Chapter - 1.

Goals in Problem solving :-

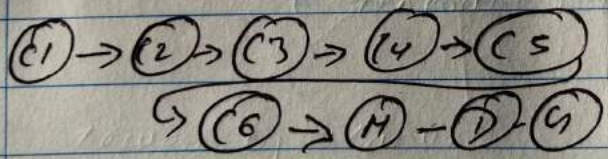
Fields / Application # qmp.

Goal :
- Target / Desired result / Accomplishment
what is to be achieved?

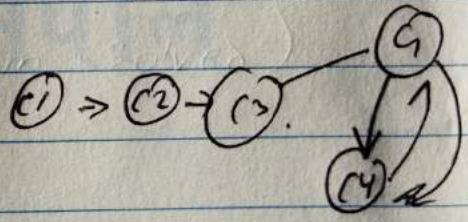
planning
- Set / Sequence of tasks to achieve a goal.

Types of planning :- (qmp)
1) linear planning
2) Non-linear planning

linear planning



Non-linear planning



- Planning Algorithm that works on one goal until completely finished before moving on to next goal.

works to achieve goal by interleaving (alternation of goals)

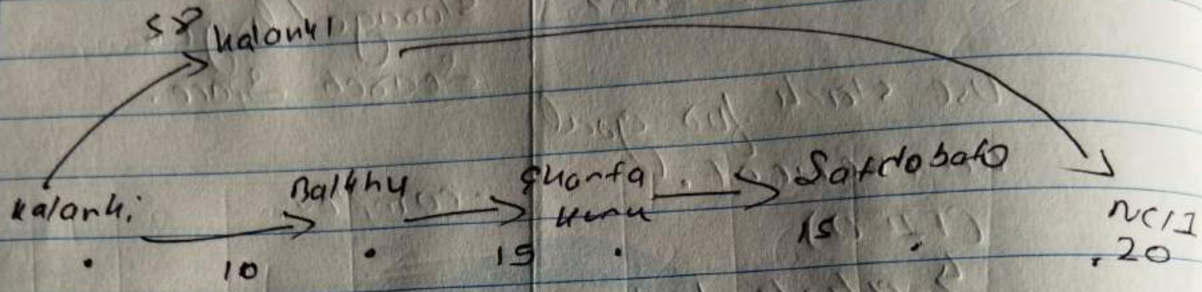
- Considers priority.

Preemptive ⇒ (switch) ~~हटे~~ ~~नहि~~ when higher priority task occurs
Non-Preemptive (doesn't switch)

S.NO.	TOPICS	PAGE NO.
	-doesn't prioritize task/no priority	- Uses Set for Goal Achievement.
		{{1,(1)}, {2,(2)}, {3,(3)}, {4,(4)}, {5,(5)}}
	- doesn't prioritize task / no priority.	- Require more storage / larger Search Space.
	- Use stack for goal achievement.	- complex.
	(FIFO)	- may produce optimal soln.
	↳ a/c2/c3/cyles	
	[AI में FIFO इस्तेमाल stack]	- complete.
	- Requires less storage/Reduced search space.	
	- Simple	
	- may not produce	
	- Incomplete.	

MEA (means-End Analysis) (imp) :-

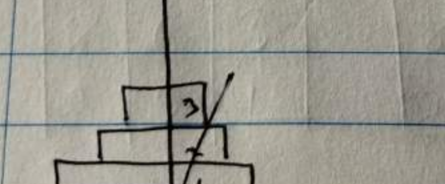
[अहिले state र दैनी final state बीच best intermediate state choose गर्ने कार्य MEA]

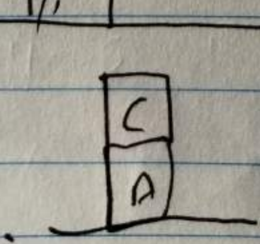


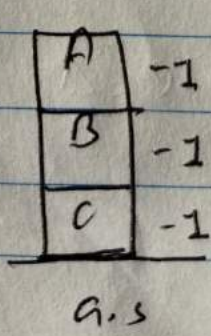
1) Current State
2) Final state
3) Goal difference. (Δ)

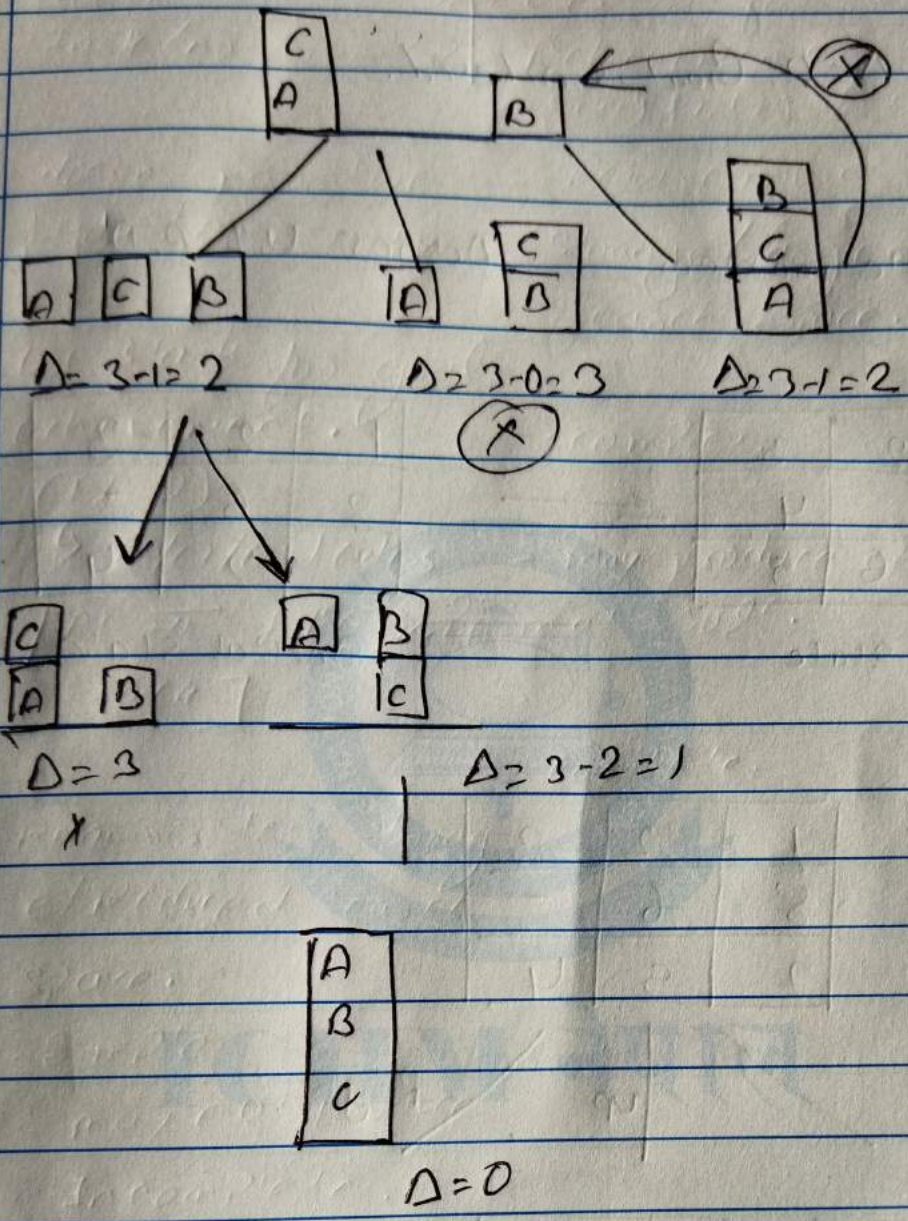
GPS (General Problem solver)

Tower of Hanoi









Algorithm that works based on 3 parameters:

1) Initial state / current
2) Final state / Goal.
3) Goal difference (Δ)

- Reducing goal difference between I.S and G.S until goal

difference (Δ), ≥ 0.

- when $\Delta = 0$, Goal is achieved.

8 puzzle problem: Actions: U, L, R, D.

1	2	3
8	-	4
7	6	5

Goal state

←

1	2	3
8	6	-
7	5	4

Initial state.

1	2	3
8	6	-
7	5	4

Down | up | Right.

1	2	-
8	6	3
7	5	4

1	2	3
8	6	4
7	5	-

1	2	3
8	-	6
7	5	4

$\Delta = 9 - 4 = 5$ ×

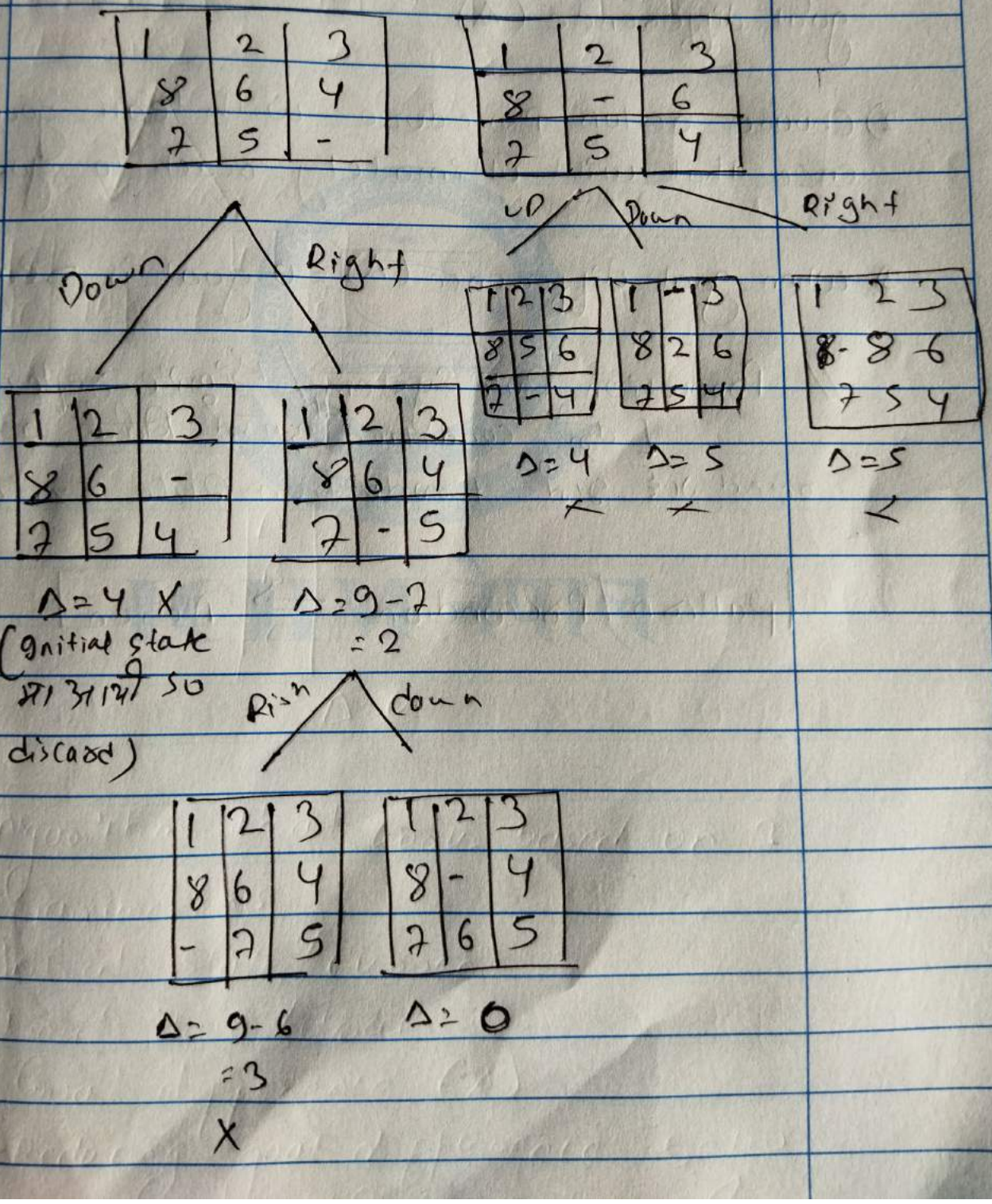
$\Delta = 9 - 6 = 3$

$\Delta = 9 - 6 = 3$

S.NO.	TOPICS	PAGE NO.

State Space Tree

- Tree which shows all possible states from current state.



Algorithm (V.V.G)

MEA Algorithm:

1) Define initial state, goal state and calculate goal difference (Δ)

2) Choose action/procedure with least goal difference that will ultimately reach to goal.

3) When goal difference (Δ) = 0 goal is reached.

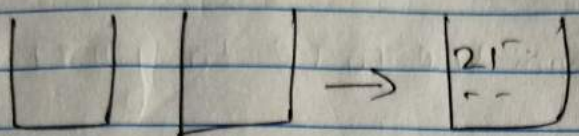
Production Rule System / Simple reflex System

[condition ... action ... production Rule system]

[Pattern / Precondition → Action]

S.NO.	TOPICS	PAGE NO.

Water Jug problem.



0	0
0	3
3	0
3	3
4	2

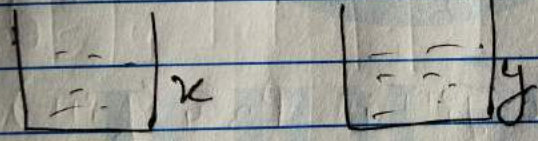
Possible actions:

1) Fill 4l Jug
2) Fill 3l Jug
3) Empty 4l Jug
4) Empty 3l Jug
5) Pour 4l to 3l
6) Pour 3l to 4l

3l full 4l not full

4l Jug full
4l Jug not full

#

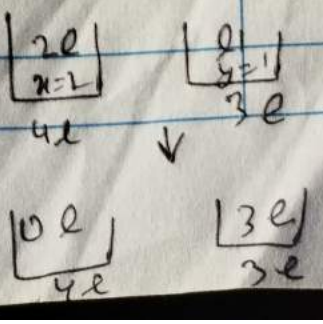


x: Quantity of water in 4l Jug
y: " of water in 3l Jug

1) Fill 4l Jug (F4)
$x < 4 \rightarrow (4, y)$

2) Fill 3l Jug (F3)
$y < 3 \rightarrow (x, 3)$

3) Empty 4l Jug (E4)
$x > 0 \rightarrow (0, y)$

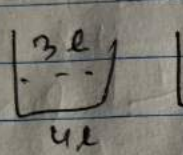


4) Empty 3ℓ jug (E3)

$y>0 \rightarrow (x, 0)$

5) Pour water from 4ℓ to 3ℓ to fill it (P4-3 (fill))

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



P4-3 fill

6 Pour water from 4ℓ to 3ℓ to not fill (P4-3 (not fill))

$x>0$ and $x+y<3 \rightarrow (0, x+y)$

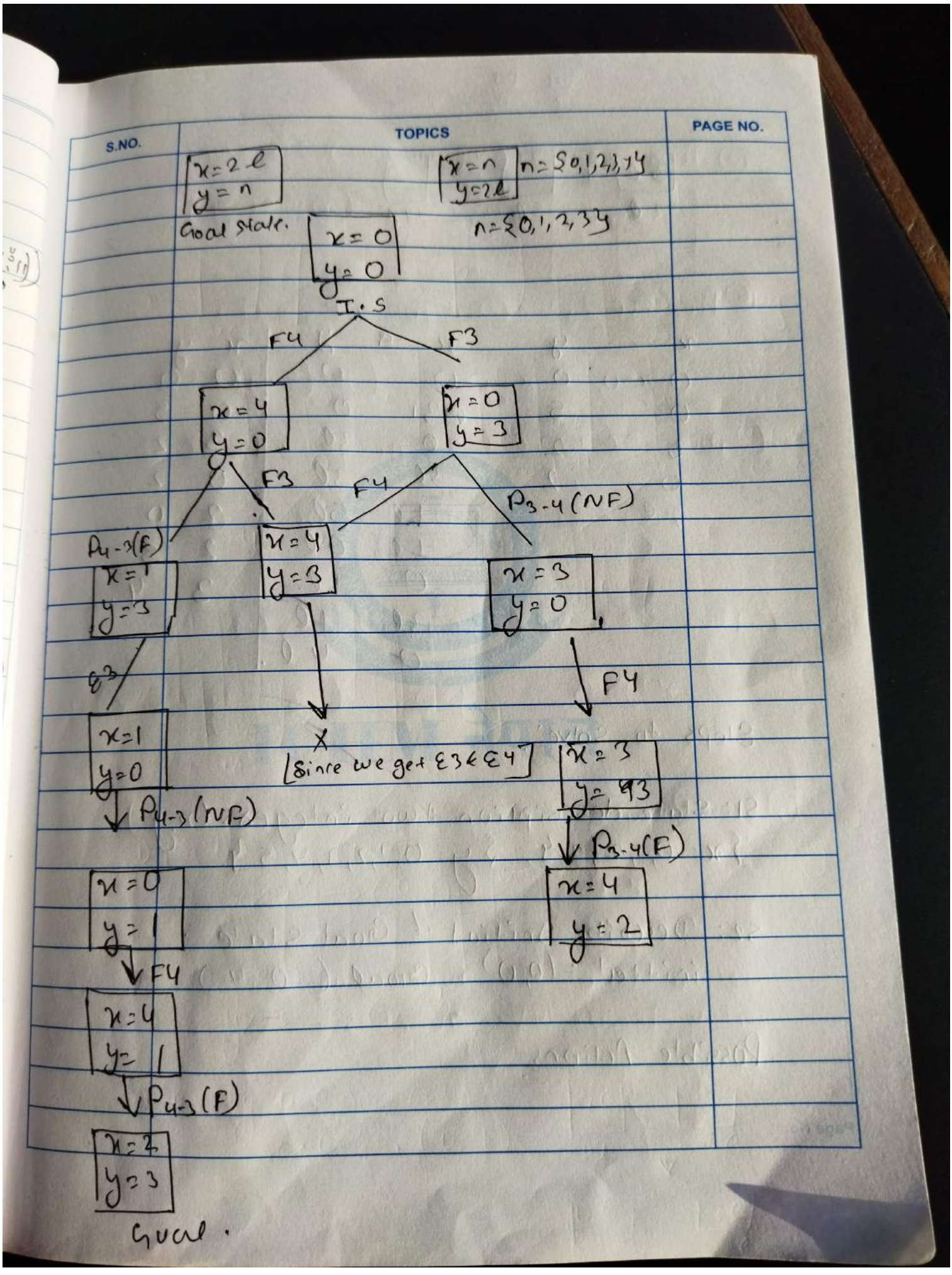
7) Pour 3ℓ to 4ℓ to fill it P_{3-4} (fill)

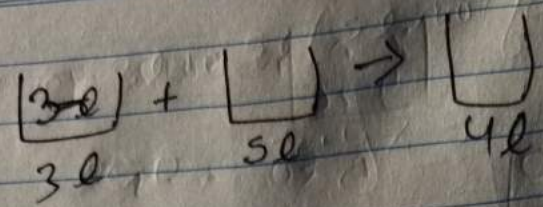
$y>0$ & $x+y \geq 4 \rightarrow (4, y-(4-x))$

8. pour 3ℓ to 4ℓ to not fill P_{3-4} (no fill)

$y>0$ & $x+y<4 \rightarrow$ ~~$(0, x+y)$~~ ~~$(x+y, 0)$~~ $(x+y, 0)$

S.NO. | TOPICS | PAGE NO.





x	y
0	0
0	5
3	2
0	2
2	0

x	y
0	0
3	0
0	3
3	3
1	5
1	0
0	1
3	1
0	4

x	y
0	0
0	5
3	2
0	2
2	0
2	5
3	4.

Steps to Solve:

S1: State description two integers (x, y)
{x: 0, 1, 2, 3}, {y: 0, 1, 2, 3, 4, 5}.

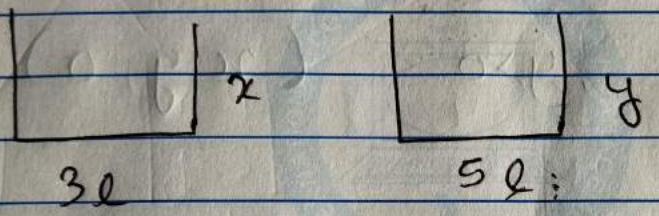
S2: Describe Initial & Goal State:
initial = (0, 0), Goal (0, 4)

Possible Actions:

S.NO.	TOPICS	PAGE NO.

(i) Fill 3L Jug
(ii) Fill 5L Jug
(iii) Empty 3L Jug
(iv) Empty 5L Jug
(v) Pour 3L to 5L ← 5L jug full
5L jug not full
vi Pour 5L to 3L ← 3L jug full.
3L jug not full.

Solving using production rule system.



(1) Fill 3L Jug (F3)

$x < 3 \rightarrow (3, y)$

(2) Fill 5L Jug (F5)

$y < 5 \rightarrow (x, 5)$

(3) Empty 3L Jug (E3)

$x > 0 \rightarrow (0, y)$

(4) Empty 5L Jug (E5)

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

(1st) (2nd)
3L 5L

(5) Pour 3L to 5L (5L not full) P_{3-5NF}

$x>0, \quad x+y<5 \quad (0, x+y)$

(6) Pour 3L to 5L (5L full) $P_{3-5(F)}$

$x>0, \quad x+y \geq 5 \quad (x-(5-y), 5)$

(7) pour 5L to 3L (3L not full) $P_{5-3(N-F)}$

$x+y<3, \quad y>0 \quad (x+y, 0)$

(8) Pour 5L to 3L (3L full (P_{5-3} F)

$x+y \geq 3, \quad y>0 \quad (3, y-(3-x))$

(b) Drawing State Space tree:

$x=0$

$y=0$

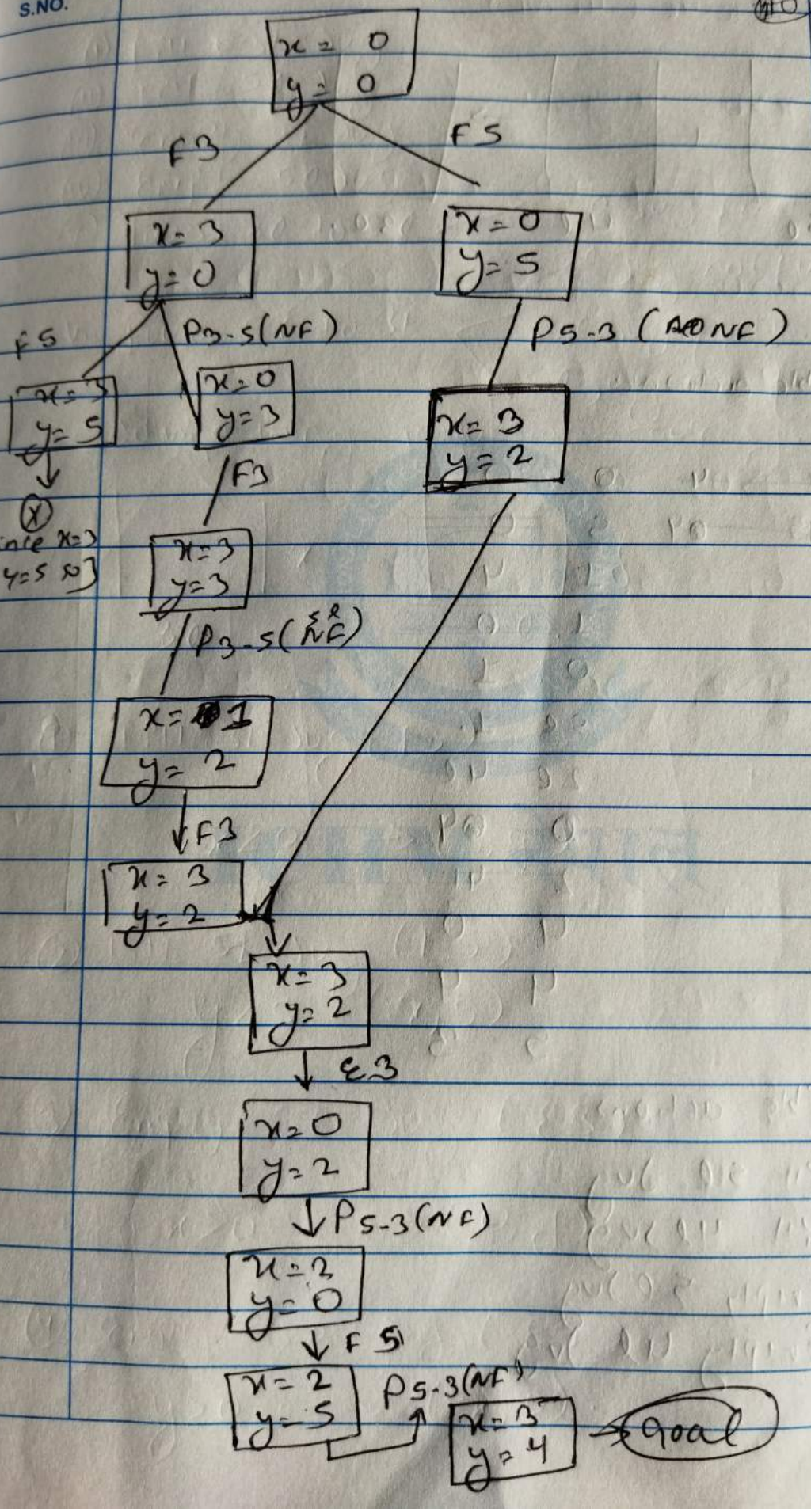
let x denote the quantity of water in 3L Jug

let y denote the quantity of water in 5L Jug.

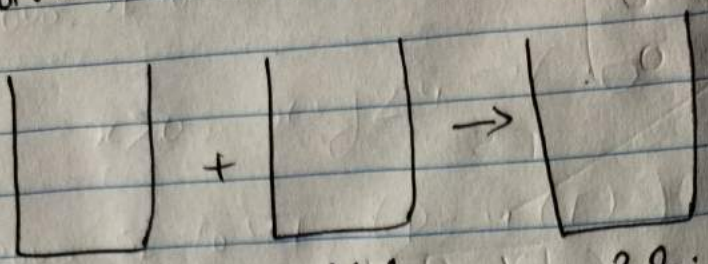
S.NO.

Page No.

S.NO. TOPICS PAGE NO.



Solve #.



Possible actions.

~~0~~	~~4~~
~~5~~	~~4~~
5	0
1	4
1	0
0	1
5ℓ	1.
2ℓ	4ℓ
~~0~~	~~4~~
~~2~~	~~4~~
4	0
4	4
5	3

0	4
4	0
	4

Possible actions?

1) Fill 5ℓ Jug
2) Fill 4ℓ Jug.
3) Empty 5ℓ Jug
4) Empty 4ℓ Jug

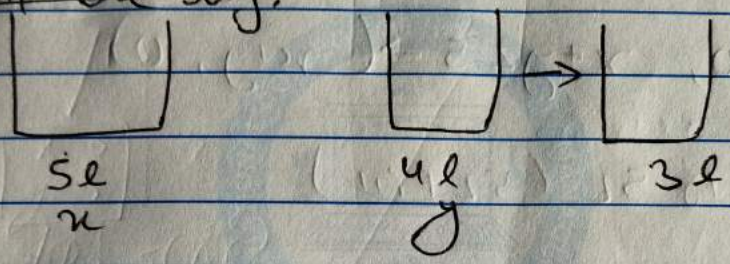
S.NO. | TOPICS | PAGE NO.

5) pour 5ℓ to 4ℓ → 4ℓ fill
→ 4ℓ not fill

6) pour 4ℓ to 5ℓ → 5ℓ fill
→ 5ℓ not fill

a) Now solving using production rule system.

1) ~~Fill 5ℓ jug.~~



let x denote the quantity of water in 5ℓ water

let y denote the quantity of water in 4ℓ water.

1) Fill 5l jug.

$x < 5 \rightarrow (5, y)$

2) Fill 4ℓ jug.

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

3) Empty 5ℓ jug.

$x > 0 \rightarrow (0, y)$

4) Empty 4ℓ jug.

$y > 0 \rightarrow (x, 0)$

5. Pour 5L to 4L (4L NF)

$x > 0,\ x+y < 4 \rightarrow (0,\ x+y)$

6. Pour 5L to 4L (P5-4(F))

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

7. Pour 4L to 5L (5L NF)

$y > 0,\ x+y < 5 \rightarrow (x+y,\ 0)$

8. Pour 4L to 5L (5L Fill)

$y > 0,\ x+y < 5 \rightarrow (5,\ y-(5-x))$

Now,

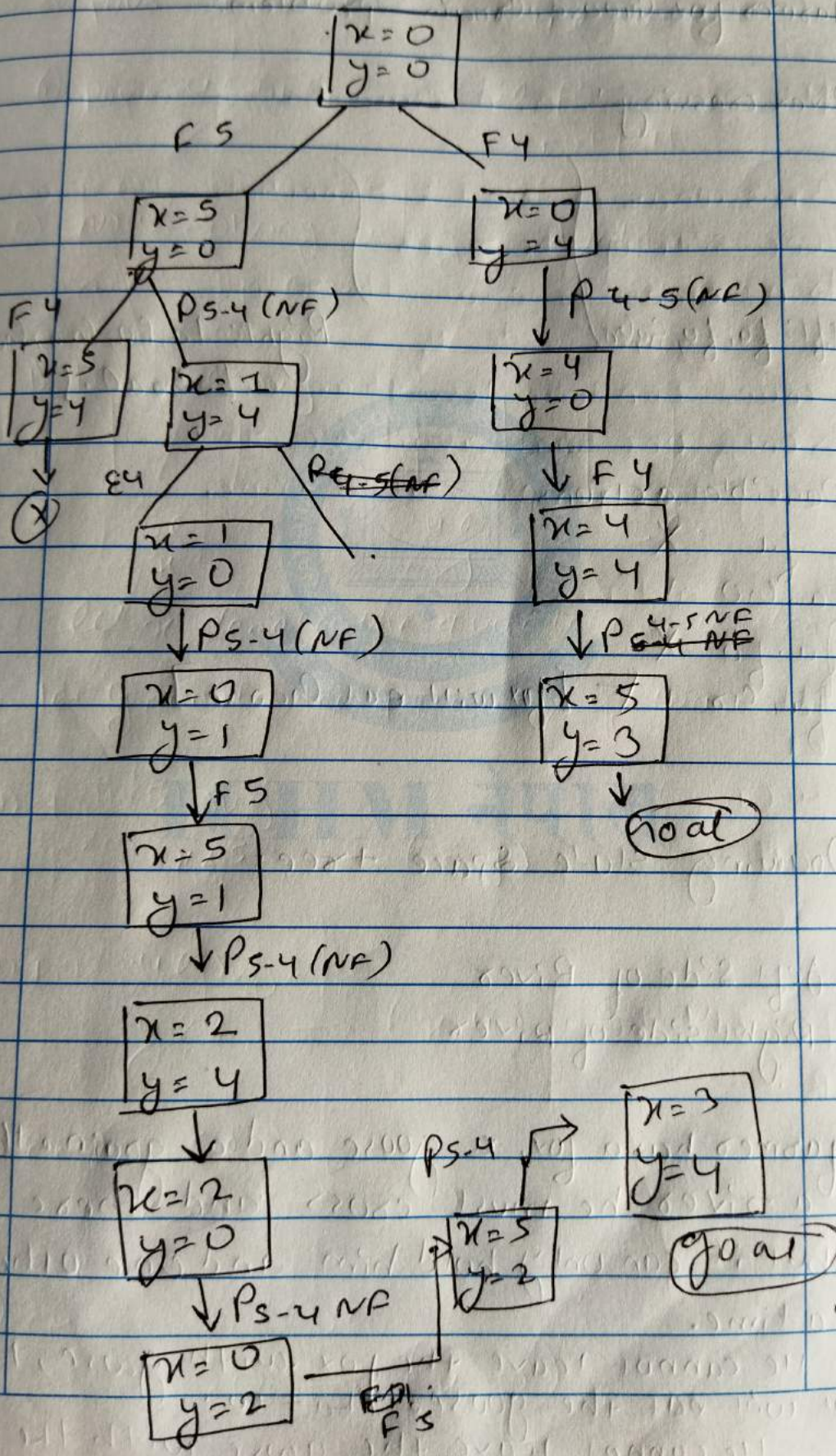
Drawing state space tree:

$x = 0$

$y = 0$

let x denote the quantity of water in 5L Jug

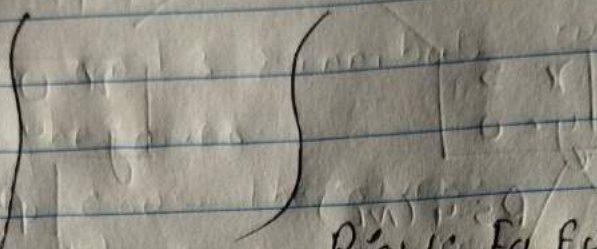
let y denote the quantity of water in 4L Jug


x=0 y=0
F5
F4
x=5 y=0
x=0 y=4
F4
P5-4 (NF)
P4-5(NF)
x=5 y=4
x=1 y=4
x=4 y=0
F4
x=1 y=0
x=4 y=4
P5-4(NF)
x=0 y=1
x=5 y=3
goal
F5
x=5 y=1
P5-4(NF)
x=2 y=4
x=2 y=0
P5-4
x=3 y=4
goal
x=5 y=2
P5-4 NF
x=0 y=2
F5

S.NO.

farmer fox Goose Grain problem:-

River crossing:



Left: fa, fo, Go, Gr

Right: Fa, Fo, Go, Gr.
[Goal].

Possible actions:

i) fa X
ii) fa, Go ✓
iii) fa, fo X
iv) fa, Gr ✗ [fox will eat Goose]

Drawing state space tree:

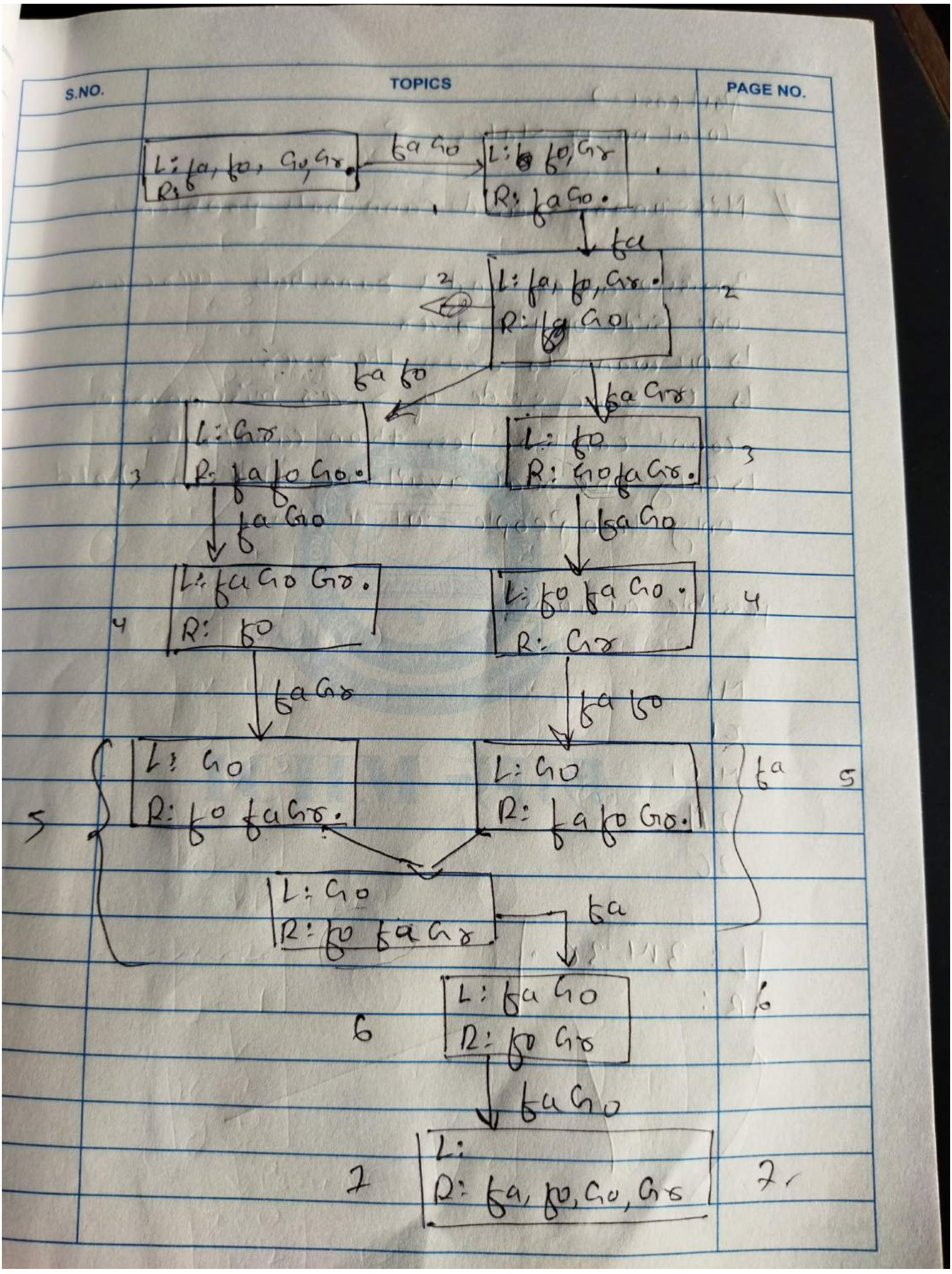
L: left side of River
R: Right side of River

#

A farmer has a fox, a goose and a grain. He comes to a river he must cross, and there is a boat but it can only hold him and one other thing at a time.

He cannot leave the fox with gose, because the fox will eat the goose. ~~That is~~.

He cannot leave the gose with the grain, because the gose will eat it.


L: fa, fo, Go, Gr.
R: f
fa Go
L: f fo, Gr
R: fa Go.
fa
2
L: fa, fo, Gr.
R: fa Go
fa fo
fa Gr
L: Gr
R: fa fo Go.
3
L: fo
R: Go fa Gr.
3
fa Go
fa Go
L: fa Go Go.
R: fo
4
L: fo fa Go.
R: Gr
4
fa Gr
fa fo
L: Go
R: fo fa Gr.
L: Go
R: fa fo Go.
fa
5
5
L: Go
R: fo fa Gr
fa
L: fa Go
R: fo Gr
6
6
fa Go
L:
R: fa, fo, Go, Gr
7
7.

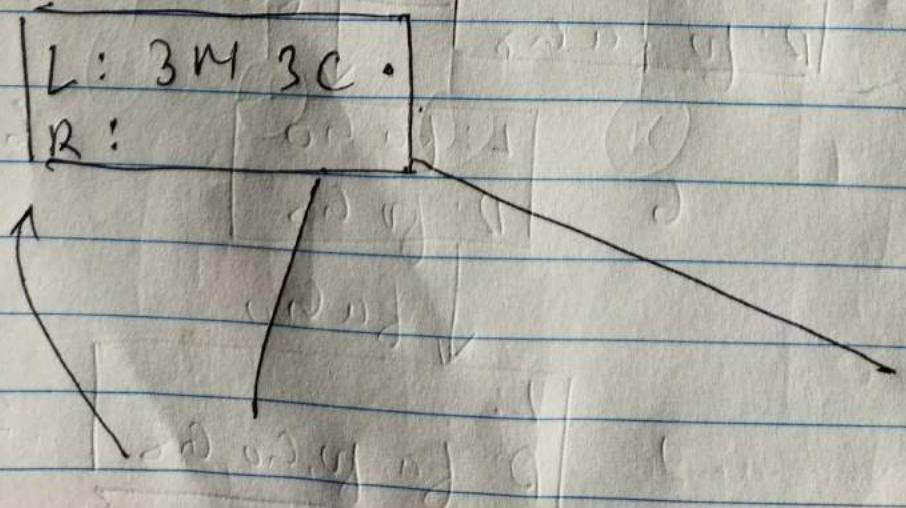
Pathcost = 7
Total no. of states = 8

Missionaries and cannibals problem :-

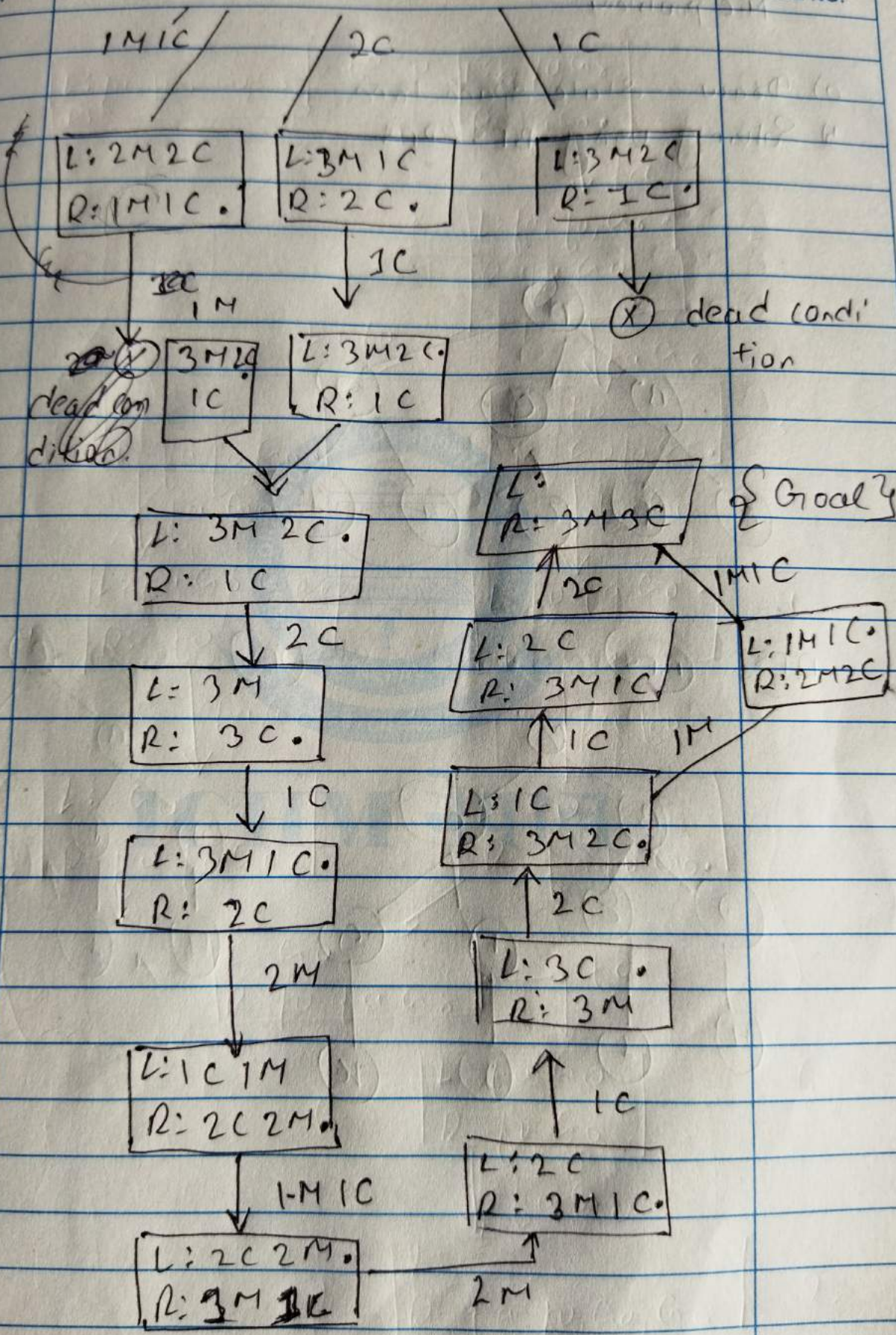
3 missionaries and 3 cannibals were on one side of the river.
↳ All want to cross the river.
↳ On same side of river missionaries count can't be less than cannibals.
↳ Only one boat available that can hold only two people at a time.

Possible actions:

M1
C1
M1C1
2M
2C



S.NO. | TOPICS | PAGE NO.

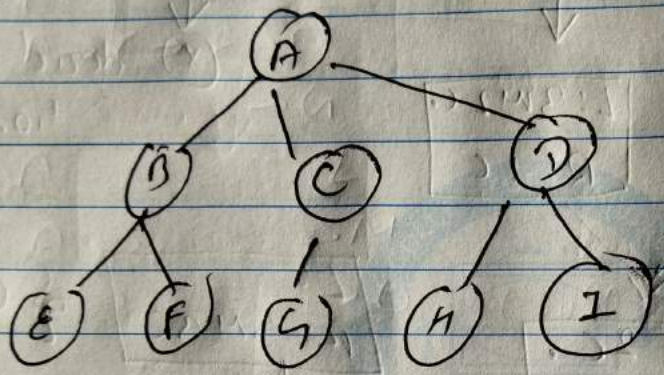


21 NOV

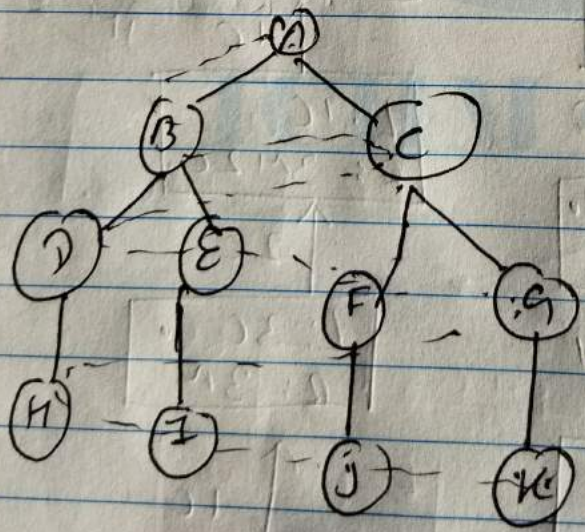
S.NO.

M-C problem:

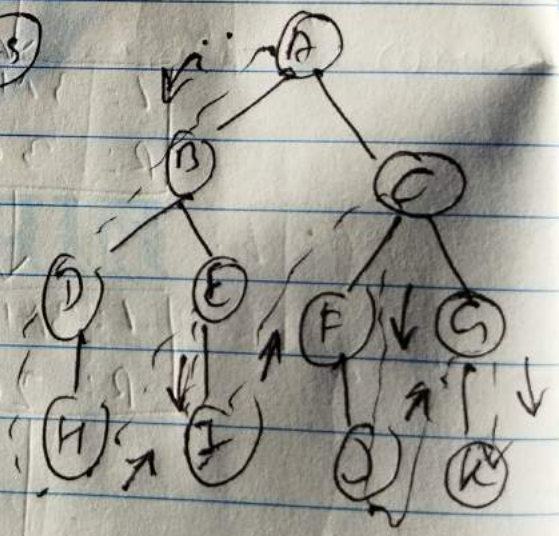
a) Draw state space tree.
b) Solving using BFS/DFS.



BFS:

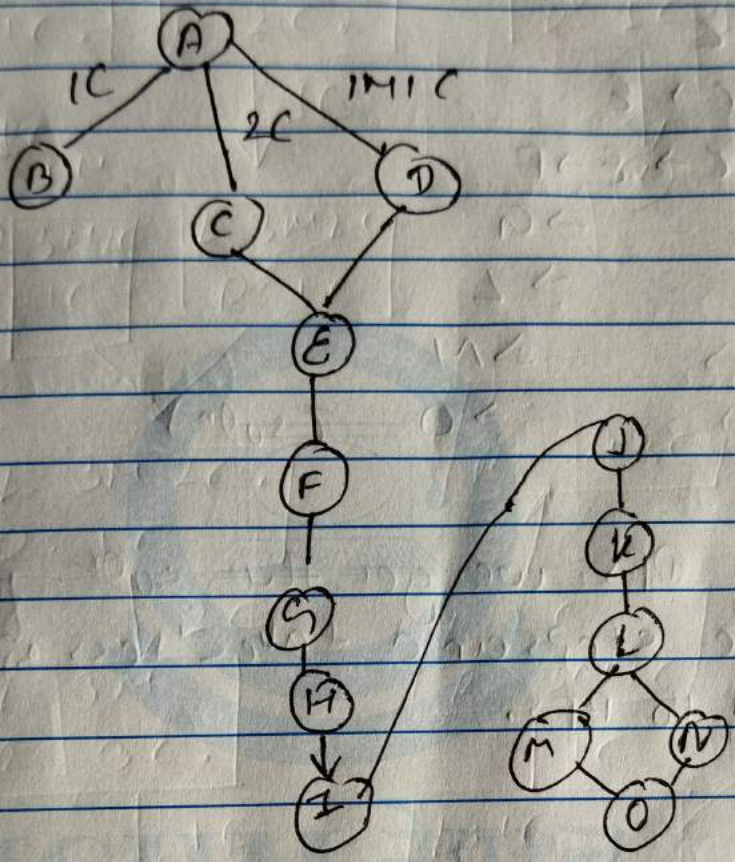


DFS:



S.NO.	TOPICS	PAGE NO.

BFS / DFS of Missioneg / Cannibal :-



BFS.

A

A → B A → C A → D

(no goal) A → C → E A → D → E

A → C → E → F A → D → E → F

A → C → ... → G A → D → ... G

A → C → ... → L A → D → ... L

A → C → ... → M A → D → E → M A → C → ... N A → D → ... N

... → O ... → O ... → O ... → O

DFS:

A

A → B (no goal)

A → C

A → C → E

A → C → E → F

- - - → G

A - - - → L

A → - - - → M

A → - - - → O (soln)

⇒ Using DFS we get the soln faster and in single steps and takes less space as compared to BFS.

S.NO.	TOPICS	PAGE NO.

Advantage of using BFS for missing values & cannibals

- Gives more solution (4 sol^n)

Disadvantage

- As it gives more solution it is time consuming.

Learning to Analogy

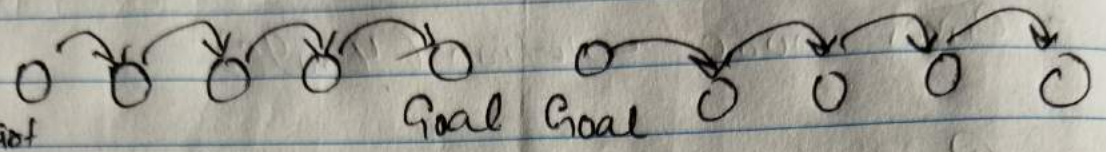
(Analogical reasoning / learning)

Tumor problem
Castle problem

- comparison of problems & applying solution from known problem to the new problem.
- 4 R's: Retrieve: Accessing from DB
  Reuse: Mapping same to sol^n
  Revise: Modification to the sol^n
  Retain: Store

22nd NOV

S.NO

Forward Chaining	Backward chaining
	
- Starts from initial state and aims for conclusion.	- starts from goal and backward search to necessary conclusion.
- Data Driven inference technique (inference = to extract something)	- Goal driven inference Technique.
- Bottom up approach	- Top-Down approach.
- Breadth First Search.	- Depth first search
eg: Diagnosis Recommendation in Disease prediction bots.	Navigation System in Driverless car.
- The number of final solution is infinite (multiple)	The no. of final solution is limited.

S.NO.	TOPICS	PAGE NO.
	Mycin	

- Expert system to identify bacterial infections and recommended antibiotics.

- 500 production rules.

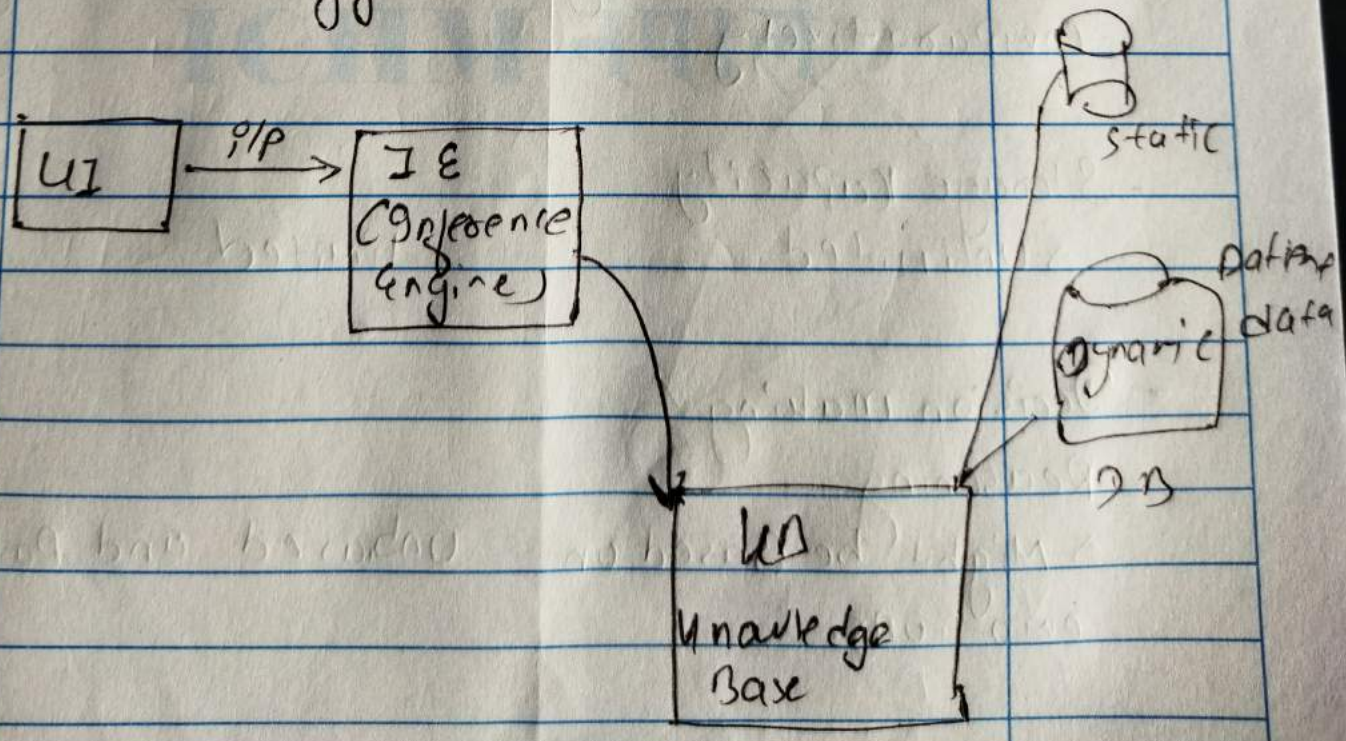
  A → B

  if A, then B

- If symptoms → Disease
  If Disease → Antibiotics

If strain of organism is negative and maphology is rod then then suggest enterobacteriocear.

rod=



Chapter

S.NO.

V.G

* HI vs MI

	Human Intelligence (Natural)	Machine Intelligence (Artificial)
- Operation ability	⇒ Performs declines over time	multitasking with consistent performance.
- Speed:	⇒ Relatively slower	comparatively faster.
~~- Inconsistent acc~~		
- Accuracy:	⇒ Inconsistent accuracy comparatively.	Consistent accuracy.
- Storage Capacity	⇒ unlimited	limited.
- Decision making / Reasoning	⇒ Might be based on emotions.	unbiased and Rational.

Page No.

S.NO.	TOPICS	PAGE NO.
	- Perception = Perceives by parties Perceives in terms of Data sets. n	

7 marks

24 NOV

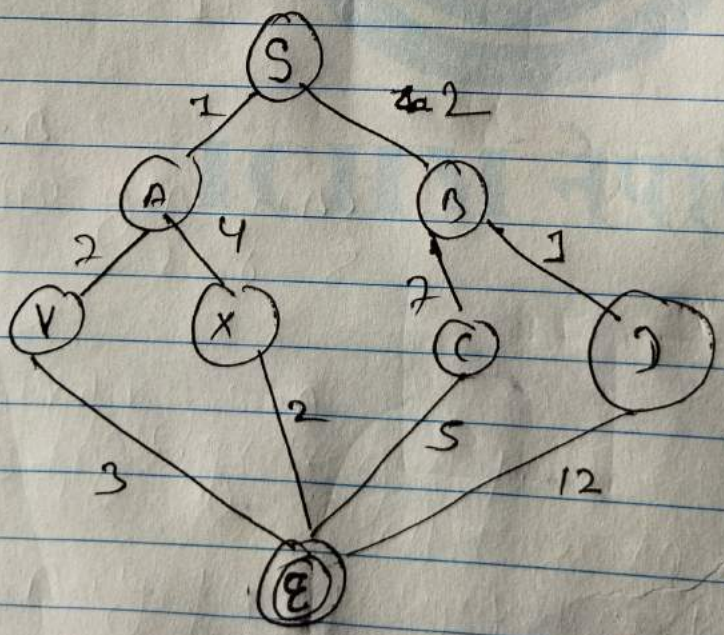
C-4

Inference and Reasoning:

Search Techniques:

1) Blind Search or Uninformed Search.
2) Heuristic Search or Informed Search. — Greedy, A*

Uninformed = no information
Informed = with information (about goal)



Values for h:

A 5
3 6

S.NO. | TOPICS | PAGE NO.

If $\varepsilon = 0$, given then that is goal node

$c = 4$
$D = 15$
$x = 5$
$y = 8$
$\varepsilon = 0$

Greedy Search

where $f(n)$ = evaluation function.
$h(n)$ = heuristic value of node 'n'
= estimated path cost from 'n' to goal.

Start node: S

$S \to A$
$f(n) = h(n)$
$f(A) = h(A) = 5$ ✓

$S \to B$
$f(B) = h(B) = 6$

Expanding $S \to A$

$S \to A \to X$: $f(x) = h(x) = 5$ ✓
$S \to A \to Y$ = $f(y) = h(y) = 8$

Expanding $S \to A \to X$
$S \to A \to X \to \varepsilon$ $f(\varepsilon) = h(\varepsilon) = 0$

∴ $S \to A \to X \to \varepsilon$ is the best path.
(pathCost = 7)

A* Search

S.NO.

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

where, $f(n)$ = evaluation function
$h(n)$ = heuristic value of node 'n'
= estimated path cost from 'n' to goal.

$g(n)$ = path cost to reach node 'n'.

S → A

$f(A) = g(A) + h(A)$
$= 1 + 5 = 6$ ✓

S → B

$f(B) = g(B) + h(B)$
$= 2 + 6 = 8$

Expanding S → A

S → A → x : $f(x) = g(x) + h(x)$
$= (1+4) + 5 = 10$

S → A → y : $f(y) = g(y) + h(y)$
$= (1+2) + 8 = 16$

[S → B has least cost so path and S → B.

Page No.

S.NO.	TOPICS	PAGE NO.

Expanding : S → B.

S → B → C

$f(B) = g(C) + h(C)$

$= 9 + 4 = 13$

S → B → D

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

$= 3 + 15 = 18$

S → A → X has least cost so expand S → A → X.

expanding S → A → X

. S → A → X → E

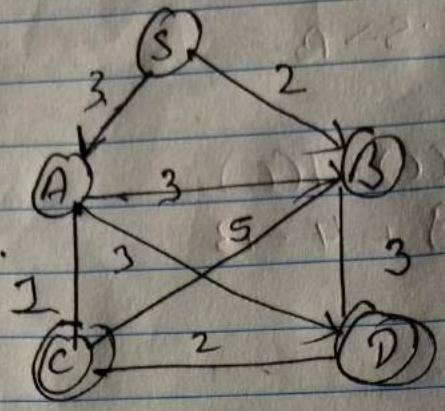
$f(E) = g(E) + h(E)$

$= (1 + 4 + 2) + 0$

$= 7$

∴ S → A → X → E is the best path.

#



h(S)	h(A)	h(B)	h(C)	h(D)
1	3	3	0	0

Values for h:

S	1
A	3
B	3
C	0
D	0

Greedy Search:-

f(n) = evaluation function
h(n) = heuristic value of node 'n'
= estimated path cost from 'n' to goal.

Start node S:

S.NO.

Page No.

S.NO.	TOPICS	PAGE NO.

$S \to A$

$f(A) = h(n)$

$f(A) = h(A) = 3$

$S \to B$

$f(B) = h(B) = 3$

Expanding $S \to A$.

$S \to A \to B$ $\quad f(B) = 3$

$S \to A \to C$ $\quad f(C) = 0$

$S \to A \to D$ $\quad f(D) = 0$

least Path.

expanding $S \to B$

$S \to B \to C$ $\quad f(C) = 0$

$S \to B \to D$ $\quad f(D) = 0$

using A*

$S \to A$: $f(A) = 6$

$S \to B$ $\quad f(B) = 5$

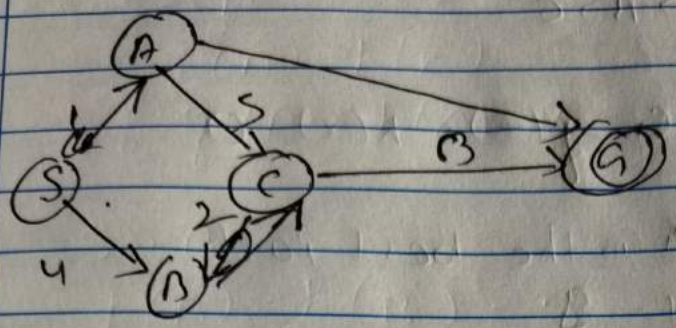
Expanding $S \to B$

$S \to B \to C$ $\quad f(C) = 07$.

$S \to D \to D$ $\quad f(D) = 5$

[$\therefore$ $S \to B \to D$ is the best path]

S.NO. | TOPICS | PAGE NO.



State	h
S	7
A	6
B	2
C	1
G	0

Start node : S

S → A

f(n) = h(n)

f(A) = h(A) = 6

S → B

f(n) = h(n)

f(B) = h(B) = 2

Expanding S → B

S → B → C : f(x) = h(x) = 1

Expanding $S \to B \to C$

$S \to B \to C \to G = f(x) = h(x) = 0$

$\therefore S \to B \to C \to G$ is the best path

path cost = 9

Now,

using A^* search,

$\underline{S \to A}$

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

$= 1 + 6$

$= 7$

$S \to B$

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

$= 4 + 2$

$= 6$

Expanding $S \to B$

$S \to B \to C : f(x) = g(x) + h(x)$

$= 6 + 1$

$= 7$

S.NO.

Since

exi

expa

S.NO.	TOPICS	PAGE NO.

Since $S \to A$ & $S \to B \to C$ has same cost expand both:

expanding $S \to A$.

$S \to A \to C$

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

$= 6 + 1$

$= 7$

ex, $S \to A \to B$

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

$= 3 + 2$

$= 5$

$S \to A \to G$

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

$= 13 + 0$

$= 13$

expanding $S \to A \to C$

$S \to B \to C \to G$

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

$= 9 + 0$

$= 9$

expanding S→A→B

S→A→B→C

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

$= .5 + 1$

$= 6$

Now,

Since S→A→C & S→A→B has same least path cost so expand both.

expanding S→A→B

S→A→C→G

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

$= 9 + 0$

$= 9$

expanding S→A→B

S→A→B→C→G

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

$= 8 + 0$

$= 8$

S→A→B.

S→A→B→C→G

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

$= 8 + 0$

$= 8$

Page No.

S.NO.

28 NOV

S.NO.	TOPICS	PAGE NO.

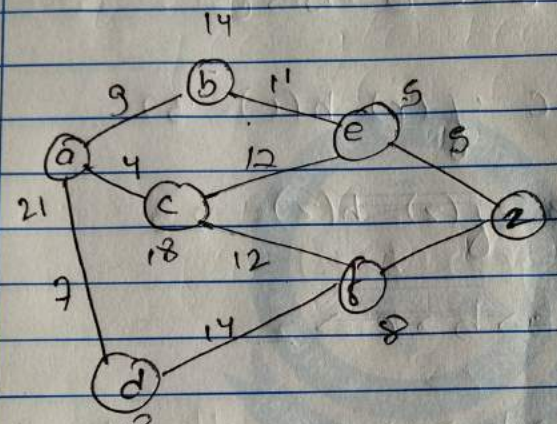
Hence, ~~S → A →~~.

Since S → A → B → C ~~→ G~~ has least cost

so,

S → A → B → C → G is the best path.

28 NOV



Start node : a.

a → b

$f(n) = h(n)$

$f(b) = h(b) = 14$.

a → c

$f(n) = h(n)$

$f(c) = h(c) = 18$

a → d

$f(n) = h(n)$

$f(d) = h(d) = 18$

Expanding a→b

a→b→e : f(e) = h(e) = 20

~~a→b→~~

expanding, ~~a→e~~ a→b→e

a→b→e→c : f(c) = h(c) = 37

a→b→e→z f(z) = h(z) = 25

~~expanding a→b→e→z~~

~~a→~~ Hence a→b→e→z, is the final path.

Now using A*

Start node a

a→b :

f(b) = g(b) + h(b)

: 9 + 14

= 23 ✓

S.NO. | TOPICS | PAGE NO.

a → d

f(d) = g(d) + h(d)
= 7 + 18
= 25

a → c

f(c) = g(c) + h(c)
= 4 + 18
= 22 ✓

Since, a → c has least path cost so we expand a → c

a → c → e

f(e) = g(e) + h(e)
= 21 + 5
= 26

a → c → f

f(f) = g(f) + h(f)
= 16 + 8
= 24 ✓

Since a → b has the least path cost so we expand
a → b.

a→ b→ e

$f(e) = g(e) + h(e)$
$= 20 + 5$
$= 25$

~~a→b~~

Since, a→c→f has the least path cost so, we expand d

a→c→f

a→c→f→z   a→

$f(z) = g(z) + h(z)$
$= 25 + 0$
$= 25$

Hence the best path cost is 25 and the path is a→c→f→z.

a→c→f→d

$f(d) = g(d) + h(d)$
$= 30 + 18$
$= 48$

16
4
30

Now,
[we h
a→
exp
e
ex
He
pa

S.NO.	TOPICS	PAGE NO.

now,

[we have to check whether a→b→c & a→c has least path]

expanding a→d

a→d→f.

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

$= 21 + 8$

$= 29$

expanding a→b→e.

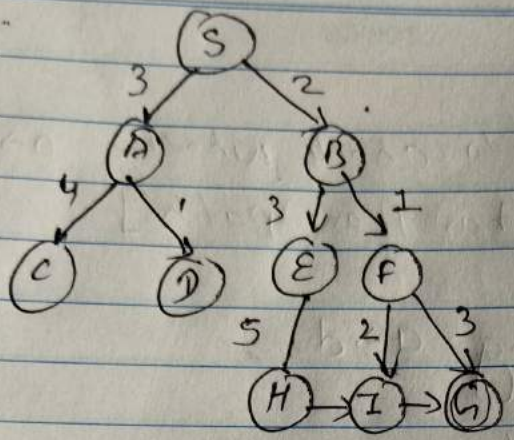
a→b→e→z

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

$= 25 + 0$

$= 25$

Hence, a→c→f→z and a→b→e→z is the best path cost.



node	h(n)
A	12
B	4
C	7
D	3
E	8
F	2
H	4
I	9
S	13
G	0

Using Greedy :

expanding S

S → A

$f(A) = 12$

S → B

$f(B) = 4$ ✓

expanding S → B → E

$f(E) = 5$

expanding S → B → F

$f(F) = 3$ ✓

S.NO.	TOPICS	PAGE NO.

expanding S → B → F

S → B → F → I : f(I)
= 5

S → B → F → G : f(G)
= 6

Hence S → B → F → G is the best path.

Now, A*

expanding S

S → A : f(A) = g(A) + h(A)
= 3 + 12
= 15

S.NO.	TOPICS	PAGE NO.

Romania question

Goal node: Buchorest

Starting node: Arad.

A → B shortest path.?

Greedy:

A → Z : f(Z) = h(Z) = 374
A → T : f(T) = 329
A → S : f(S) = 253 ✓

expanding A → S

A → S → F : f(F) = 178 ✓
A → S → R : f(R) = 193

expanding A → S → F

A → S → F → B : f(B) = 0

∴ A → S → F → B is the shortest path.

(140 + 99 + 211)
= 450

A*

A→Z : f(Z) = 449
A→T : f(T) = 329 + 118 = 442
A→S : f(S) = 393

expanding A→S

A→S→F : f(F) = g(F) + h(F) = 417
A→S→R : f(R) = 413

Expanding A→S→R

A→S→R→C : f(C) = 526
A→S→R→P : f(P) = 415

expanding A→S→R→P.

A→S→R→P→C f(C) = 615
A→S→R→P→B f(B) = 418

Expanding A→S→F

A→S→F→B f(B) = 450

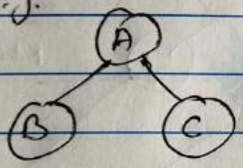
∴ A→S→R→F→B is the best path. (418)

S.NO.	TOPICS	PAGE NO.

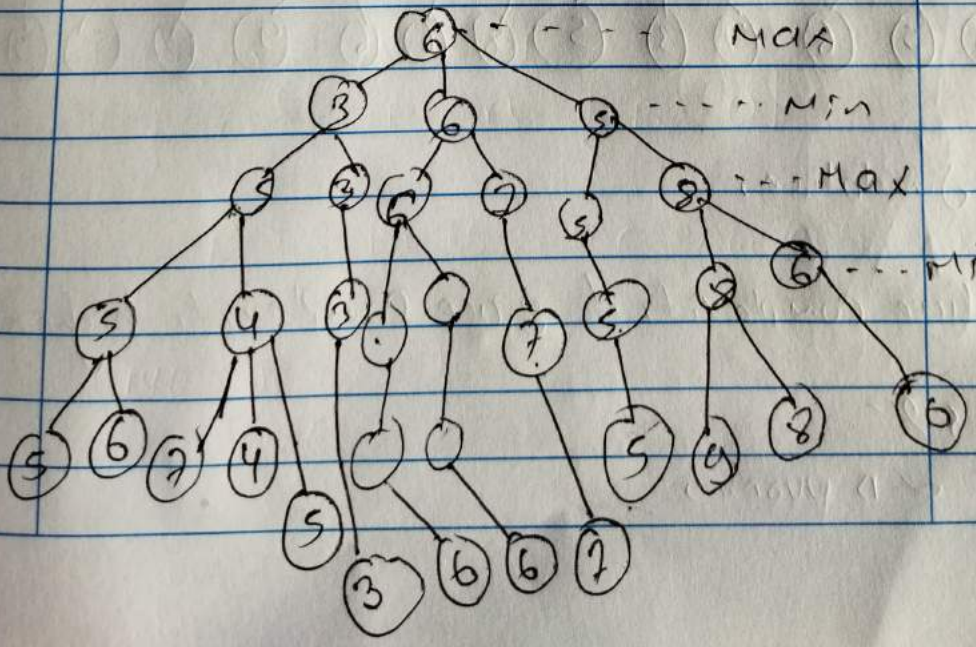
MinMax Algorithm:

Greedy	A*
1) $f(n) = h(n)$	$f(n) = g(n) + h(n)$
2) Heuristic value	Also, considers path cost
3) Not required	Backtracing
4) Simple & faster	Comparatively slower
5) Comparatively lower	Time complexity & Space complexity higher.

e.g:



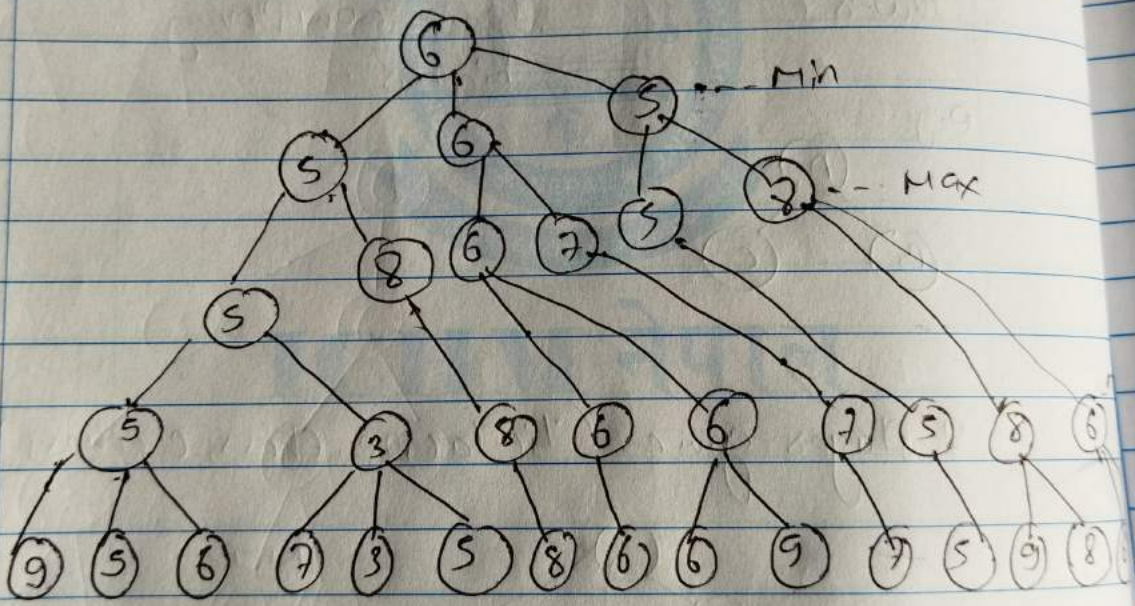
2-player games (Tic-Tac-Toe, Chess, Checkers)



MinMax Algorithm

→ Game playing algorithm used in 2 player game. (Tic Tac Toe, chess)

→ Two players: Maximizer: Selects Max value
Minimizer: Selects Min value



Limitations:

- Time complexity & space complexity is higher.

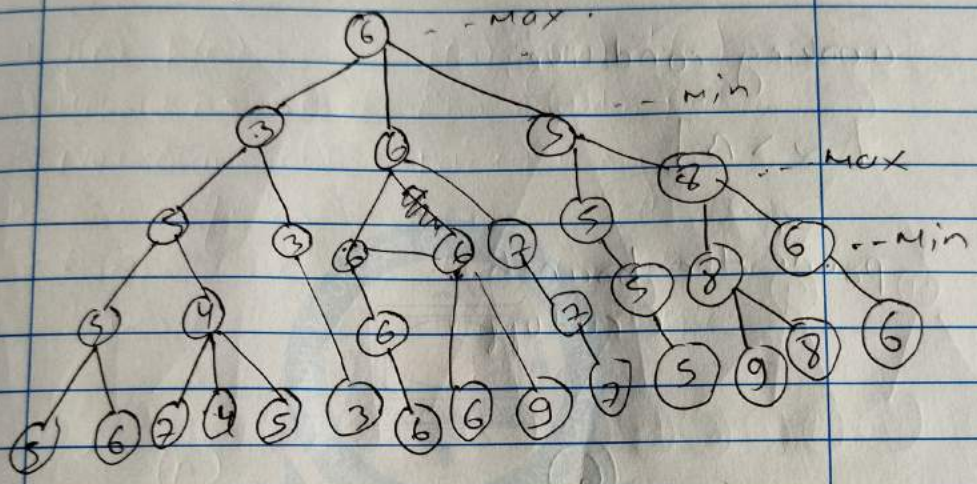
Solⁿ

Page No. α-β pruning

S.NO.	TOPICS	PAGE NO.

Alpha-beta pruning overcomes these limitations.

MinMax Algorithm



2 players games
- Maximizer: Selects max value
- Minimizer: " min value.

Limitations:
- Time complexity and Space complexity higher.

α-β Pruning (Alpha-Beta Pruning)

Initial value (worst case value)

$\alpha = -\infty$

$\beta = \infty$

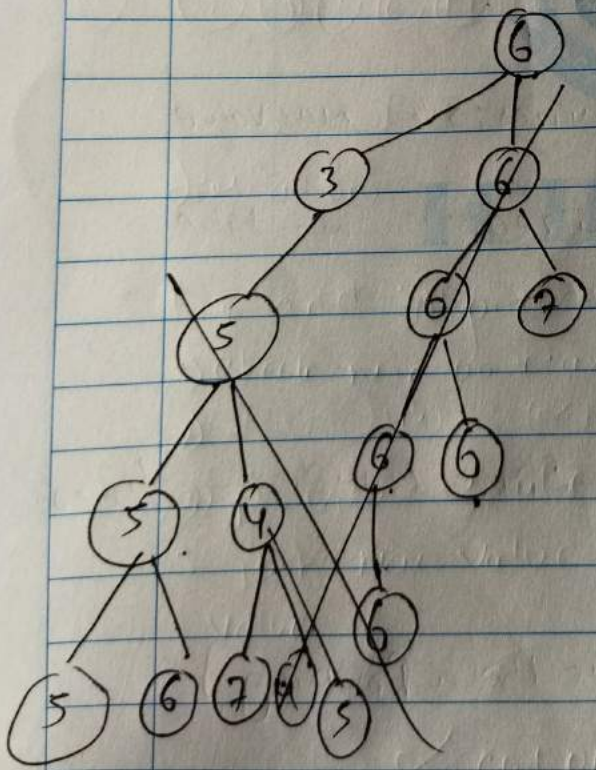
(β) (∝)

1) Check min or max level
2) Update ∝-β values
3) Push min max value.

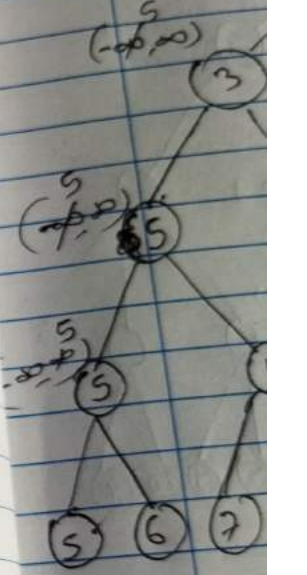
working condition:

$[\alpha \geq \beta]$

prune the branch
(cutoff)

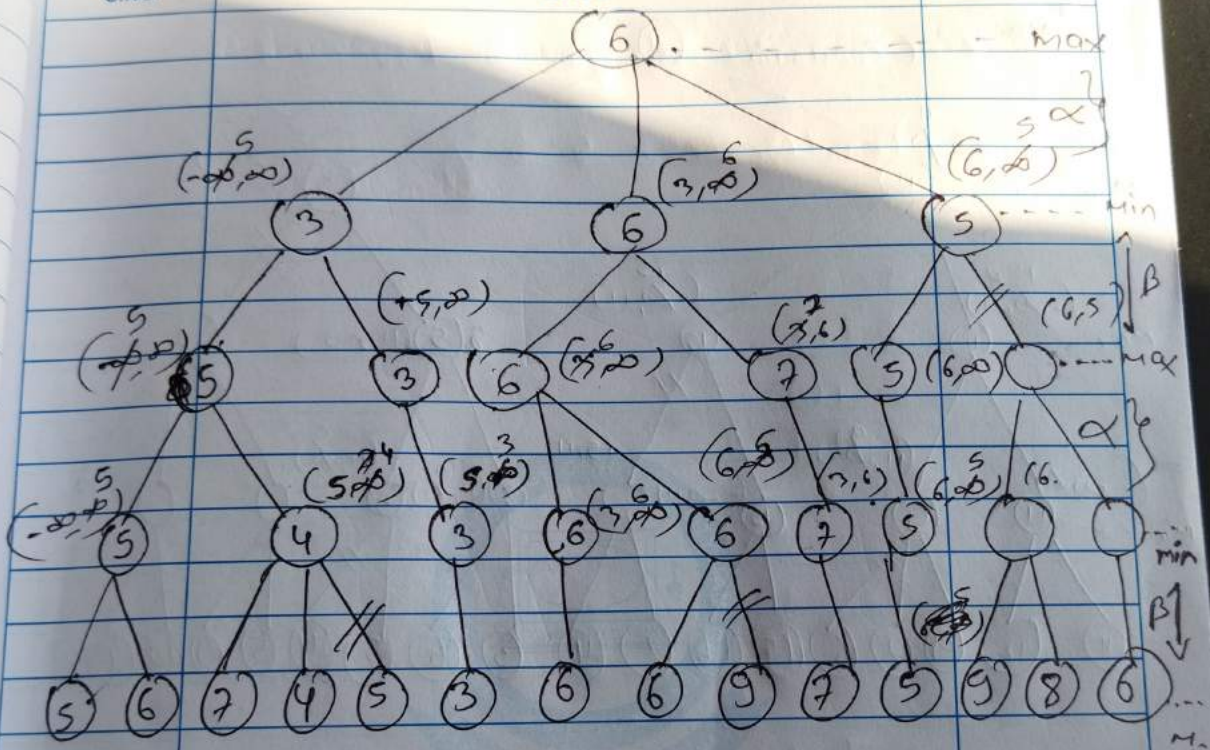


S.NO.



Page No.

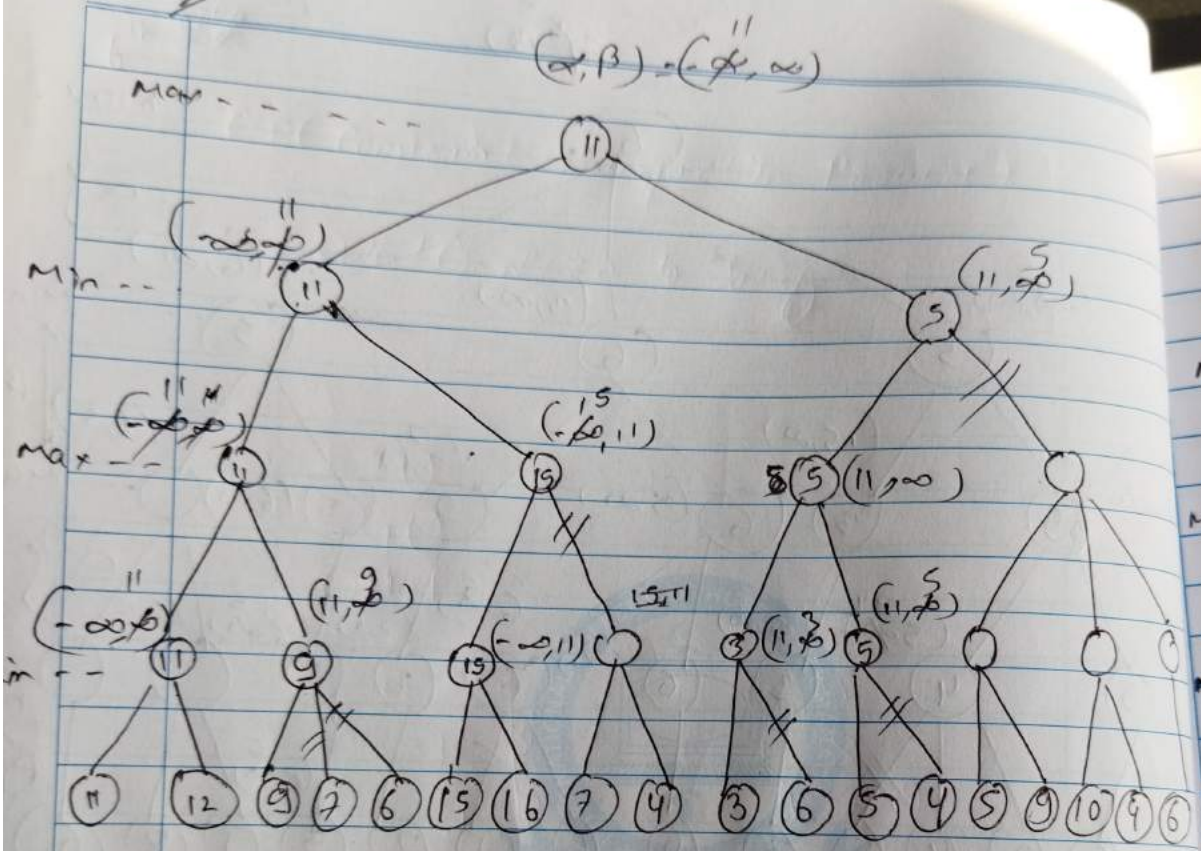
S.NO. | TOPICS | PAGE NO.



α-cutoff = 0

β-cutoff = 3

5th dec.

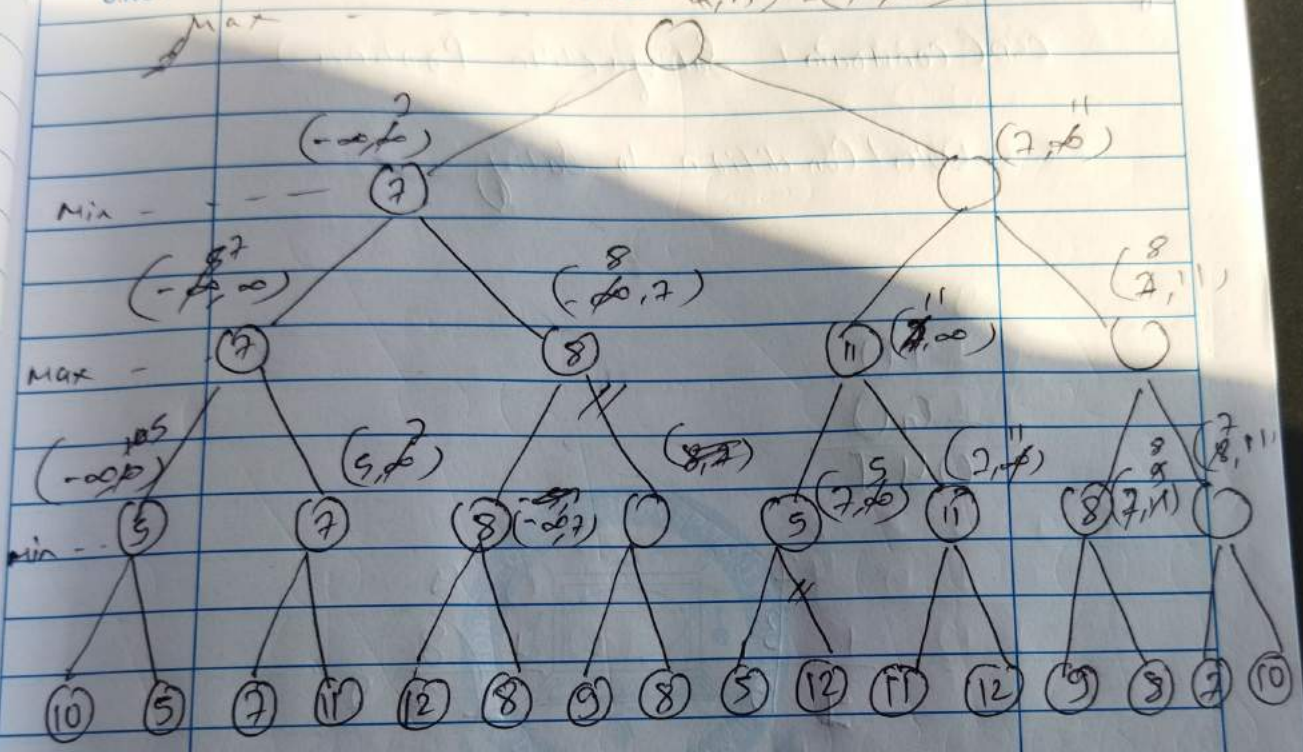


limitations:

- Time and Algorithm
  1) Check min or max level
  2) (β) (α)
  2a) update α val-e if in α-level else update β val-e
  3 push max value in α-level else push min value.

α-cut off = (Max) : 7
β-cut off = (min) : 5

S.NO. TOPICS PAGE NO.



S.NO.	TOPICS	PAGE NO.
	)(Q k [face card]	
	Uncertainity with Bayes' Rule	
	$P(A/B) = \frac{P(A \cap B)}{P(B)}$	
	$P(\text{king}/\text{Face}) = \frac{P(\text{king} \cap \text{face})}{P(\text{face})}$	
	$= \frac{4/52}{12/52} = 1/3$	
	what is uncertainity? sources? How to overcome? Baye's Rule? Eg?	
	=> Unable to predicted outcome / conclusion.	
	Sources of uncertainity: (insufficient)	
	1) Incomplete data / unknown data (underfitting)	
	2) Vague data / Redundant data. (Overfitting) / weak implication	
	3) Imprecise language:	

use of ambiguous words (hardly ever, rarely, often, sometimes)

4) Combining / multiple views of experts.

Using Baye's Rule we can overcome uncertain

Baye's Rule / Bayesian Network / BBN (Bayes Belief Network)

- handles uncertainity / uncertain data.
- deals with conditional probability, probabili of an event 'A'

$P(A/B)$ given that 'B' has occured prior.

$$P(A/B) = \frac{P(A \cap B)}{P(B)} \quad \text{— (i)}$$

$$P(A \cap B) = P(A/B) \cdot P(B)$$

S.NO.

Page No.

S.NO.	TOPICS	PAGE NO.

$$P(B/A) = \frac{P(B \cap A)}{P(A)} \quad \text{— (2)}$$

$$P(A \cap B) = P(B/A) \cdot P(A)$$

$$\boxed{P(A/B) = \frac{P(B/A) \cdot P(A)}{P(B)}} \quad \text{— (3)}$$

While watching a game of Champion league football in a cafe, you observe someone who is clearly supporting manchester united.

- what is the probably that they were "born" actually within 25 Miles of manchester.

Assume that the probability that a randomly selected person in that bar is born with in 25 mile of manchester is $1/20$

The chance that a person born within 25 mile of manchester actually supports manchester united is $2/10$?

The probability that a person not born with in 25 mile of manchester supports manchester united with probability $\frac{1}{10}$?

Sol^{n}

Let S & B denote,

S = Supports MU

B = Born within 25 Miles of Manchester.

$\bar{B}$ = not born

by question,

$P(B) = \frac{1}{20}$ $\quad : 1 - \frac{1}{20} = \frac{19}{20}$

$P(S/B) = \frac{7}{10}$

$P(S/\bar{B}) = \frac{1}{10}$

Now,

$$P(B/S) = \frac{P(S/B) \cdot P(B)}{P(S)}$$

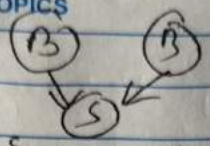
Now $P(S) = ?$

S.NO.

Page No.

$(S/\bar{B})$

S.NO. | TOPICS | PAGE NO.



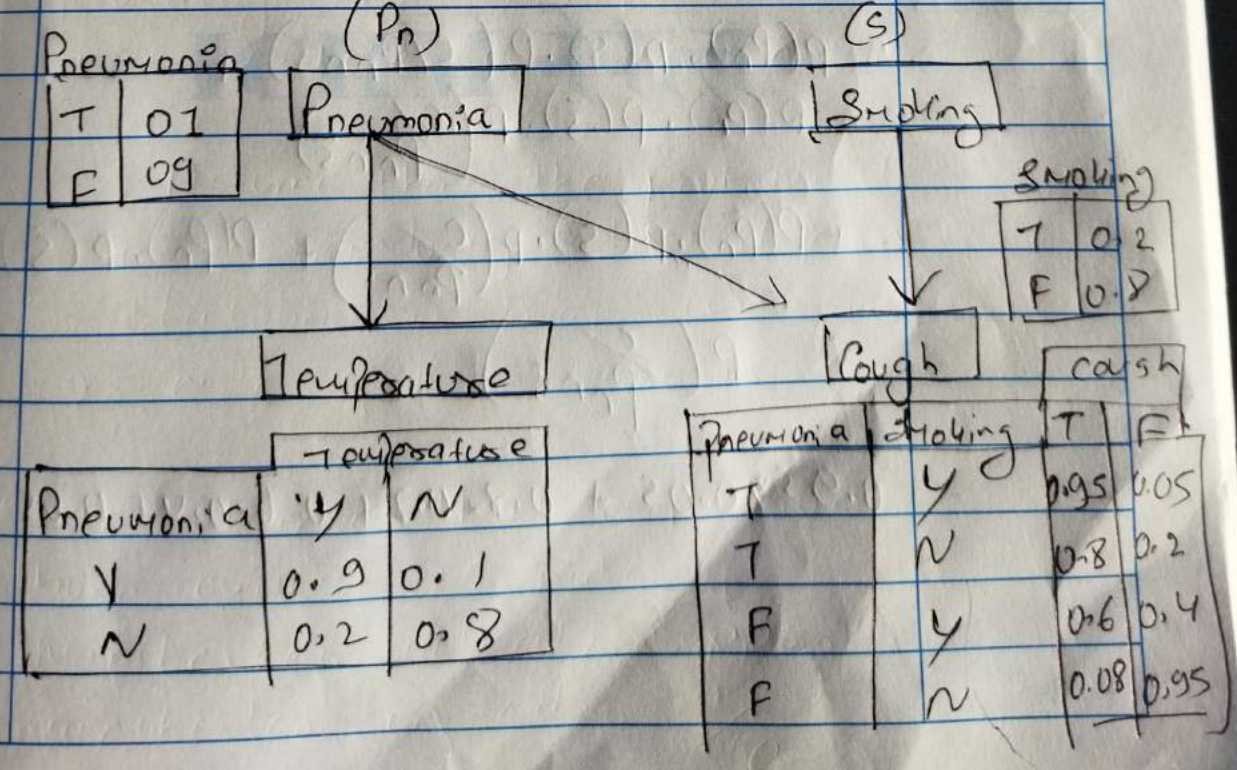
$$P(S) = P(B) \cdot P(S/B) + P(\bar{B}) \cdot P(S/\bar{B})$$

$$= 1/20 \cdot 7/10 + \frac{19}{20} \times \frac{1}{10}$$

$$= \frac{7}{200} + \frac{19}{200}$$

$$= \frac{26}{200}$$

Bayesian Network. (Not for exam just for understanding)



1) $P(C/S \cap P_n)$
2) $P(C)$
3) $P(C/S)$

1)

$P(C/S \cap P_n)$ = Probability of cough due to smoking & Pneumonia

$= 0.95$

$P(C)$ = prob of cough

$= \sout{P(P)\ P\left(\frac{C}{P}\right)} +$ ↗ (Pneu [illegible] smoking [illegible] 2 [illegible] cough [illegible])

$= P(P_n) \cdot P(S) \cdot P(C/P_n \cap S) +$
$P(P_n) \cdot P(\bar{S}) \cdot P\left(\frac{C}{P_n \cap \bar{S}}\right)$

$+ P(\bar{P_n}) \cdot P(S) \cdot P\left(\frac{C}{\bar{P_n} \cap S}\right) + P(P_n) \cdot P(\bar{S}) \cdot$
$P\left(\frac{C}{\bar{P_n} \cap \bar{S}}\right)$

$= 0.95 * 0.05 + 0.8 \times 0.2 + 0.6 \times 0.4 + 0.05 \times 0.95$

$=$

S.NO.

iii) $P(C/S) = P$

$= P(P_n)$

$=$

iv) $P(C/P_n) =$

only smoking आते आते
P=F S=Y

S.NO.	TOPICS	PAGE NO.

iii) $P(C/S)$ = Prob of cough due to smoking.

$= P(P_n) \cdot P(S) \cdot P(C/P_n \cap S) + P(\bar{P_n}) \cdot P(S) \cdot P(C/\bar{P_n} \cap S)$

=

iv) $P(C/P_n) = P(P_n) \cdot P(\bar{S}) \cdot P(C/P_n \cap \bar{S})$

$+ P(P_n) \cdot P(\bar{S}) \cdot P(C/P_n \cap \bar{S})$

$= 0.95 \times 0.05$

1) Unary Operator ($\neg$) : single operand

2) Binary operator ($\wedge, \vee, \rightarrow, \leftrightarrow$) : two operands

2) Logic
- Propositional Logic
- Predicate Logic

(UT) 15 Dec (Wed) PTMV (7-8:30)

Truth Table:

P	Q	$\neg P$ (Not)	$\neg Q$ (Not)	$P \vee Q$ (Or / Disjunction)	$P \wedge Q$ (And / Conjunction)	$P \rightarrow Q$ ("if-then" / Conditional / Implication)	$P \leftrightarrow Q$ (Biconditional / Double Implication / "if and only if")
T	T	F	F	T	T	T	T
T	F	F	T	T	F	F	F
F	T	T	F	T	F	T	F
F	F	T	T	F	F	T	T

* $P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$

* $P \rightarrow Q \equiv \neg P \vee Q$

Logically equivalent

$P \rightarrow Q$

Hypothesis → Conclusion

Condition → Action

Antecedent → Consequent

P : "It is raining"

Q : "I will take an umbrella"