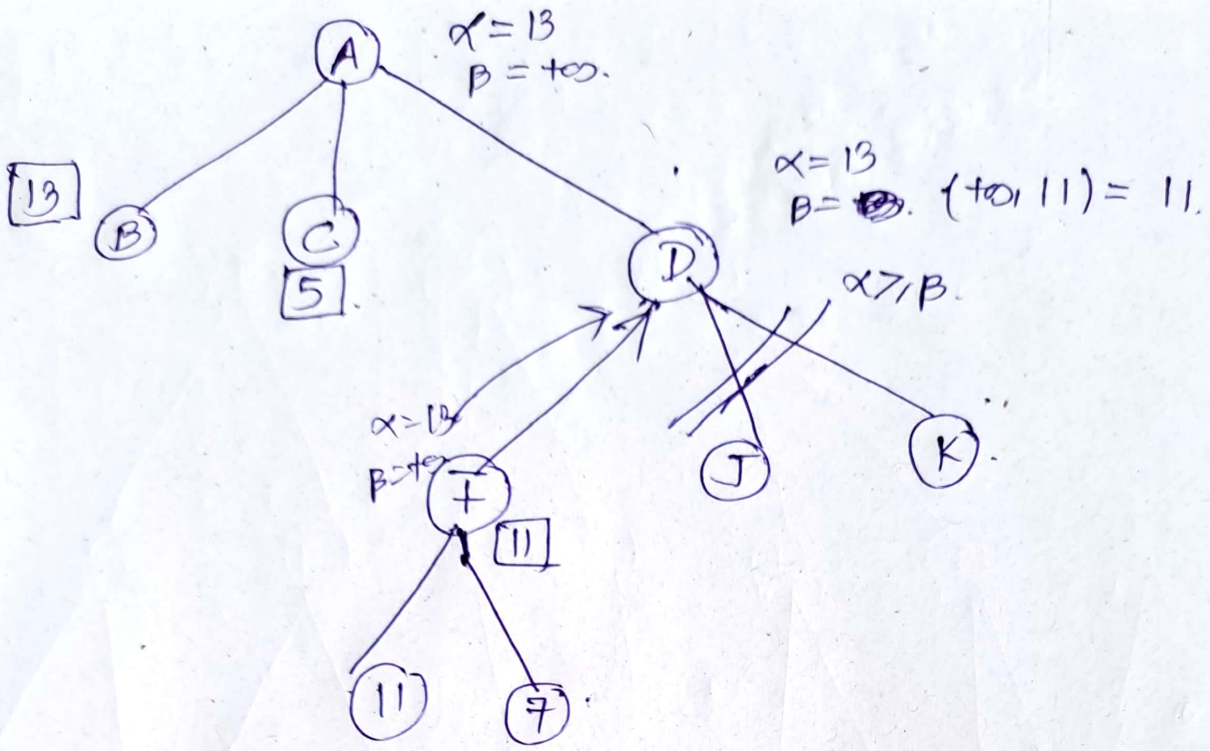
Node (H) = 13.

Node C

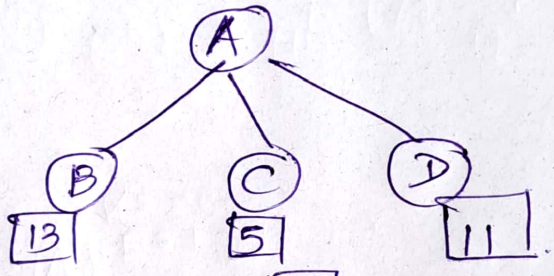
β, min

β = min (13, 25) = 13.

∴ $\boxed{\alpha = 13 \quad \beta = 13}$



∴ <u>Finally</u>



$\alpha = \max(13, 5, 11) = \boxed{13}$

Last class before midsem.

All topics discussed till now will be asked.

common topics of all section will be asked.

Scanned with OKEN Scanner