1	2	3
4	5	6
7	8	X

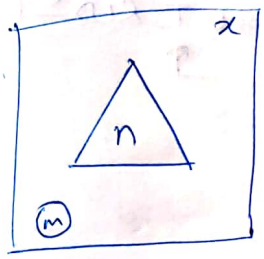
1	2	3
8	X	4
7	6	5

← Goal

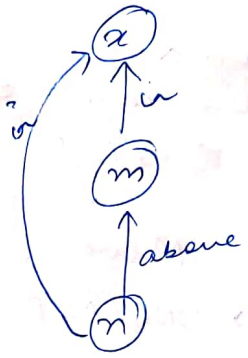
Blank piece ←

X	1	3
8	2	4
6	7	5

← Initial



m in x
n in x
m below n

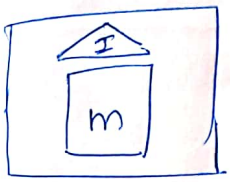


helps in natural language processing →

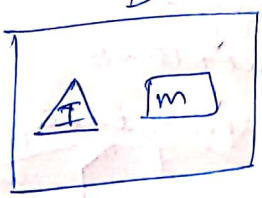
"You have to put it into form such that the machine understands"

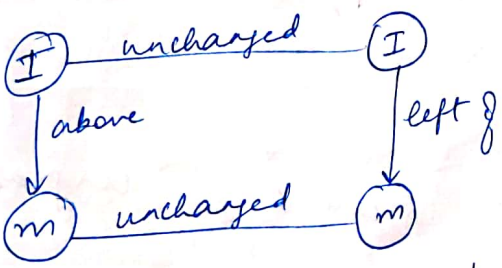
Given

A



B





← semantic form.

Reasoning - Rule based expert systems

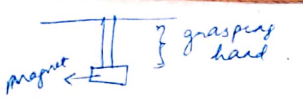
→ Forward chaining
→ Backward chain

Ex: comes in medical systems

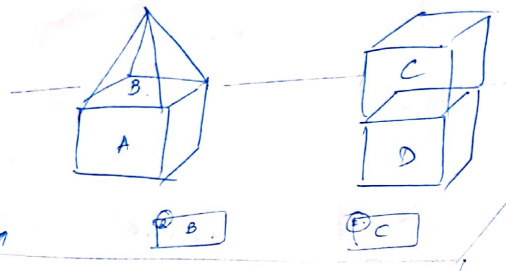
Asks several questions

if all the symptoms are given to the PC then it recognises the prob (like cold) → suggested medicines

↳ THIS IS FORWARD CHAIN.



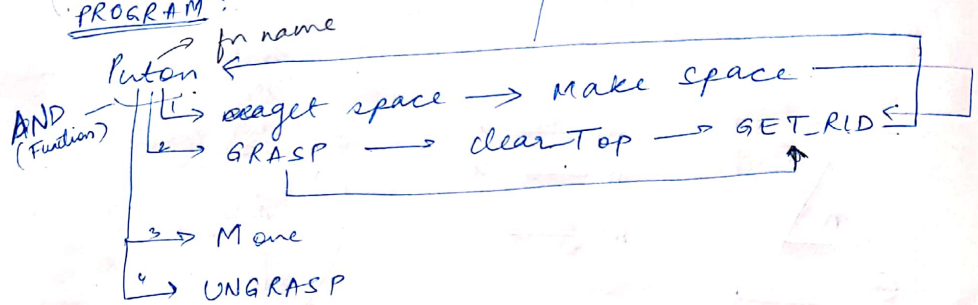
Reduction Method

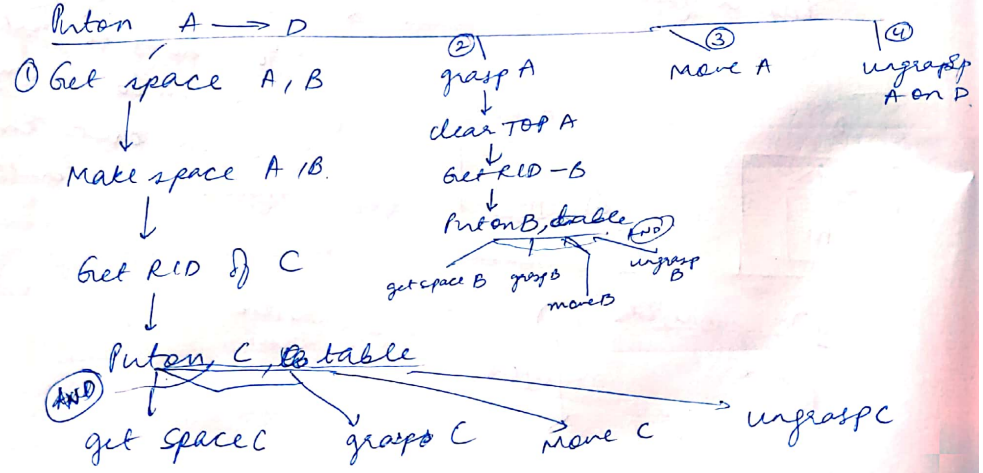


Objective:
Put move A on D

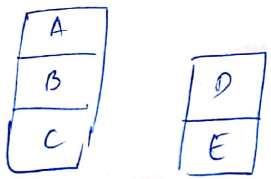
1. Move C on Table
2. Move B on table
3. Move A on D.

PROGRAM:





8.



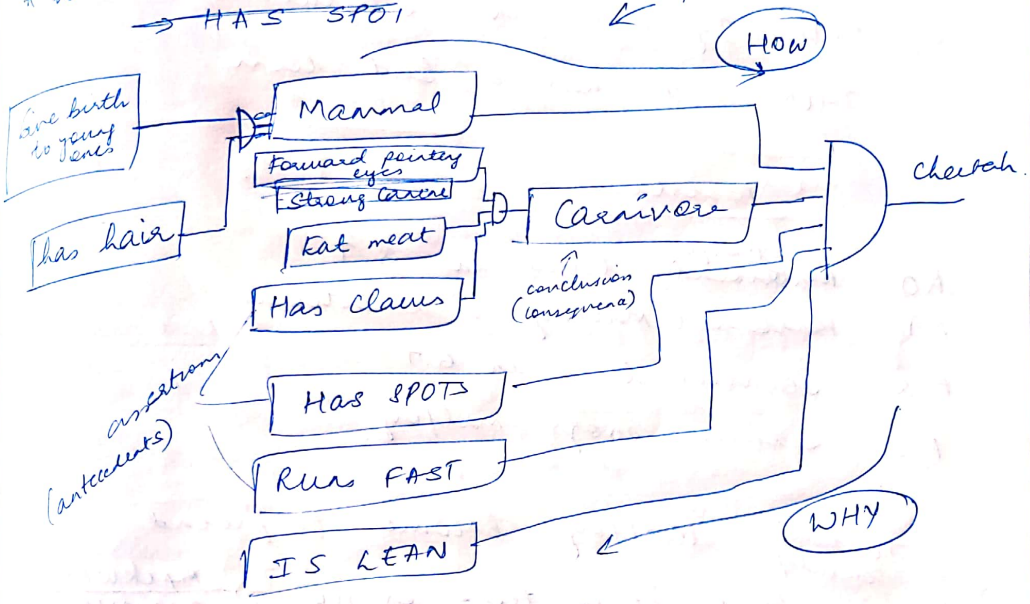
→ Put E on C

∨ – OR

∧ – AND

* How do you identify a cheetah? →

~~→ HAS SPOT~~ ← FORWARD CHAINING



why is it a cheetah? → Backward chaining

RULES :

P0 : If ?x is ambitious
AND ?x is a squib
THEN ?x has a bad term

P1 : If ?x lives in ~~Slytherin~~ Gryffindor Tower
THEN ?x is PROTOGANIST

P2 : If ?x lives in Slytherin Tower
THEN ?x is a villian } 2 outcomes
?x is ambitious

P3 : If (?x is a protagonist OR ?x is a villian)
AND (?x is ambitious)

THEN ?x studies a lot

P4: If (?x studies a lot AND ?x is a protagonist)
THEN ?x becomes Hermoine's friend

P5: If ?x snog ?y
AND ?x lives in Gryffindor Tower
AND ?y lives in Slytherin Tower (S.T)
THEN ?x has a bad team.

Assertions

A0: ~~Nilticat~~ Malfoy lives in S.T
A1: ~~Nilticat~~ Malfoy is ambitious
A2: Seamus lives in G.T
A3: Seamus snogs Malfoy.

To prove/contended: Malfoy becomes H friend
Proving that it is true or not => BACKWARD CHAINING.

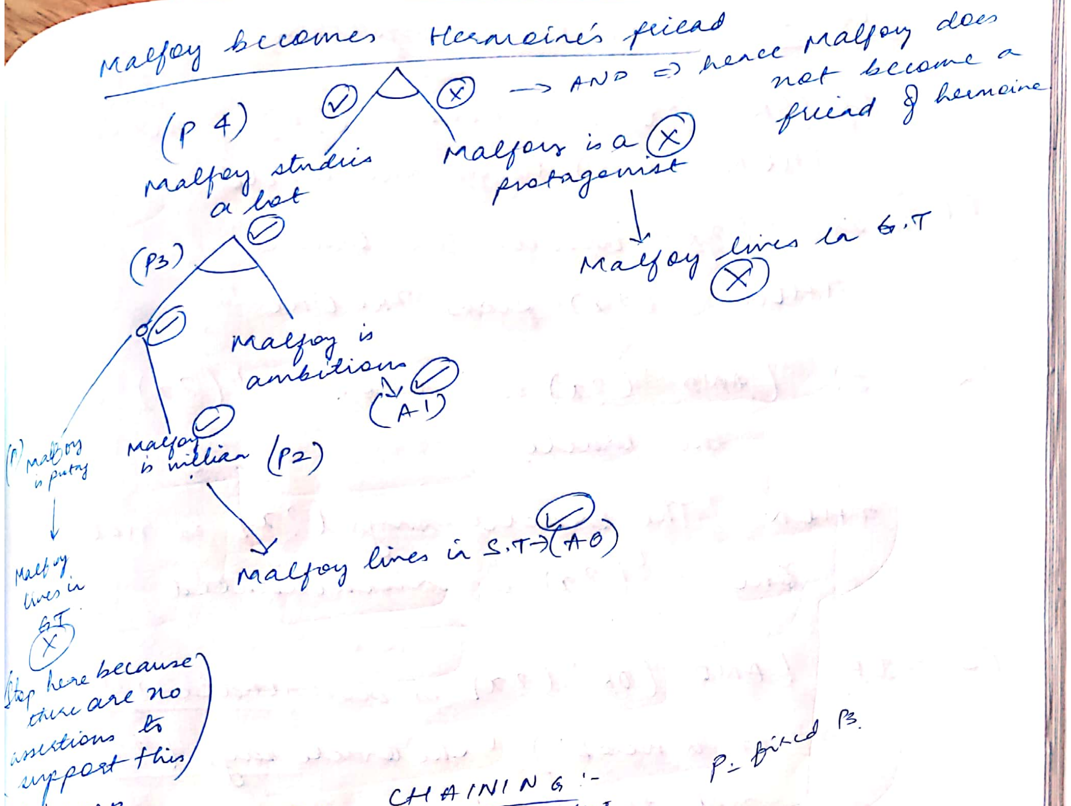
Now, x = Malfoy
y = Harmoine

ALWAYS CHECK ASSERTION FIRST THEN RULE

Malfoy become Harmoine's friend

depth 1st search

Malfoy studies a lot Malfoy is a protagonist



19/1/2020

FORWARD CHAINING :-

have to look at antecedent

P = fired P3.

Situation	Matches	Fired	New assertions
1.	P2, P1, P5	P2	A4 : Malfoy is a villian
2.	P3, P2, P1, P5	P3	A5 : Malfoy studies a lot
3.	P2, P3, P1, P5	P1	A6 : Seamus is a protagonist.
4.	P2, P3, P1, P5	P5	A7 : Seamus has a bad term.

Since the statement "Malfoy becomes H friend" is not come out as a 'new assertion', hence ans is : Malfoy cannot become H's friend.

PROBLEM 2 : Rule based systems

P0 : If (OR '(?x) is a noob', '(?x) is disenchanted')
THEN '(?x) visits the Oracle'

P1 : If '(?x) visits the Oracle'
THEN '(?x) picks The One'

P2 : If (AND '(?x) is a noob', '(?x) visits the Oracle')
THEN 'The Oracle says (?x) is not the One', '(?x) is disenchanted

P3 : IF (AND (OR '(?x) is disenchanted', '(?x) is a noob'), 'The Oracle says (?x) is not the One')
THEN '(?x) practice Kung Fu'

P4 : If (AND '(?x) is disenchanted', '(?x) love (?y)')
'(?x) proclaims love for (?y)'
THEN : ~~'(?x) is disenchanted~~

P5 : IF (AND 'The Oracle says (?x) is not the One', '(?y) proclaims love for (?x)'
'(?y) picks the One')
THEN '(?x) is the One'

Forward chaining

→ (some kind of explain is expected for the 4 exam (Rules))

Iteration	Matches	Fire	New Assertion
1	P0, P4.	P0	A4 : Neo visits th… A5 : Trinity visits the…
2.	P0, P1, P2, P4	P1	A6 : Neo picks the… A7 : Trinity picks the…
3.	P0, P1, P2, P4	P2	A8 : The Oracle says … is not the One A9 : Neo is disenchan…
4.	P0, P1, P2, P5, P4	P5	A10 : Neo is the One

See PPT 8. To prove : Fido is a dynomicans.

→ Describe and match.

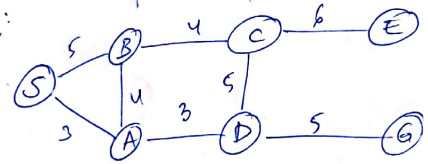
Ex. →

→ Means and analysis.

Ex:

Goa 600km	walking	cycling	2wheeler	Train	Airplane
	30 day	15 day	3 day	1 day	

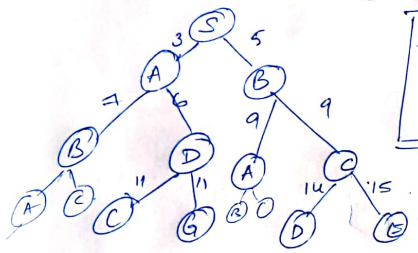
fig! :



Search Algorithms :-

1. British Museum } Blind folded search
2. DFS
3. BFS
4. Branch & Bound
5. BB Beam width
6. Extended dist
7. A*
8. AO*
9. Hill climbing

① British Museum search



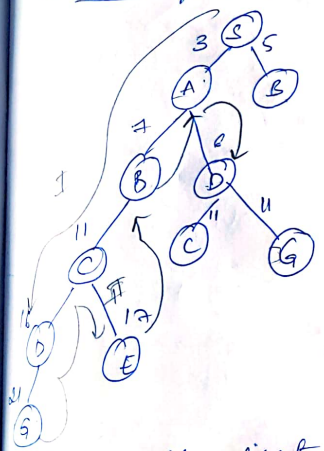
lexically you'll have solve. ie A & B have same net, then you should take A.

— Blind search
— Mean's analysis

Since count through C = 15 > 11 so the search stops.

- will have 2 queues:
  ① Path Queue = [ ]
  ② Path traverse

② depth first search (DFS) → always goes left.



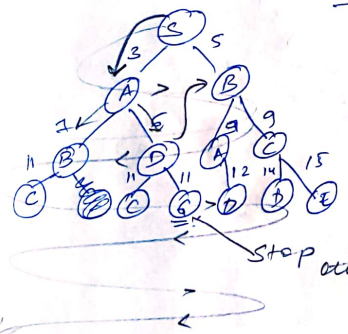
→ Blind search

[S, A, B]
[S, A, B, D, B]
[S, A, B, C, D, B]

→ *(DFS is more blind than BFS.)

③ Breadth first search (BFS)

→ Blind search



[S, A, B]
[S, A, B, B, D, A, C]
[S, A, B, B, D, A, C, C]
[S, A, B, B, D, A, C, C, D]

Stop because other nodes already >11 & if you add anything to that, obviously it'll be >11. So stop here.

[ppt → diagram for DFS, BFS. >

from 's' to 'w'

[S A B C]

[S A D E B C]

[S A D K L E B C]

[S A D K U V L E B C]

[S A D K U K V L E B C]

[S A D L W X E B C] → Goal path, not traversed path

from 's' to 'G'

DFS

S A B C

S A D E B C

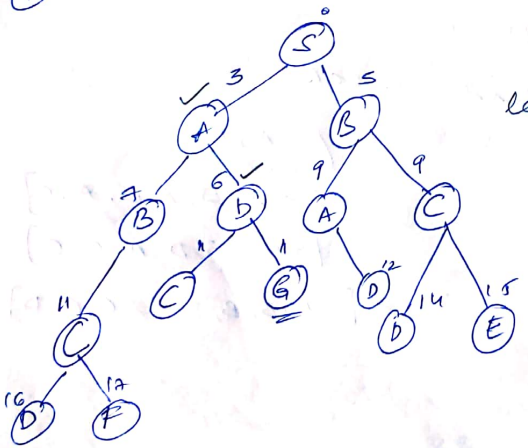
S A D K L E B C

S A D K U V L E B C

S A D K U V L W X E B C

⋮

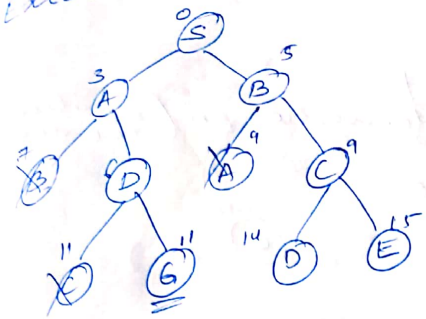
④ Branch and Bound. (see fig 1)



least dist = A (3)

least dist = D (6)
then B (7)
then A (9)
C (9)

Extended dist

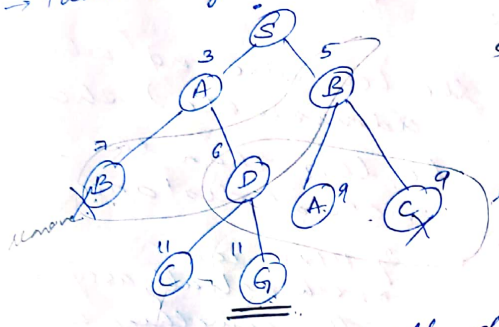


0 3 5 6 | 7 9 9 | 11 11 14 15
S A B D | B A K | C G D E

"whatever I (node) have traversed, I will not traverse through it again"

Optimal.

(3) BB Beam width — (only path) — No Heuristic
→ Take only 2 nodes at a time → B = 2.



5, 6, ~~7~~. least 2.

lexically removed

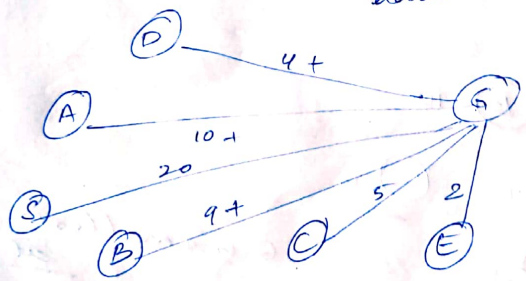
→ In Test, give a small description before solving the problem. about the method.

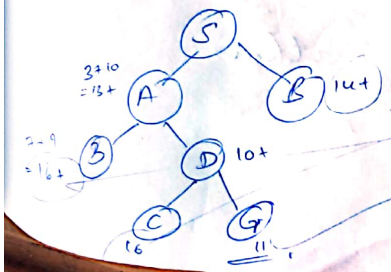
(4) Hill climbing :-

(see fig 1)

Fairline distances.

fig 2 :



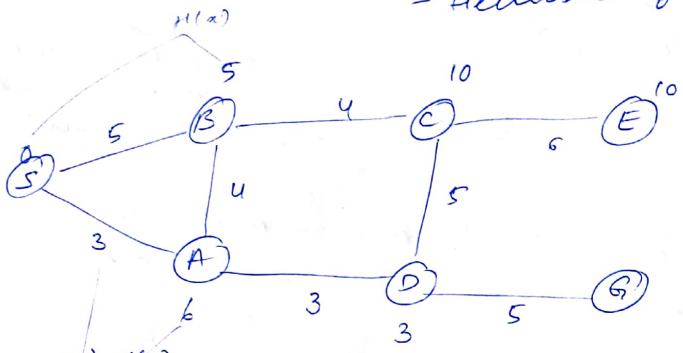


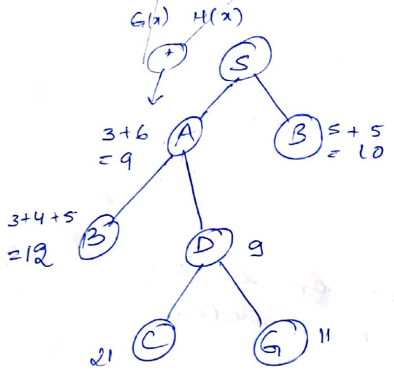
bigger than 11. So
→ stop here

(8) A*

$$f(x) = G(x) + H(x)$$

→ Heuristic fn (applied at the node)

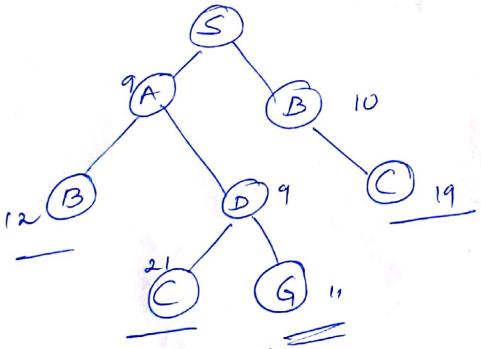




the heuristic no (which is admissable) should be less than the actual distance.

If H(x) = 100
* work actually assigned etc
Then something is wrong

now B => (10) < G(11)
So continue.



G is smallest & arrived at the goal.

other searching methods/algorithms:
* annealing
* Travel salesman
* ant colonisation

Initial state

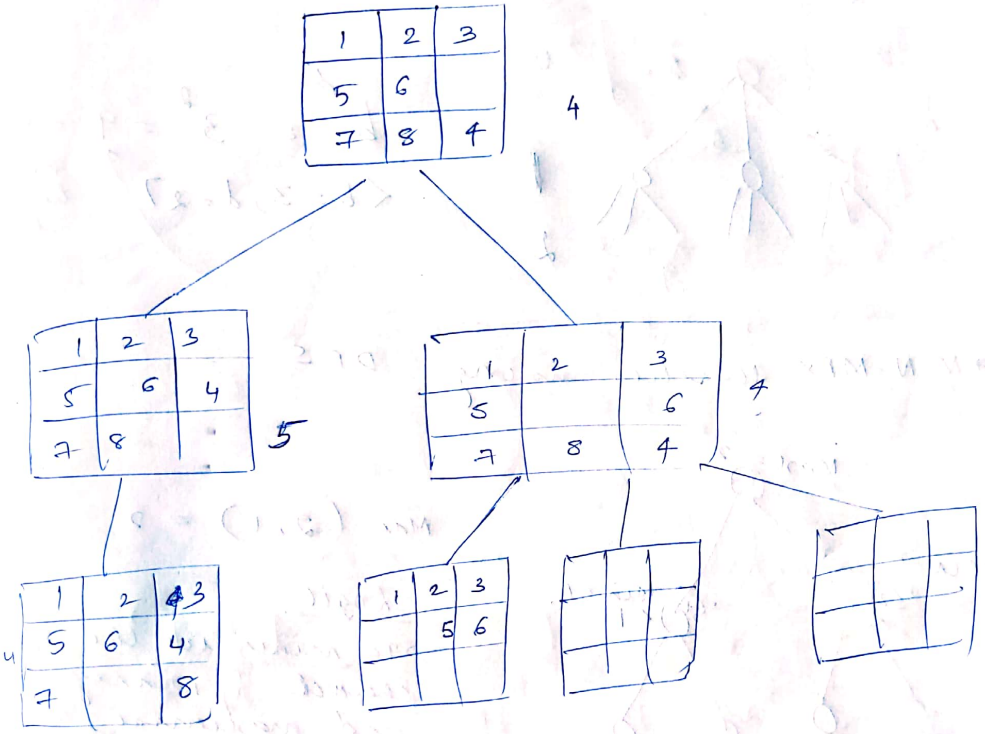
1	2	3
5	6	
7	8	4

⟹

final state

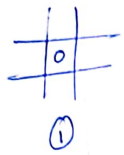
1	2	3
8		4
7	6	5

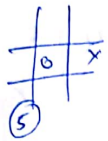
→ use DFS to reach final state

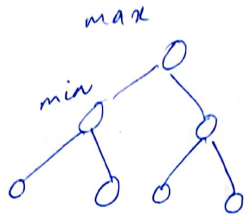


Unit 1 → (Chapt 2 & 3) of Textbook.

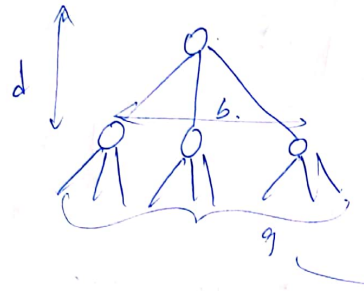
* MIN-MAX







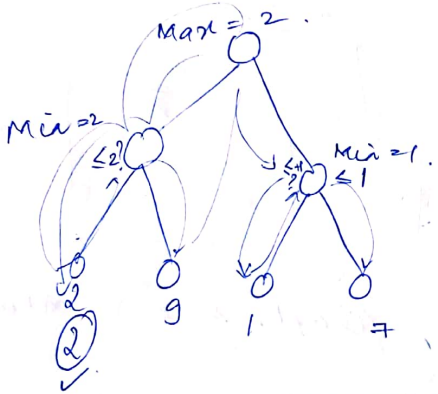
$S = g(c_1 f_1 + c_2 f_2 + c_3 f_3 + \dots + c_n f_n)$ = static fn



$b^d = 3^2 = 9$ = Game no. nodes/ties at level

$\langle b = 3, d = 2 \rangle$

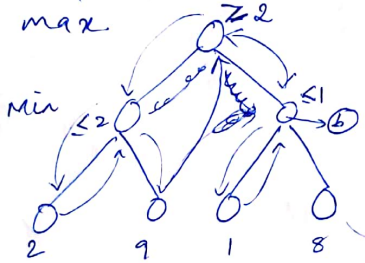
*MIN-MAX searches using DFS.



Max (2, 1) = 2

Logic:
→ To minimise the opponent chances of winning and maximise my chance of winning

② α-β pruning



max is searching for ≥ 2
but in node (b), it is looking for ≤ 1 so obviously, you don't have to visit node = 8

So, one node is less traversed "pruning"

Magic squares

10	3	8
5	7	9
6	11	4

3 × 3

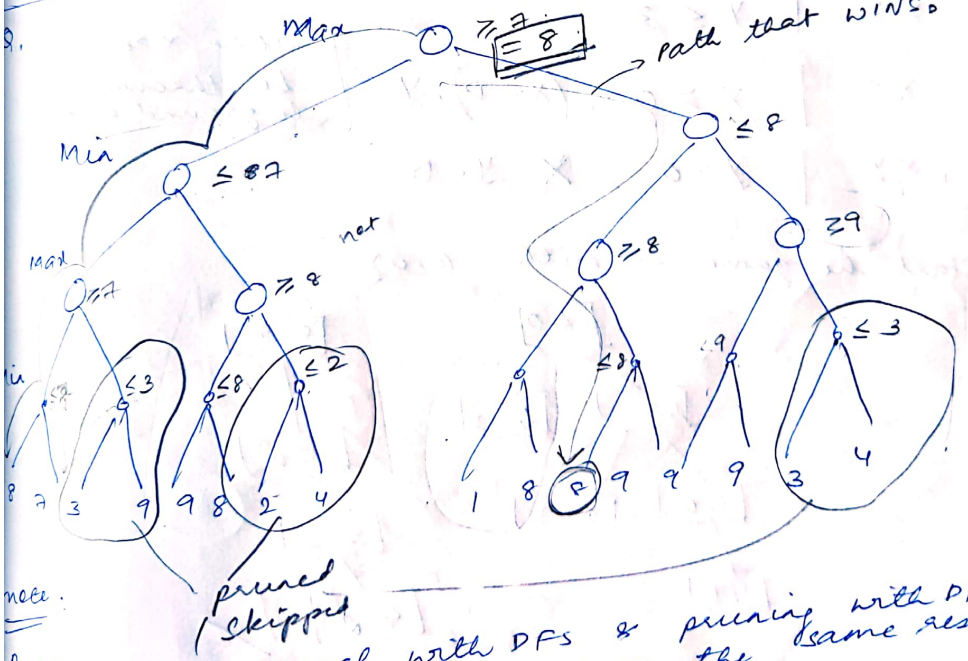
$\frac{15}{3} = 15 - 4 = (1)$

$\frac{21}{3} = 7 - 4 = (3)$ starting no

each row / col sum = 21.

for (even no × even no) logic is diff.

2.



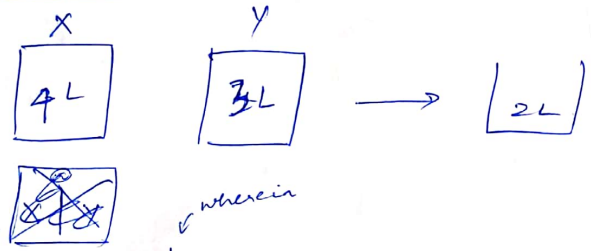
note:

Both british search with DFS & pruning with DFS methods will produce the same result.

Rules:

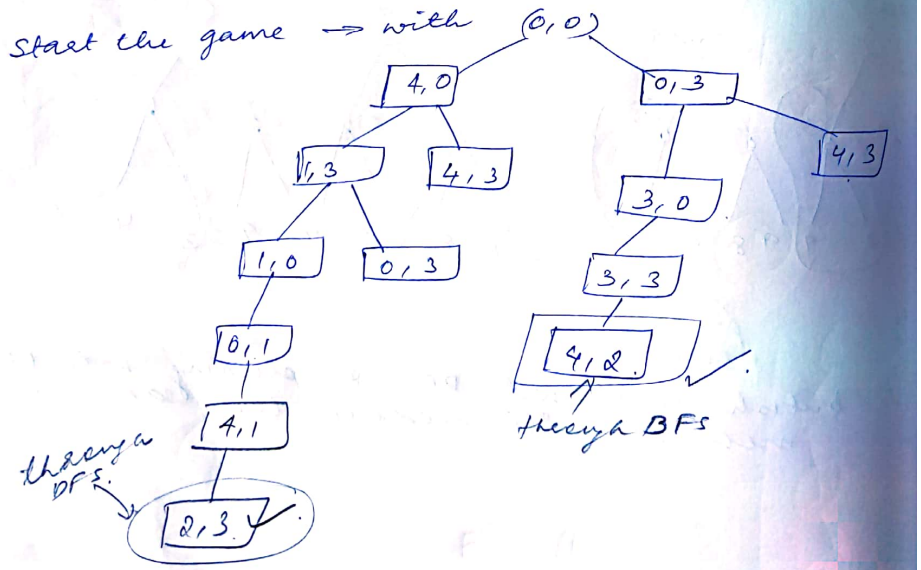
- There is a need for rules to play a certain ga
  or reach a goal.
- Also the rules should be generic enough to app
  them.

JUG PROBLEM:

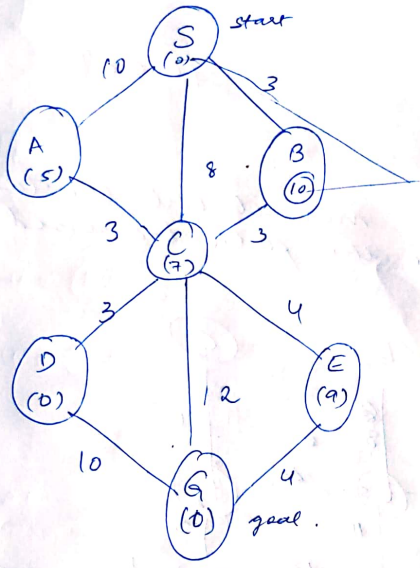


*RULES:-

			wherein
①	X, y	$X \leq 4$	
②	X, y	$Y \leq 3$	
③	X, y	$X > 0$, $(X-D)$, Y	[when $x > 0$, D can be thrown to ground or to Y
④	X, y	$Y > 0$, X, $Y-D$	



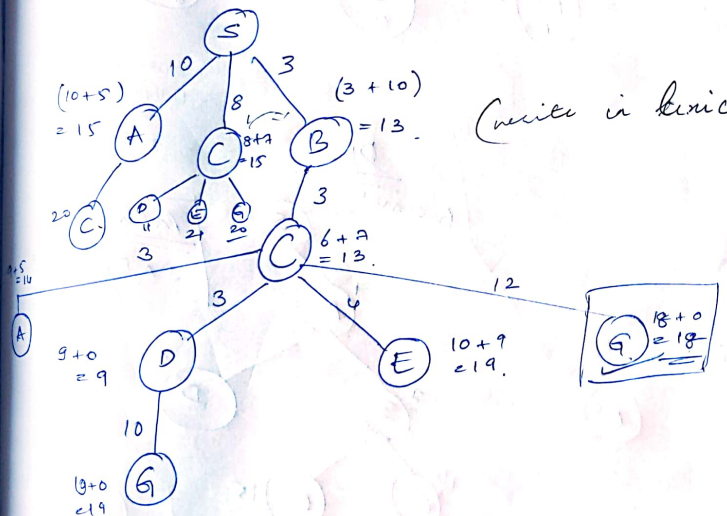
8.



A* search.
to search shortest path from S to G.

Heuristic value.

$F(x) = G(x) + H(x)$



(write in lexical order)

[ ]

[A B C]

[S]

[A B C] → sort

[S]

[B A C]

[S]

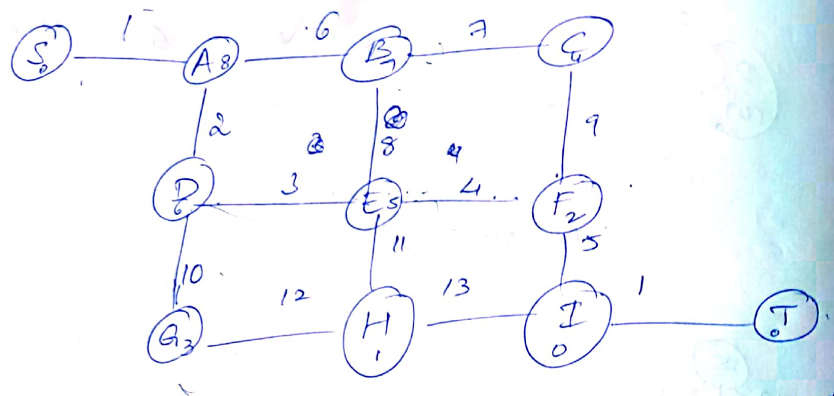
[B C A C]

[SB]
[CAC]

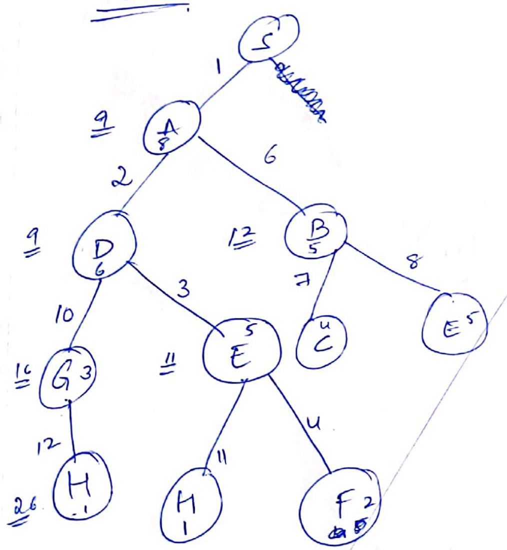
[ACC]

15/2/2020

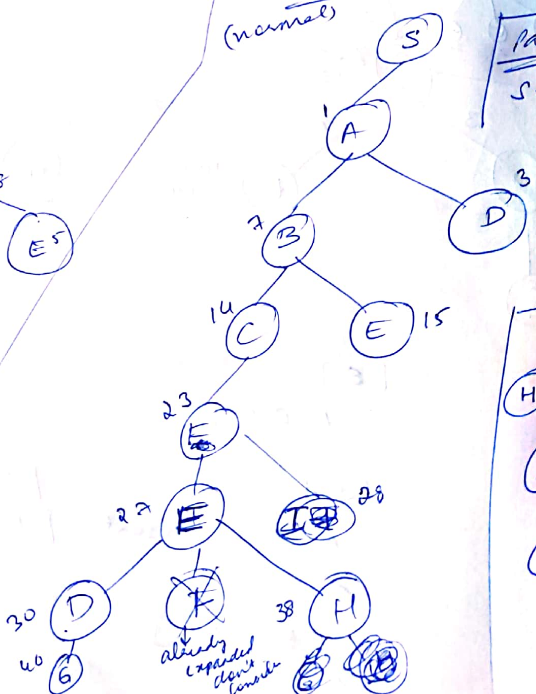
DFS → stop the min it hits the target
→ & write the path traversed.



BFS

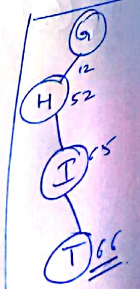


DFS
(normal)

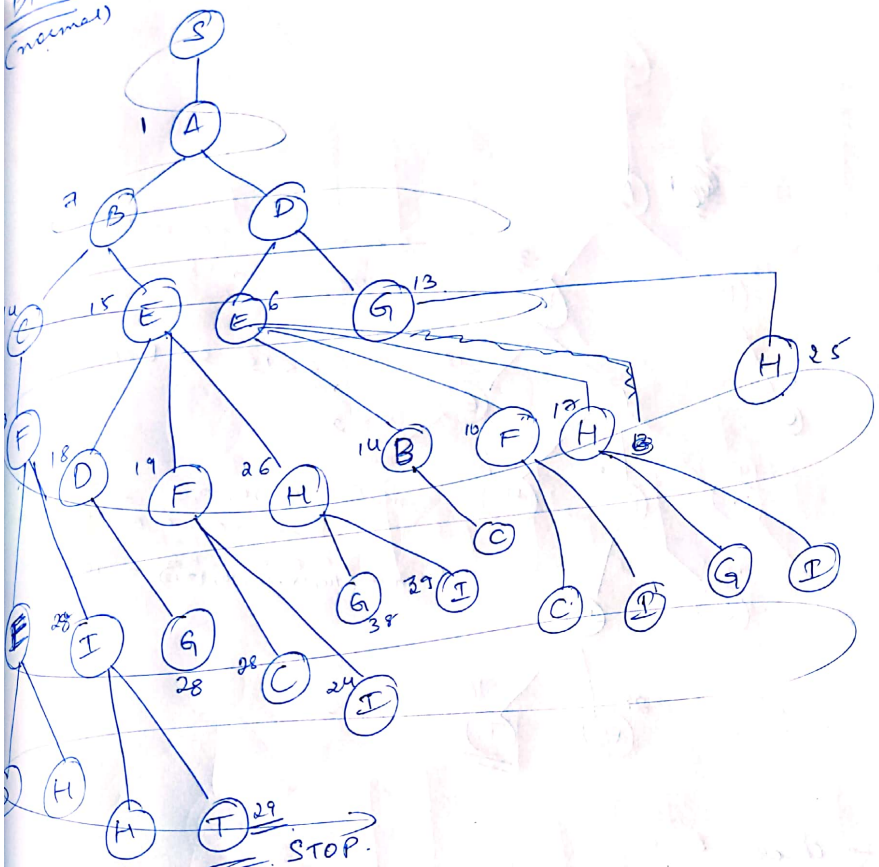


- Dashes & Formulas
- Hill climbing
  Block picking
- Jig puzzle

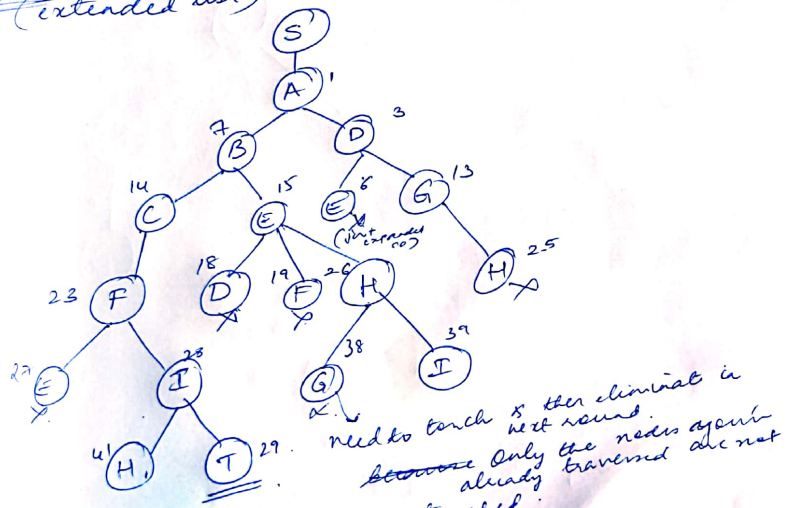
Path :
SABCFEDG



BFS
(normal)



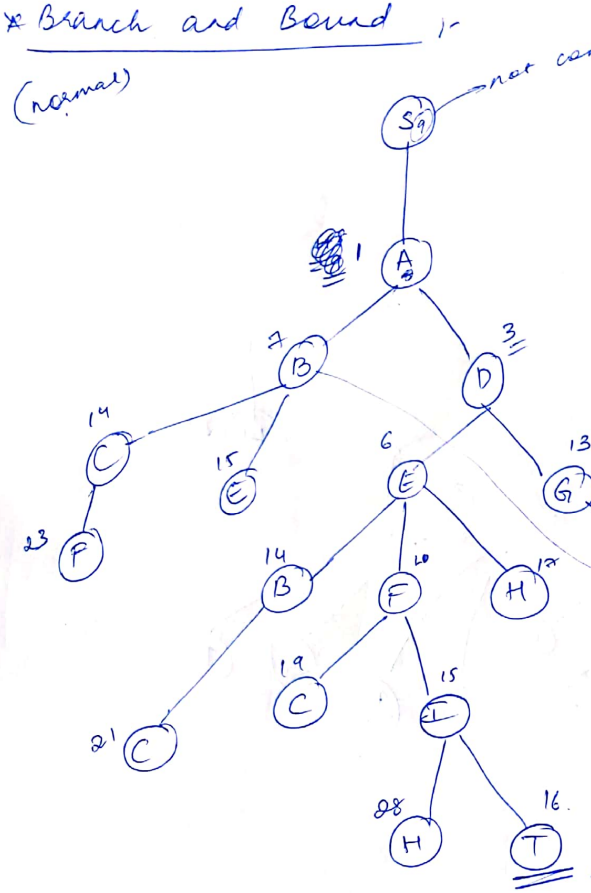
BFS (extended list)



need to touch & then eliminate in next round.
~~because~~ Only the nodes again already traversed are not touched.

* Branch and Bound :-

(normal)



3 < 7 so go with expansion of D.

6 < 13 so Expand 'E'

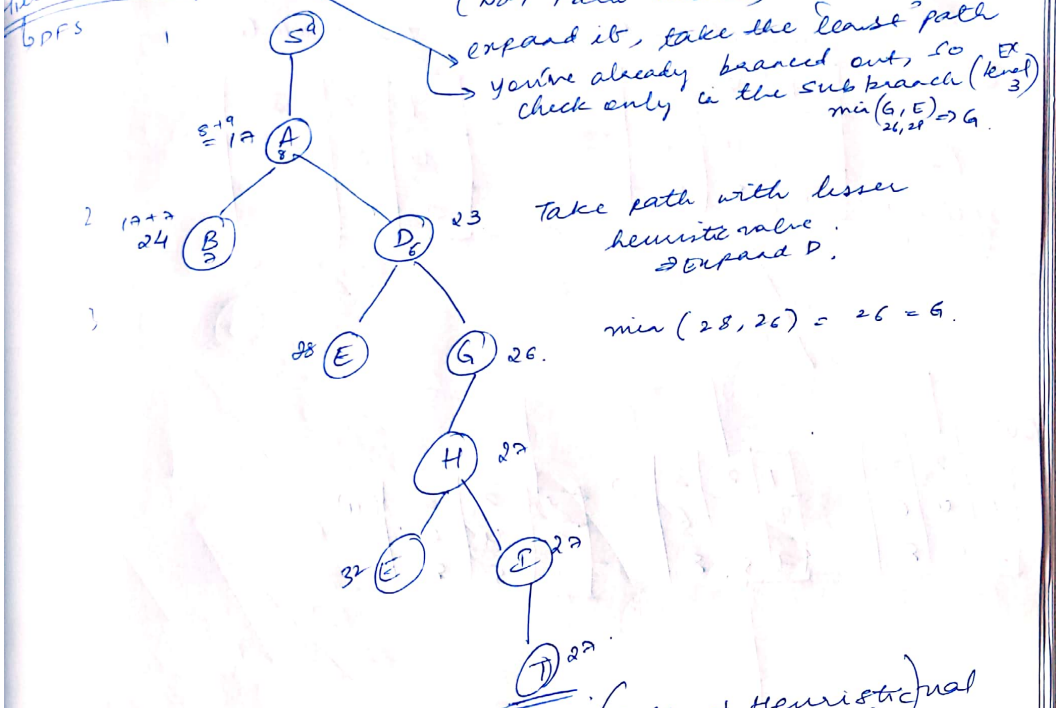
min(7, 14, 10, 17, 13) = 7
so expand 'B' (left now)

min (C E B F H G
14, 15, 14, 10, 17, 13) = 10 ⇒ F

expanded → Check

Hill climbing :- → consider ONLY Heuristic values. (NOT Path values)

DFS



→ expand it, take the "least" path

→ you've already branced out, so check only in the sub branch (level 3)

min (G, E) (26, 28) ⇒ G.

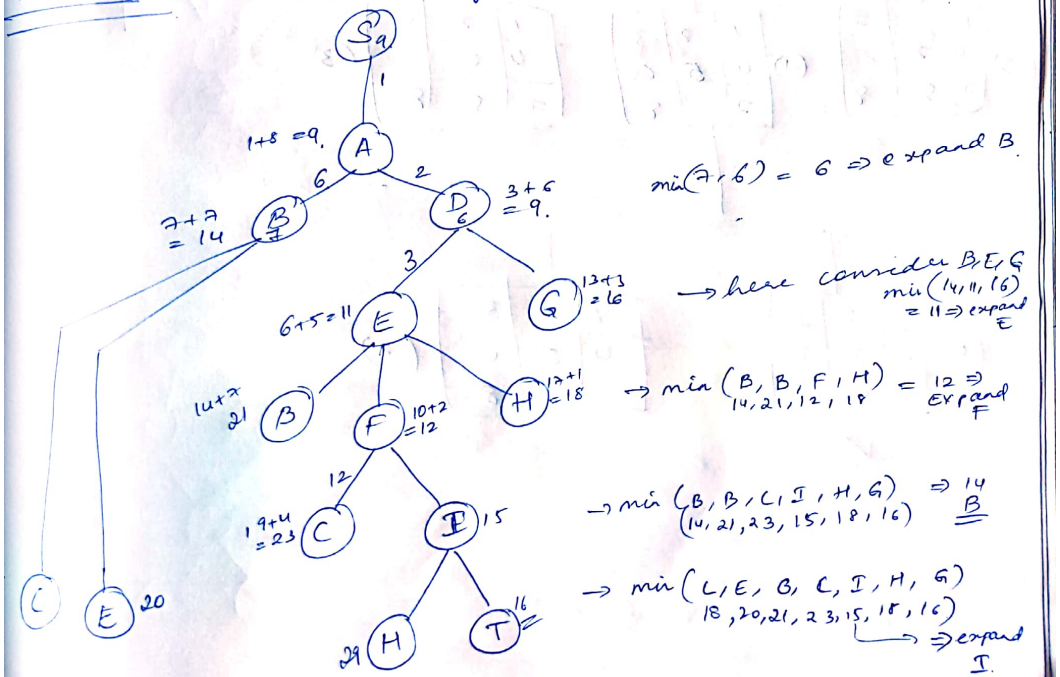
Take path with lesser heuristic value.
⇒ Expand D.

min (28, 26) = 26 = G.

BOTH (path + Heuristic) cumulat heuristic val.

A *

→ consider
→ add only



min (7, 6) = 6 ⇒ expand B.

→ here consider B, E, G
min (14, 11, 16)
= 11 ⇒ expand E

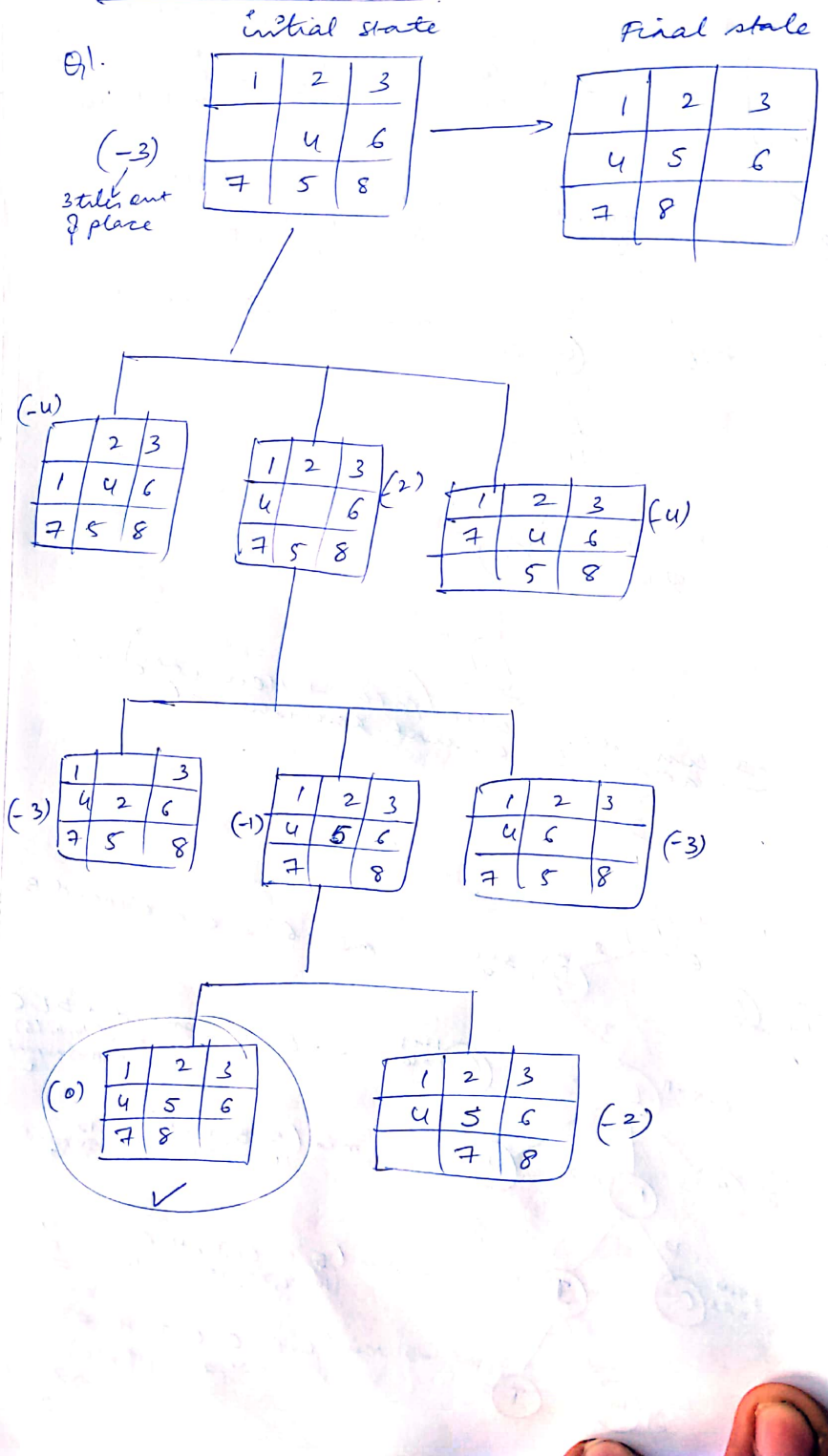
→ min (B, B, F, H) = 12 ⇒ expand F
(14, 21, 12, 18)

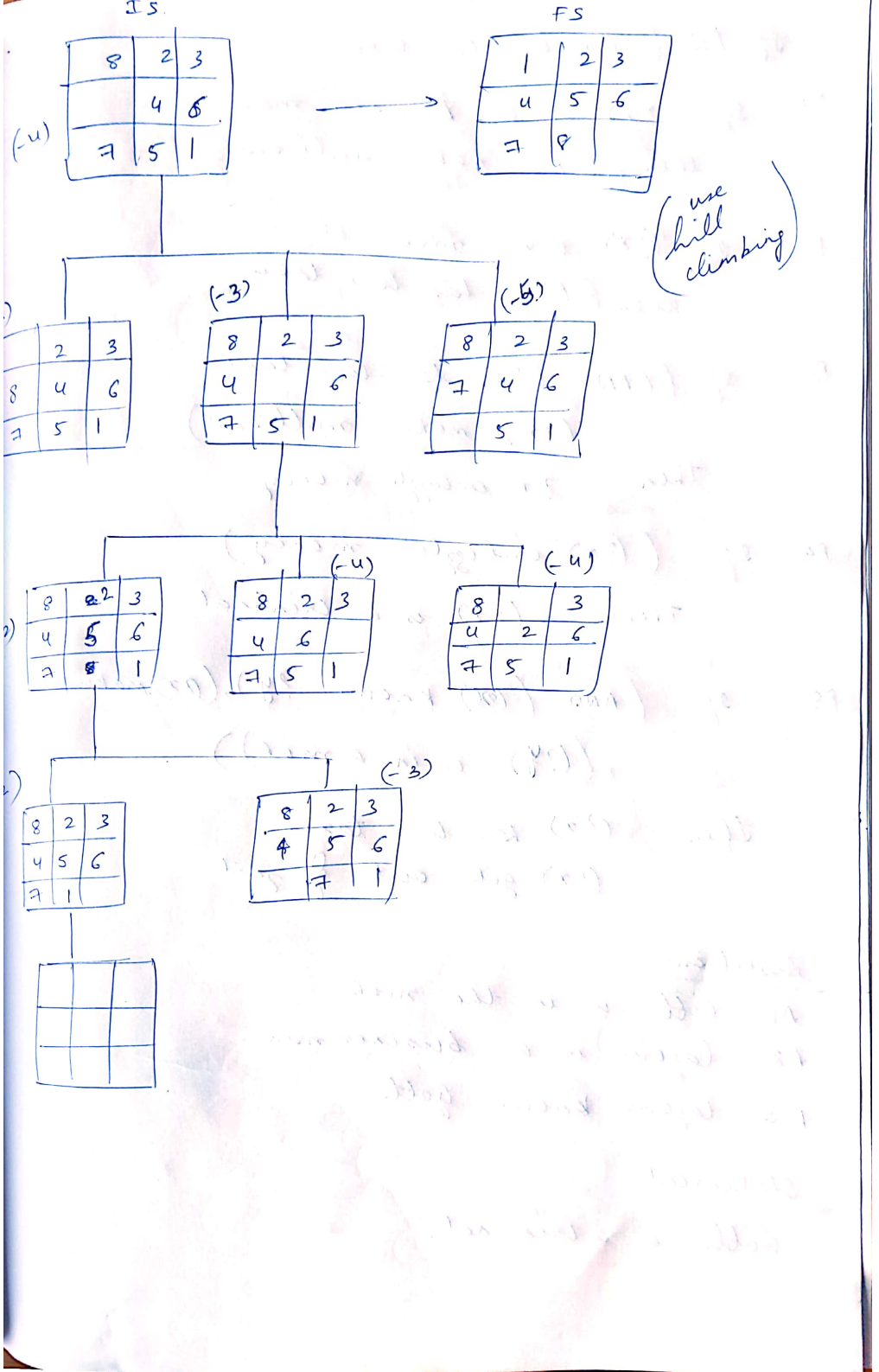
→ min (B, B, C, I, H, G) ⇒ 14 B
(14, 21, 23, 15, 18, 16)

→ min (C, E, B, C, I, H, G)
(18, 20, 21, 23, 15, 18, 16)
→ ⇒ expand I.

TILE PROBLEM

Q1.





Q. ~~If AND (?x) has low taxes~~.

P1: If (?x) is a business man
then (?x) makes millions

P2: If (?x) is in a mob
then (?x) has low taxes

P3: If ((AND (?x) has low taxes
(?x) makes millions))
Then ?x embazzle money

P4: If ((?x) embezzles money)
Then (?x) is a criminal

P5: If (AND ((?x) knows ?y), ((?x) makes mil
, ((?y) is in a mob))
Then (?x) has low taxes
(?x) gets out of jail

Assertions:

A1: Gotti is in the mob
A2: Cagoni is a businessman.
A3: Cagoni knows Gotti

Statement.

Gotti is a criminal.

Backwa

Gotti

Gotti is a

→ so a

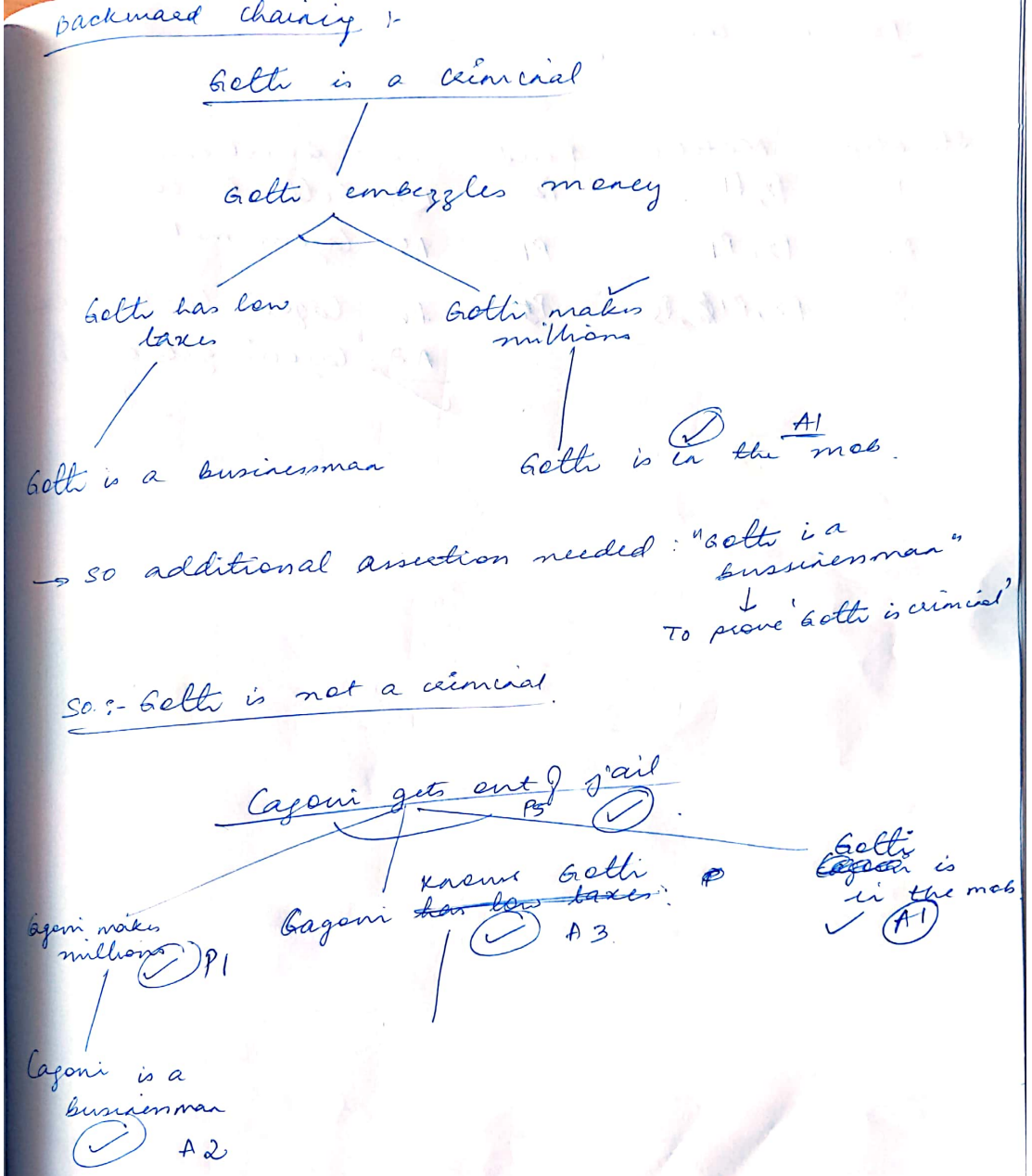
So :- Go

Cagoni makes millions

Cagoni is busine

Hence

Backward chaining :-



Forward chaining

Iteration	Matches	Fired	New Assertions
1.	P2, P1	P2	A4: Goth has low taxes
2.	P2, P1	P1	A5: Cagoni makes million
3.	P2, P1, P5	P5	A6: Cagoni has low taxes.
			A7: Cagoni gets out of jail.