18.07.2023 ~~Monday~~ Tuesday

Lecture-1

Lecture 1

Date: Page:

Data + Info = Knowledge

Computer	Human Brain
① Efficiency ↓ on knowledge ↑	① opp.
② Storage & Retrieval remains same	② Forget old but new one remain same
③ Patience ↳ same at each time	③ No Patience
④ Experience Newer one ⇒ good	④ opp. Pride issue
⑤ Emotions Test lie ✗	⑤ ✓
⑥ capacity of learning (Can be done by programming)	⑥ खुद से पढ़ सकता
⑦ नहीं।	अपग्रेड कर सकता है

20.07.2023
Thursday

Lecture. 2

Date: Page:

APPLICATIONS OF AI

1. Game Playing — Tic Tac Toe
2. Weather forecasting (Prediction Model)
3. N.L.P.
4. Neural Networks.
5. Learning
6. Fuzzy Logic (Not clear) — Small camel → 15, 20, 25

[0, 1]

Fuzzy Set of Small = {15, 10, 25, 30}
= {10 - 30}

7. Planning
8. Robotics

State Space Representation — Problem Characteristics

for every problem

How in a world we will show it using State

State Space Representation

1. Problem Definition
2. Initial state of a problem (start state)
3. Goal / final / finish state of a problem.

finite है तो सारे बनाओ। IS O

Or we can for infinite just represent by ...
(2 to 3 states) GS O

Date: Page:

④ (optional) Rules / operators / Instruction / Restrictions / constraints required to solve the problem.

⑤ One possible solⁿ from Initial State (IS) to Goal State (GS)

Farmer, Fox, Goose & Gram
(F_1, F_2, G_1, G_2)

① अकेला छोड़ा तो Fox Goose को खा जायेगा — Goose Gram को खा जायेगा।

wants to take all belonging from one side of river, he needs to take it opp side. At a time only 2 can go in boat & one is farmer for sure.

② LHS: F_1 F_2 G_1 G_2 B | RHS: | IS → River

can draw pictorially

③ | F_1 F_2 B G_1 G_2 | GS

④ | F_1 F_2 G_1 G_2 B | IS

→ F_1 G_1 B

| F_2 G_2 | F_1 G_1 B | GS

← B F_1

Fo : Fox
Gr : Grain
Fa : Farmer
Go : Goat

$F_1F_2G_2B$	G_1

F_1F_2B ↙ ↘ F_1G_2B

Fa Fo Go Gr	

↓

Fo, Gr	Fa, Go

↓

Fa, Fo, Gr	Go

↓

Fo	Fa, Gr, Go

↓

Fo Go Fa	Gr

↓

Go	Fa Gr Fo

↓

Fa Go	Gr Fo

↓

	Fa Gr Go Fo

F_2	$F_1G_1G_2B$

↓ F_1G_1B ←

$F_1F_2G_1B$	G_2

↓ F_1F_2B →

G_1	$F_1F_2G_2B$

↓ F_1B ←

F_1G_1B	F_2G_2

↓ F_1G_1B →

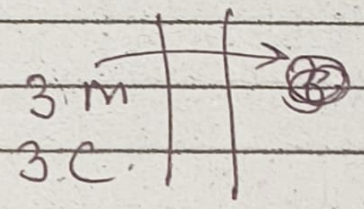
	$F_1F_2G_1G_2B$

min. no. of boat movements required = 7
(this are only)

21/09/2023 12:20

Missionaries & Cannibals

↳ Priests & Nun ↳ man eaters



At a time only 2
But one should

Cannibals should not outnumber missionaries

① 3 M 3 C on LHS Get them to RHS
Boat is tiny need to get 3M & 3C from LHS to RHS C should not outnumber M

② IS

LHS	RHS
$M_1 M_2 M_3$ $C_1 C_2 C_3$	

③ GS

LHS	RHS
	$M_1 M_2 M_3 C_1 C_2 C_3$

④ Boat is very tiny (only 2)
C should not outnumber M

All possible moves

(i) $M_1 M_2 \rightleftarrows M_1 M_2$
(ii) $M_1 \rightleftarrows M_1$
(iii) $C_1 \rightleftarrows C_1$
(iv) $M_1 C_1 \rightleftarrows M_1 C_1$
(v) $C_1 C_2 \rightleftarrows C_1 C_2$

✗

$\boxed{M_1 M_2 M_3 C_1 C_2 C_3 \mid \quad}$

$\downarrow \rightarrow M_1 C_1$

$\boxed{M_2 M_3 C_2 C_3 \mid M_1 C_1}$

$\downarrow \leftarrow M_1$

$\boxed{\quad \mid M}$

$\boxed{M_1 M_2 M_3 C_1 C_2 C_3 \mid \quad}$

$M_1 C_1 \rightarrow$

$\boxed{M_2 M_3 C_2 C_3 \mid M_1 C_1}$

$\downarrow M_1 \leftarrow$

$\boxed{M_1 M_2 M_3 C_2 C_3 \mid C_1}$

$\downarrow M_1 C_2 \rightarrow$

$\boxed{M_2 M_3 C_3 \mid M_1 C_2 C_1}$

$\downarrow C_2 \leftarrow$

$\boxed{M_2 M_3 C_2 C_3 \mid M_1 C_1}$

$\downarrow M_2 C_2 \rightarrow$

$\boxed{M_3 C_3 \mid M_1 M_2 C_2 C_1}$

$\downarrow M_3 C_2 \leftarrow$

$\boxed{M_3 C_2 C_3 \mid M_1 M_2}$

$C_1 C_2 \rightarrow$

$\boxed{M_1 M_2 M_3 C_3 \mid C_1 C_2}$

$M_1 M_2 \rightarrow$

$\boxed{M_3 C_1 C_2 C_3 \mid M_1 M_2}$

अब M पर उगा गेस
तो बढ़िया।

Minimum 11 boat movements are required.

$M_1 M_2 M_3 C_1 C_2 C_3 B$	

$C_1 C_2 B \rightarrow$

$M_1 M_2 M_3 C_3$	$C_1 C_2 B$

$C_1 B \leftarrow$

$M_1 M_2 M_3 C_1 C_3 B$	C_2

$C_1 C_3 B \rightarrow$

$M_1 M_2 M_3$	$C_1 C_2 C_3 B$

$C_1 B \leftarrow$

$M_1 M_2 M_3 C_1 B$	$C_2 C_3$

$M_1 M_2 B \rightarrow$

$M_3 C_1$	$M_1 M_2 C_2 C_3 B$

$M_1 C_2 B \leftarrow$

$M_1 C_2 M_3 B$	$M_2 C_3$

$M_1 M_3 B \rightarrow$

C_3	$M_1 M_2 M_3 C_3 C_1$

$C_3 B \leftarrow$

$C_1 C_2 B \rightarrow$

$C_1 B \leftarrow$

$C_1 C_3 B \leftarrow$

GG

$M_1 M_2 M_3 C_1 C_2 C_3$	

Date: Page:

$M_2 M_3 C_2 C_3$	$M_1 C_1$
$M_1 M_2 M_3 C_2 C_3$	C_1
$M_1 M_2 M_3$	$C_1 C_2 C_3$
$M_1 M_2 M_3 C_1$	$C_2 C_3$
$M_1 C_1$	$M_2 M_3 C_2 C_3$
$M_1 M_2 C_1 C_2$	$M_3 C_3$
$C_1 C_2$	$M_1 M_2 M_3 C_3$
$C_1 C_2 C_3$	$M_1 M_2 M_3$
C_1	$M_1 M_2 M_3 C_2 C_3$
$C_1 C_2$	$M_1 M_2 M_3 C_3$
	$M_1 M_2 M_3 C_1 C_2 C_3$

Tic Toc Puzzle

Date: Page:

8 Puzzle Problem

15 Puzzle.

Take any one I.S. & any F.S.

	3	7
1	4	5
2	5	6

1	4	6
2		7
3	5	8

A tile can only be moved to blank tile.

	3	7
1	4	5
2	8	6

Place the no. from 1 to 9 ~~are~~ into 3×3 matrix such that the sum of elements across every row, column & diagonals are equal.

up

left

1	3	7
	4	5
2	8	6

3		7
1	4	5
2	8	6

down

left

1	4	6
2		7
3	5	8

GS

Tic Toc Puzzle

Date: Page:

8 Puzzle Problem

15 Puzzle.

Take any one I.S. & any F.S.

	3	7
1	4	5
2	8	6

1	4	6
2		7
3	5	8

A tile can only be moved to blank tile.

	3	7
1	4	5
2	8	6

Place the no. from 1 to 9 ~~in~~ into 3×3 matrix such that the sum of elements across every row, column & diagonals are equal.

up

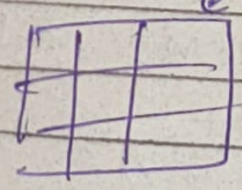
left

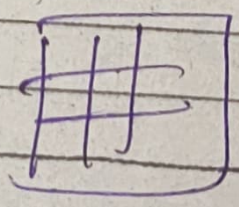
1	3	7
	4	5
2	8	6

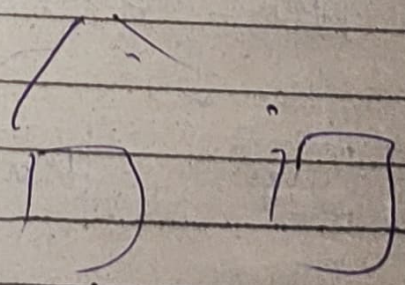
3		7
1	4	5
2	8	6

down

left







1	4	6
2		7
3	5	8

GS

intelli ~~elipse~~ VSC

code gu C++

Lecture-3

] understand ~~envir.~~ settings

24.07.2023
Monday

Date: Page:

Games maths game

AI agents
AI programme.

~~AI agents.~~

studio

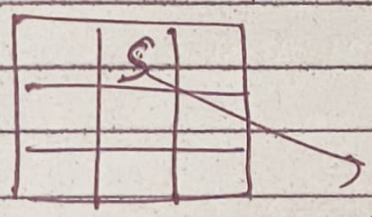
run a program

"architecture same"

Magic Square :-

n×n matrix → n → even (own)
→ n → odd

N=3 3×3 matrix.



1 to 9

all rows & columns & diagonals sum == equal

Start from here (start with middle element of 1st row

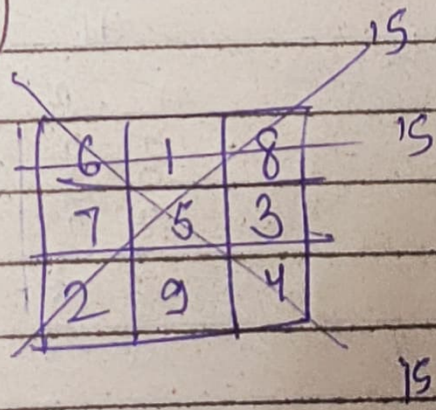
go up & right.

put in vertical down & vertical right last

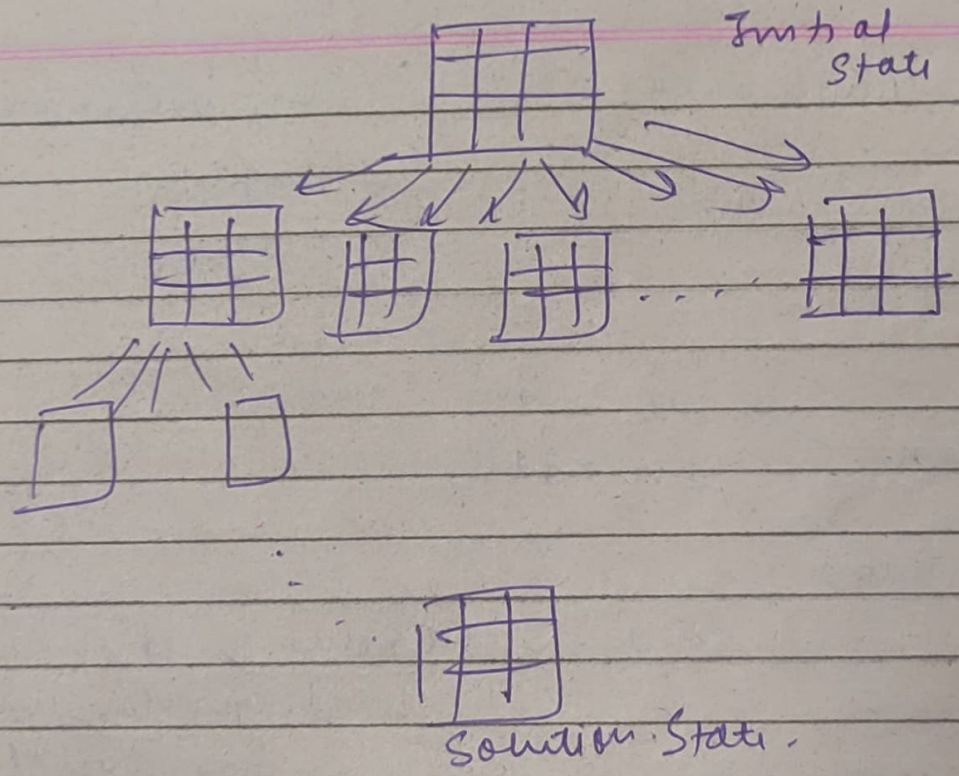
if fails (field or outside)

then put below the current (always empty)

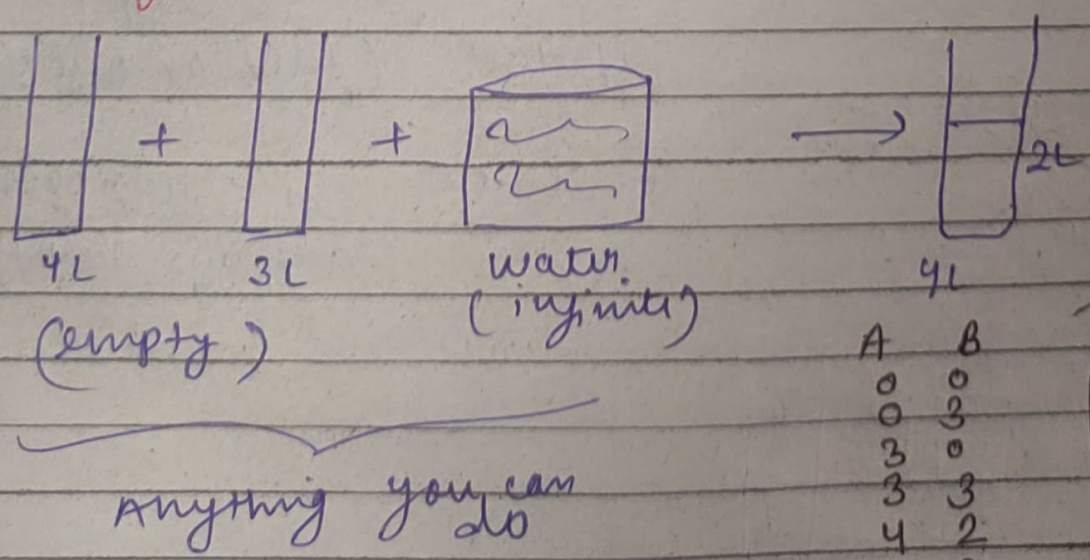
Start from first row middle element



left ←
↑ up



Water Jug Problem



A	B
0	0
0	3
3	0
3	3
4	2
0	2
2	0

Human

$y-4-x$
$y-4+x$

Can't throw milk

Milkman problem

① (Jug A, Jug B) ≡ (x, y) → and y L of water in Jug A

where $x \in 0 \to 4$

$y \in 0 \to 3$

Problem Definition

First jug should be half filled

Goal State : $(2, n)$

$n \in 0 \to 3$

$x-(3-y)$

② Initial State $(0,0)$

③ Goal State : $(2, n)$ $n \in 0 \to 3$

④

Rule No.	Current State	Preconditions	Next State	Description
1.	(x, y)	$y<3$	$(x, 3)$	Fill 3L jug
2.	(x, y)	$x<4$	$(4, y)$	Fill 4L jug
3.	(x, y)	$y>0$	$(x, 0)$	Empty 3L jug on ground
4.	(x, y)	$x>0$	$(0, y)$	Empty 4L jug on ground
5.	(x, y)	$y>0, x<4$	$(4, y-(4-x))$	Pour water from 3 to 4 until 4 is filled
6.	(x, y)	$x>0, y<3$	$(x-(3-y), 3)$	4 to 3 — 3 is filled
7.	(x, y)	$x<4, y>0$ $x+y \leq 4$	$(x+y, 0)$	Pour all water from 3 to 4
8.	(x, y)	$y<3, x>0$ $x+y \leq 3$	$(0, x+y)$	4 to 3
9.	$(0, 2)$		$(2, 0)$	If 3 has 2L then empty & pour 2 (in

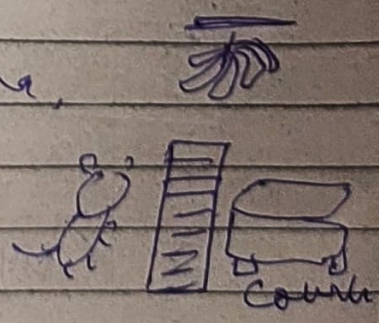
	x	y	Rule
I.S	0	0	1
	0	3	7
	3	0	1
	3	3	5
	4	2	4
	0	2	9
G.S.	2	0	GS

Q) (8, 6)

	x	y	Rule
I.S	0	0	1
	0	6	7
	6	0	1
	6	6	5
	8	4	4
	0	4	9
G.S.	4	0	GS

Monkey Banana Problem

Climb, Move the couch under Banana,
Move the stairs under the Banana



Date: Page:

(1) Measure 6L water from 4L & 9L bucket

(2) You have 3 jugs they can hold 12L, 8L & 5L of milk. 12L jug is full of milk Now you have to split ~~it using~~ in half only by using jugs in hand

(3) Make 1L from 5L & 7L → (1, n) ⇒ GS
→ (n, 1)

(4) We have 3 jugs of capacity 3L, 5L & 8L. Initially 8L jug is full of water & other 2 jugs are empty The goal is to fill 4L of H_2O in 8L jug. (No extra reservoir)

(5) 8L & 6L jug capacity. How can you get 4L of water in 8L jug.

CONSTRAIN SATISFACTION

Farmer, Fox Goose Problem

↳ Rules

Cryptarithmetic → used by military men

↳ crypt the message using arithmetic

+
−
*
/

① No two different alphabet to same digit

② 0 to 9 (10 digits available)

C_4 C_3 C_2 C_1

```
  S E N D
+ M O R E
---------
M O N E Y
```

→ $M = C_4$

$C_4 = 0$ → $M = 0$ ✗ First digit $\neq 0$ in a number आगे दिक्कत देगा

$\boxed{C_4 = 1}$ → $\boxed{M = 1}$

$C_4 == 1$ ∴ 0 should be 2 digit ∴ $O \geq 10$

$S + M + C_3 = O$

$O = 10$ $\boxed{O = 0}$

$S + M + C_3 = 10$

$\boxed{C_3 = 0}$

$S + M = 10$

$\Rightarrow S + 1 = 10$

$\Rightarrow \boxed{S = 9}$

$C_2 = 0$

$E + O + C_2 = N$ $\{0, 1, \ldots 9\} \rightarrow$ except taken

$\Rightarrow E + O + C_2 = N$

$\Rightarrow E + C_2 = N$

1
0
9

$C_2 = 0$ $\quad$ $\boxed{C_2 = 1}$

$E = N$ ✗

$E + 1 = N$

$N = 2, E = 1$ ✗

$\boxed{N = 3 \mid E = 2}$

$C_2 = 1$

$E + 1 =$	N
2	3
3	4
4	5
5	6
7	8
8	9

$D + E + C_1 = Y$

$D + 2 + C_1 = Y$

$Y \geq 12$ " $\quad C_1 = 0$ $\quad$ $C_1 = 1$

$D + 2 + 0 = Y$

$D + 2 = Y$

$D + E = Y$

$D + E = Y$

$E = 5$

$N = 6$

$D = 7$

$Y = 2$

$R = 8$

$C_5\ C_4\ C_3\ C_2\ C_1$

ROADS

\+ CROSS

DANGER

$C_5 = 0$ → $D = 0$ ✗

$C_5 = 1$ → $\boxed{D = 1}$

$C_4 = 0$: $R + C = A$

$\boxed{10} = A$

12

13

14

15

$C_4 = 1$: $1 + R + C = A$

$\boxed{R\ 2} + \boxed{C\ 8}$

$C_3 + O + R = N$

$C_3 = 0$: $O + R = N$

$O + 2 = N$

$N - O = 2$

2 0 ✗

1 ✗

2 ✗

$C_3 = 1$

$\boxed{5\ N}\ \boxed{3\ 0}$

$A + O + C_2 = G$

C_2: $A + O = G$

$C_2 = 1$

03.08.2023
Thursday

Date: Page:

KNOWLEDGE REPRESENTATION

Predicate logic — Propositional logic

Backward Chaining
Resolution

1
2
3
prog:

Predicate logic Resolution

① ES → PL (English with statements → Predicate logic)
② PL → Clause Form

$\vee$, $\neg$ } allowed
$\neg$ → unbound brackets not allowed
↳ bound on only

$\neg(\neg a) = a$

$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$ — a → Literal
↳ use this to divide in 2 part

$\neg(a \vee b) = \neg a \wedge \neg b$

$a \to b \approx \neg a \vee b$

$a \leftrightarrow b = (a \to b) \wedge (b \to a)$

i) $(a \vee b)$
ii) $(a \vee c)$

Quantifiers
- Universal $\forall$ (easily drop)
- Existential $\exists$ (use skolemization) → $\forall$ convert into Universal → drop this

$\forall x \; \exists y$ — making the scope of x to y

Prenex Normal form → drop prefix

① $a \vee b$
② $\neg a \vee c$
③ $\neg c \vee \neg b$
④ $d \vee a$
⑤ $\neg a$

$d = ?$

get its negate

unification algo

$\neg d$ $\quad$ $d \vee a$ (④)

a $\quad$ $\neg a \vee c$ (②)

c $\quad$ $\neg c \vee \neg b$

$\neg b$ $\quad$ $a \vee b$ (①)

a $\quad$ $\neg a$ (⑤)

$\therefore$

Ex:
Marcus was a man

Find generalise & specific

man $\quad$ marcus

Generalise (specific) ☆

man(marcus)

predicate symbol $\quad$ arguments

born_in(x, y) = x was born in y.

① ②

Marcus was born in 40 AD, $\quad$ born_in(marcus, 40 AD)

Arguments should be same in order to cut.

Q)
1. Fido is a dog.
2. All dogs are animals
3. All animals will die.

Prove by resolution "Fido will die"

(Semi English sentence) PL → CF Rename a variable such that each quantifier is defining one quantifier linked with unique variable

SES →

PL (Predicate Logic)

1. dog(Fido)
2. For all x if x is dog then x is an animal

$\forall x : dog(x) \longrightarrow animal(x)$

3. $\forall x : animal(x) \longrightarrow die(x)$

PL → CF (Clause form).

1. dog(Fido)
2. $\forall x : \neg dog(x) \vee animal(x)$ ($p \to q$ = $\neg p \vee q$)

Prenex Normal Form

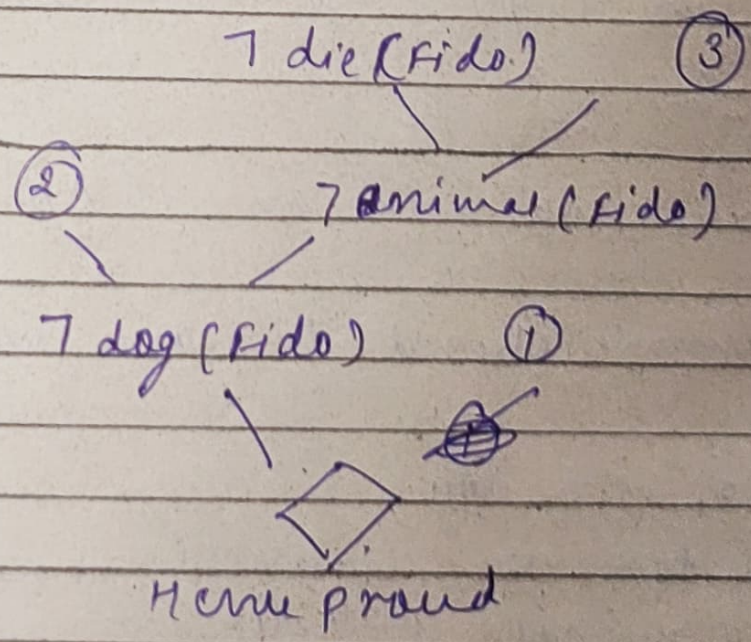
(∴ drop $\forall x$) $\neg dog(x) \vee animal(x)$

3. $\neg animal(x) \vee die(x)$. (Renaming to avoid confusion)

To Prove

die(Fido)

Take contradiction



Q.2) → Q rewrite files

Date: Page:

1. ~~Animal~~ Anyone passing his history exam & winning the lottery is happy

2. Anyone who studies or is lucky can pass all his exams. John did not study but he ~~will~~ is lucky. Anyone who is lucky wins the lottery.
"Is John happy" - Prove

Sol: PL

1. $\forall x : pass(x, history) \wedge wins(x, lottery) \rightarrow happy(x)$

2. $\forall x : studies(x) \vee lucky(x) \rightarrow \cancel{pass(x,}$ $\forall y : pass(x, y)$

3. $\neg Study(John) \wedge Lucky(John)$.

4. $\forall x : lucky(x) \rightarrow wins(x, lottery)$

CF

1. $\neg pass(x, history) \vee \cancel{lucky}\ \neg wins(x, lott) \vee happy(x)$

2. $\forall x, \forall y : \neg [Study(x) \vee Lucky(x)] \vee pass(x, y)$
$\Rightarrow [\neg Study(x) \wedge \neg lucky(x)] \vee pass(x, y)$

$\Rightarrow [\neg Study(x) \vee pass(x, y)] \wedge [\neg lucky(x) \vee pass(x, y)]$

2.1) $\neg Study(x) \vee pass(x, y)$
2.2) $\neg lucky(x) \vee pass(x, y)$

soln के substitute

Date: Page:

3.1) ¬ study (John)

3.2) lucky (John)

4. ¬ lucky (x_2) ∨ wins (x_2, lott)

To prove: happy (John)

¬ happy (John) (1)

x / John

(2.2) ¬ pass (John, history) ∨ ¬ wins (John, lottery)

x / John

(3.2) ¬ wins (John, lott) ∨ ¬ lucky (John)

¬ wins (John, lottery) (4)

x_2 / John

(3.2) ¬ lucky (John)

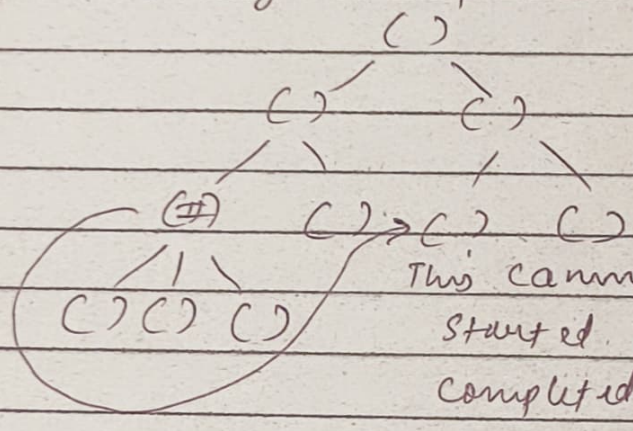
□

31.07.2023
Monday

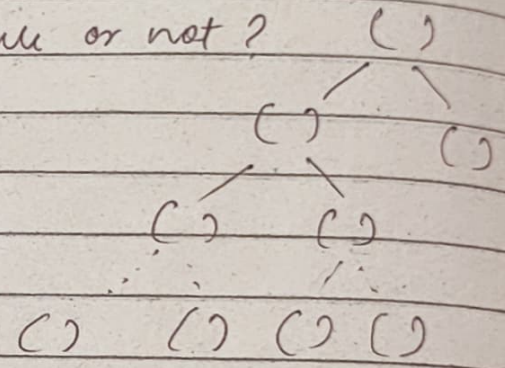
Give Reason too (one liner)

7 ~~Stage~~ Problems Characteristics

① Whether the problem is decomposable or not?
Partially decomposable problem.



This cannot be started until # is completed.



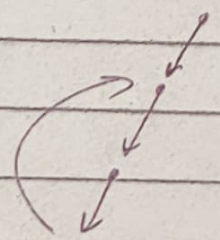
$$\int (f_1(x) + f_2(x) + \cdots f_n(x))dx$$

$$\int f_1(x).dx \quad \int f_2(x).dx \quad \cdots \quad \int f_k(x)\,dx$$

Chess can't be decomposed.

* Water Jug can be decomposable as well as Non decomposable (Justify with reason)

② Whether problem is recoverable or not?



8 Puzzle → R
Water - Jug → R
Chess → NR
Milk Jug → R but with modification

dr cp s sc K
↳ at supreme court K

Color cubic Sq. game

Date: Page:

③ Whether the outcome of the problem is certain / not?
do you the outcome?

Water Jug → Certain
8 Puzzle → Certain
Chess → Not predictable
Cricket → Not Predicable

Watching and game can predict

] Can't say till last who will win.

④ Whether the ~~solution to the~~ problem is any path / best path?
TSP

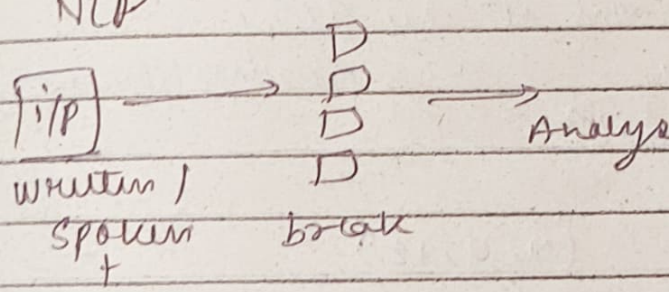
absolute / relative

Water Jug
Cubic square

TSP (Traveling ~~Salesman Prob.~~)

⑤ Whether the solution to the problem is a state or a path?
solution path है।

NLP



Cryptarithmetic
Water Jug

⑥ Whether the problem is solitary / commensational?

अकेला
खेल solution देगा।

बार बार problem पूछता है ?

2 P Games

depends on program.

(7) What amount of knowledge is required to solve the problem?

a little — a large

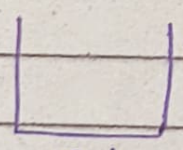
Reading a newspaper and explaining about it
↳ A large amount of language

Other all problems required little knowledge

Solve the following using state space Representn

(1) A milkman carries a full 12 L container of milk. He needs to deliver 6 L of milk but the customer only has 8L & 5 L jugs (i) how can it be done
(ii) write → production rules
(iii) Give the 2 possible solⁿ.

Sol:



12 L

(0,0)

(0,5)

(0,5,7)

(0,0,12)

(8,0,4)

(3,5,4)

~~(3,6,3)~~

~~(6,6,0)~~

$$\begin{array}{r} 12L \\ -\ 8 \\ \hline 4 \end{array}$$

Capacity दी हो तो
throw मत करना

C 5 8

(12, 0, 0)

(8, 5, 0) → (0, 5, 7)

(8, 5, 0) → (8, 0, 5)

(12, 0, 0) → (4, 0, 8)

5 L	8 L	12 L
0	0	12
0	8	4
5	3	4
0	3	9
3	0	9
3	8	1
5	6	1
0	6	6

Q) Analyze each of the following problems on 7 problems discussing

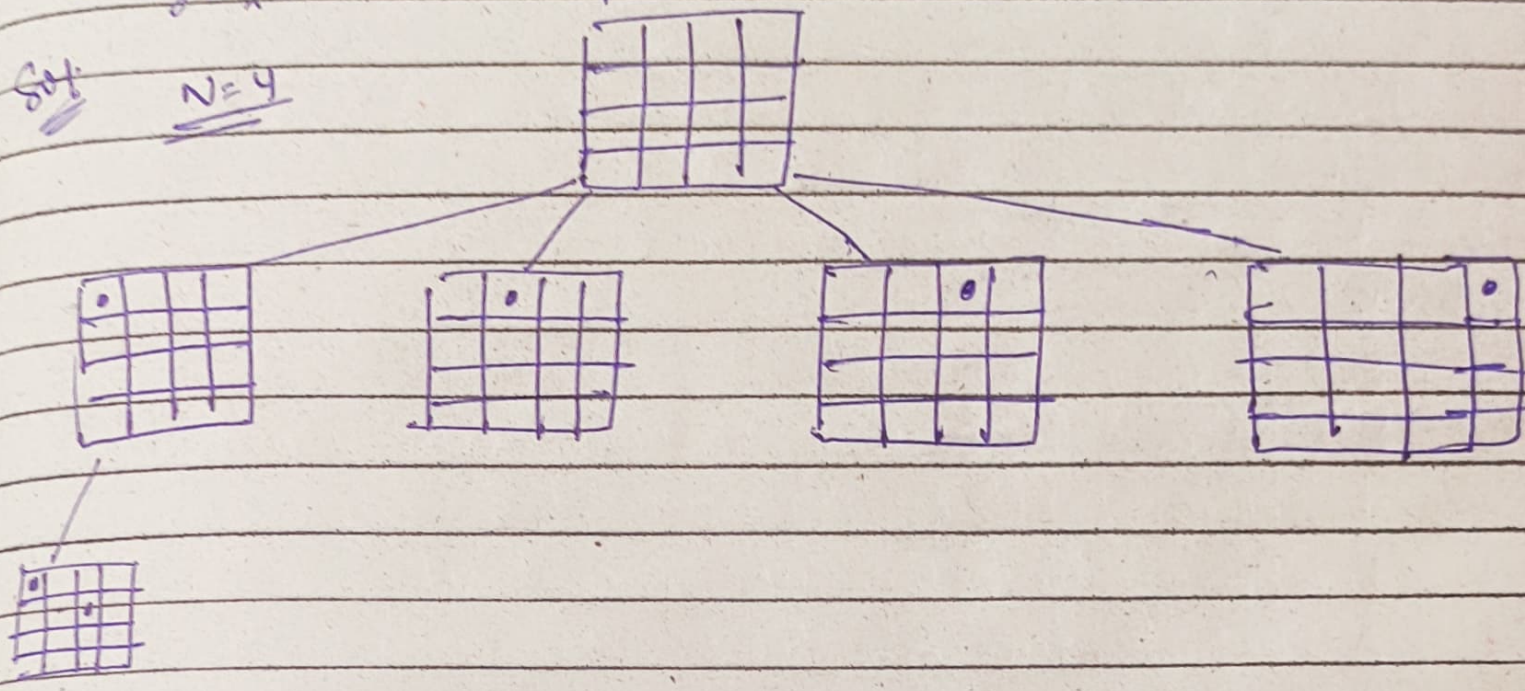
Problem	Decomp/ Not Dec	Reversable/ Not	certain/ not	any path/ best path	state/ path	Solitary/ Commutative	little/ Large
Chess	ND	NR	NP (Not Predi-ctable)	AP	S	C	L
Water-Jug	ND/D	R	C	AP	S		
8-Puzzle							
Travelling Salesman							
Missionaries & Cannibals							
Tower of Hanoi							

(Take any N)

Date: Page:

5) Write the state space representation of N-Queen. And analyze it based on 7 problem characteristics

Sol: N=4



Hypath, Solitary, Certain, Recoverable, Pat

Q) HERE = COMES − SHE

Assume S = 8 Find the value of R+H+O

$C_5\ C_4\ C_3\ C_2\ C_1$

Sol:

```
  COMES
-  SHE
-------
  HERE
```

$8 - E = E$

$\boxed{E = 4.}$

$18 - E = E$

$\cancel{E = 9}$

$\boxed{C_5 = 0}$

$C = 0$

$\Rightarrow \boxed{C = 1}$

$\boxed{C_4 = 1}$

$O = H$

$\boxed{H = 9}$

$\cancel{E = 4}$

$\cancel{4 = H}$

$E = 4$

$4 - H = R$

$E - H = R$

$14\ \cancel{4} - 9 =$

$\boxed{R = 5}$

$\boxed{C_2 = 1}$

10

$\boxed{O = 0}$

$9 + 5 + 0 = 14$

$M - 1 - S = E \Rightarrow (M - 1) - S = 4$

```
   A P R          H S P
 x O C T        x D U O
 -------        -------
   P U R A        U T P S
 R O J R x      H S P x
R E C U x x    P P A P X X
-----------    -----------
R A A J A A    P H O T O S
```

Q)
Consider the following constraint satisfaction problem where each variable is represented by single English letter has to be defined by single digit
Identify a 'K' substitution for which the CSV problem holds true for both the problems
(Contraint satisfaction Problem)

(i) CP + IS + FUN = TRUE

(ii) find the value of T + E + N?

if ONE + ONE + ONE + ONE = TEN

$C_4\ C_3\ C_2\ C_1$

(i)

```
   CP
   IS
+ FUN
-----
 TRUE
```

$C_3 = 1$

$F = 9$

$R = 0$

$C_4 = 1$

$T = 1$

$C_2 + I + C + U = U$

$C_2 + I + C = 0$

$C_1 + P + S + N = E$

Date: Page:

O	N	E
O	N	E
O	N	E
O	N	E
T	E	N

4O = T

4N = E

4E = N

27.08.2023
Monday

$p \to q$
$\neg p \vee q$

Q) Considering the following
All people who are not poor & are smart are happy.
Those people who read are not stupid.
John can read & is ~~happy~~ wealthy
Happy people having exciting life,
Does some people has exciting lifes? Proof this by resolution

Sol: (i) $\forall x : \neg poor(x) \wedge smart(x) \to happy(x)$
[x : people]
(ii) $\forall x : read(x) \to \neg stupid(x)$
(iii) $read(John) \wedge$ ~~has~~ $wealthy(John)$
(iv) $\forall x ; happy(x) \to exciting\text{-}life(x)$ → life(x, exciting)
No. of predicate symbols appearance should be > 1

Antonyms
stupid ~~X~~ small (replace) wealthy × poor

$\neg poor \approx wealthy$ $\neg stupid \approx smart$

PL
(i) $\forall x : wealthy(x) \wedge smart(x) \to happ(x)$
(ii) $\forall x : read(x) \to smart(x)$
(iii) $read(John) \wedge wealthy(John)$
(iv) $\forall x : happy(x) \to exciting\text{-}life(x)$

CF
(i) $\neg(wealthy(x) \wedge smart(x)) \vee happy(x)$
$\equiv \neg wealthy(x) \vee \neg smart(x) \vee happy(x)$
(ii) $\neg read(x) \vee smart(x)$

M.1. Take opp. & find negate

M.2. units to prove as next clause. & find 2 opp. formula.

Date: Page:

(iii) .1 read (John)

(ii) .2 ~~read~~ ~~exciting life (x)~~ wealthy (John)

(iv) ¬happy(x) ∨ exciting life (x)

To prove : exciting life (x)

Contradict

¬ exciting life(x) (iv)

¬ happ(x) (i)

¬ wealthy (x) ∨ ¬ smart (x) (iii) .2

x = John

¬ smart (John) (ii)

x = John

(iii) .1 ¬ read (John)

" "

BACKWARD CHAINING.

#Theoritical analysis नहीं करो

↳ To prove करना है उसे Right से पढ़ो।

Consider the following

(i) marcus was a man

(ii) marcus was a pompeian

(iii) All pompeians were roman

(iv) Caesar was a ruler

(v) All romans were either loyal to caesar or hated him.

(vi) Marcus try to assasinate caesar.

(vii) Everyone is loyal to someone

(viii) People try to assasinate rulers they are not loyal to

(Prove that Marcus is not loyal to caesar.) Find whether Marcus is loyal or not loyal to caesar)

FL

(i) man (marcus)

(ii) pompeian (marcus)

(iii) ∀ x : ~~romans (x)~~ ~~x : pompeian~~

pompeian (x) → Romans (x)

(iv) Ruler (Caesar) ✓ (I)

(v) ∀x : Romans (x) → loyal-to (x, caesar) ∨ hated (x, caesar)

(vi) try-to-assasinate (marcus, caesar)

(vii) ∀x, ∃y : loyal-to (x, y) ✓ (II)

(viii) ∀x, ∀y : people (x) ∧ Ruler (y) ∧ try-to-assasinate (x,y) → ¬ loyal-to (x, y) (III)

(ix) Assume man is similar to people.

man (marcus) = people (marcus)

Backward Chaining : # clause form की जरूरत नहीं।

start think from here & find RHS of each find left side satisfiability

M: Backward chaining

¬ Loyal to (M, C)

↑ (viii) x/M , y/C

people (M) ∧ ruler(C) ∧ try-to-assa(M, C)

↑ (vi)

people (M) ∧ ruler (caesar)

↑ (iv)

~~people (M)~~

↑ (ix)

man (Marcus)

(I) → अकेला नहीं है।

(II) LHS नहीं है।

(III) Right to stand

Clause Form

$p \to q$

$\neg p \vee q$

(i) man (M)

(ii) pompeian (M)

(iii) ¬ pomeian (x) ∨ Romans (x)

(iv) Ruler (caesar)

(v) ¬ Romans (x) ∨ Loyal to (x, C) ∨ hated (x, C)

(vi) try-to-assa (M, C)

(vii) Loyal to (x, f(x)) where f(x) = y "can't subst"

(viii) ¬ people ∨ ¬ Ruler (y) ∨ try to-assa (x, y)

∨ ¬ Loya-to (x, y)

Possible variation of Resolution

(Answer to the Question)

Ans

Loyal to (M, C) | ¬ Loyal to (M, C)

Start with

Loyal to (M, C) ∨ ¬ Loyal-to (M, C)

↳ एक बचेगा (Not empty this time)

vii) → Can't substitute (unnormalizing)

Loyal to (M, C) ∨ ¬ Loyal to (M, C) (viii)

x/M , y/C

¬ Loyal to (M, C) ∨ ¬ people (M) ∨ ¬ Ruler (C) ∨ ¬ try to assa (M, C) (vi)

¬ Loyal to (M, C) ∨ ¬ people ∨ ¬ Ruler (C) (i)

man ⇒ people

¬ Loyal to (M, C) ∨ ¬ Ruler (C) (iv)

$\boxed{\neg \text{ Loyal to } (M, C)}$

Ans.

Synonym & Antonym लगाने के लिए Arguments Same होने चाहिये।

Consider the following fact.

(i) Marcus was a man

(ii) Marcus was a pompeian

(iii) Marcus was born in 40 AD

(iv) All men are mortal

(v) All pompeians are die when volcano erupted in 79 AD

(vi) No mortals lives longer than 150 yrs.

(vii) It is now 1991

(viii) Alive means not dead

(ix) If someone dies then he is dead at all later times.

When did Marcus died?

$\neg$ died (M, t)
Start here

$\neg$ died (M, t) $\vee$ died (M, t)

Represent above facts using predicate logic. Find out whether Marcus is alive.

~~M : Marcus~~

Sol: (i) man (M)

(ii) pompeian (M)

(iii) born_in (M, 40 AD)

(iv) $\forall x$: ~~mortal(x)~~ man(x) $\rightarrow$ mortal(x)

(v) ~~$\forall x$: pompeian(~~

~~volcano~~ erupted (vol., 79 AD) $\wedge$ $\forall x$: pompeian (x) $\rightarrow$ died(x, 79 AD)

(vi) $\forall x \forall t_1, \forall t_2$ mortals(x) $\wedge$ born_in(x, t_1) $\wedge$ gt ($t_2 - t_1$, 150) $\rightarrow$ died (x, t_2)

(vii) now = 1991

(viii) $\forall x, \forall t_1$ alive (x, t_1) $\rightarrow$ $\neg$died (x, t_1)

(ix) $\forall x, \forall t_1 \forall t_2$ died (x, t_1) $\rightarrow$ died (x, t_2) $\wedge$ gt (t_2, t_1)

died (M, 1991) $\xrightarrow{(vii)}$ died (M, now)

(vi) $\uparrow$ x/Marcus, t_1/1991, t_2/40

mortal (M) $\wedge$ born_in (M, 40 AD) $\wedge$ gt (1991 - 40, 150)

$\uparrow$ (iii)

mortal (M) $\wedge$ gt (1991 - 40, 150)

$\uparrow$ (iv) x/~~mortal~~ M

man (M) $\wedge$ gt (1991 - 40, 150)

$\uparrow$ (i)

gt (1991 - 40, 150)

$\uparrow$

~~unknown~~ truth

Backward Chaining

work RHS for died (M) taking (vi) as start for LHS to be made

Date: Page:

Q(i) John like all kinds of food
(ii) Apples are food
(iii) Chicken is food
(iv) Anything anyone eats & isn't killed by is food.
(v) Bill eats peanuts & is still alive
(vi) Sue eats everything, Bill eats.
(a) Translate into predicate logic & formulas.
(b) Prove that John like peanuts using backward chaining
(c) Convert predicate logic → Clause form
(d) Prove that John like peanuts using resolution
(e) Use Resolution to ~~prove the~~ answer. What food does Sue eat?

Sol: (i) $\forall x$ food(x) → likes(J, x)
(ii) food(apple)
(iii) food(ch)
(iv) $\forall x \forall y$ eats(x, y) ∧ ¬killed(~~y~~ x) → food(y)
prove food

can be kept

(v) eats(Bill, peanuts) ∧ alive(Bill, ~~peanuts~~)
(vi) $\forall y$ eats(Bill, y) → eats(Sue, y)

likes(B, x) x : Peanuts

food(y)

eats(x, y) ∧ ¬ killed(x)

10.08.2023

Q) (i) Stem only likes easy courses
(ii) science courses are hard.
(iii) All courses in basket weaving dept are easy
(iv) BK301 is a basket weaving course
use Resolution to answer the question. What courses would stem like?

(i) $\forall x$ easy(x) → likes(stem, x)
~~easy(course) →~~ ~~course(stem,~~
~~$\forall x$ likes(stem, easy course)~~
(ii) $\forall x$ science(x) → hard(x)
(iii) $\forall x$ ~~basket weaving(x)~~ basket(x) → easy(x)
(iv) basket(BK301)

~~like(stem, x)~~

(i) $\forall x$ ¬easy(x) ∨ likes(stem, x)
(ii) $\forall x$ ¬science(x) ∨ hard(x)
(iii) $\forall x$ ¬basket(x) ∨ easy(x)
(iv) basket(BK301)

hard = ¬easy

~~like(stem x)~~ (i)
likes(stem, x) ∨ ¬easy(x) (iii)

likes(st., x) ∨ ¬basket(x) (iv)
x | BK301

likes(st, BK301).

Stem likes BK301

M:2

1. ¬ course(x) ∨ ¬ easy(x) ∨ likes(st, x)
2. ¬ science(x) ∨ hard(x)
3. ¬ course(x2) ∨ ¬ dp(BV) ∨ easy(x2)
4. i) course(BK301)
   ii) dp(BV)

(1) likes(st, x3) ∨ ¬ likes(st, x3)

x/x3

(4) i) ¬ course(x) ∨ ¬ easy(x) ∨ likes(st, x3)

x3/BK301

(3) ¬ easy(BK301) ∨ likes(st, BK301)

(4) ii) ¬ course(BK301) ∨ ¬ dp(BV) ∨ ~~easy(BK301)~~ likes(st, BK301)

(4.ii) ~~¬ course(BK301) ∨ ¬ dp(BV) ∨ likes(st, BK301)~~

¬ dp(BV) ∨ likes(st, BK301)

likes(st, BK301)

(hostile = enemy)

Q) The law says that it is crime for an American to sell weapons to hostile nations. [The ~~country~~ country Nono, an enemy of America] [has some missiles] & all of its missiles were sold ~~to~~ to it by Col. West, who is an American.
Prove that west is a criminal.

Sol. 1. $\forall x, \forall y, \forall z$; American $(x) \wedge$ hostile $(y) \wedge$ weapon $(z) \wedge$ sells $(x, y, z) \longrightarrow$ criminal (x)

2. Enemy (America, Nono)
3. $\exists x$: missiles $(x) \longrightarrow$ owns (Nono, x)
4. $\forall x$: missiles $(x) \wedge$ own (Nono, x) $\longrightarrow$ sells (West, Nono, x)
5. American (west)

Assumptions (for synonyms) –
6. Enemy of America counts as hostile.
$\forall x$: Enemy (America, x) $\longrightarrow$ hostile (x)
7. Missiles are weapons
$\forall x$: missiles $(x) \longrightarrow$ weapons (x)

Predicate Logic:-
1. $\neg$ America $(x) \vee \neg$ hostile $(y) \vee \neg$ weapons $(z) \vee \neg$ sells $(x, y, z) \vee$ criminal (x)

Assumptions ऐसे think करो की उसका opp. मिल जाए question में।

Date: Page:

~~2. $\forall x$ hostile (¬ Enemy (America, x) ∨ hostile (x)~~

~~Criminal (West)~~ (1)

~~American (x) ∧ hostile (y) ∧ weapon (z) ∧ sells (x, y, z)~~

American (West) ∧ ~~Ex~~ hostile (x) ∧ ~~missile~~ weapons (x) ∧ sells (West, Nono, x)

American (West) ∧ Enemy (Nono, x) ∧ Missile (x) ∧ sells (West, Nono, x) (4)

American (West) ∧ Enemy (Nono, x) ∧ missile (x) (3)

American (West) ∧ Enemy (America, Nono) (2)

American (West) (5)

17.08.2023
Thursday

PROPOSITION

It is raining : R
It is sunny : S

$\left.\begin{array}{l} R \\ S \end{array}\right\}$ $R \rightarrow \neg S$

Consider the following formula in propositional logic.
A prove 'a' using resolution

prove a.

① $(b \wedge c) \rightarrow a$
② b
③ $(d \wedge e) \rightarrow c$
④ $e \vee f$
⑤ $d \wedge \neg f$

Sol: Backward Chaing
where a is available in right part

① ~~b v c~~ $\neg b \vee \neg c \vee a$
② b
③ $\neg d \vee \neg e \vee c$
④ $e \vee f$
⑤ (i) d
(ii) $\neg f$

$\neg a$ ①
$\neg b \vee \neg c$ ②
$\neg c$ ③
$\neg d \vee \neg e$ ④
$\neg d \vee f$ ⑤(ii)
$\neg d$ ⑤(i)

Hence proved

→ Assumption required for S for solving

Q) ① P
② $(P \wedge Q) \rightarrow R$
③ $(S \vee T) \rightarrow Q$
④ T
Prove R.

Barkumar charmy would jail
↓
S should be given

Sol:

① P
② $\neg P \vee \neg Q \vee R$
③ $(\neg S \wedge \neg T)) \vee Q$ → (i) $\neg S \vee Q$ (ii) $\neg T \vee Q$
④ T

$\neg R$ ②

$\neg P \vee \neg Q$ ③ (ii)

$\neg P \vee \neg T$ ④

$\neg P \vee$ ①

" "

~~Q) Consider~~

we can't represent quantifiers

Date: Page:

Q1) If a car has good battery (B), good spark plugs (S), gas (G), good tyres (T) then the car is good (M).

2) If the battery is good (B) then there is electricity (E).

3) If there is electricity (E) & the spark plugs are good (S) then the spark plug will fire (F).

4) If the spark plug fires (F) & there is gas (G) then the engine will run (R).

5) If the engine runs (R) & there are good tyres (T) then the car will move (M).

~~(from 5th statement using support)~~ → ~~By Default~~

Use 2 to 5 & prove (1)

Proposition:-

1) ~~good (bat) ∧ good~~ $(B \wedge S \wedge G \wedge T) \rightarrow M$

2) $B \rightarrow E$

3) $E \wedge S \rightarrow F$

4) $F \wedge G \rightarrow R$

5) $R \wedge T \rightarrow M$

Clause

1) $\neg B \vee \neg S \vee \neg G \vee \neg T \vee M$

2) $\neg B \vee E$

3) $\neg E \vee \neg S \vee F$

4) $\neg F \vee \neg G \vee R$

5) $\neg R \vee \neg T \vee M$

(2) (3)

$\neg B \vee \neg S \vee F$

(4)

$\neg B \vee \neg S$

(5)

(1) →

Date: Page:

~~SK~~

SKOLEMIZATION

$\exists \longrightarrow \forall$ / not $\exists$

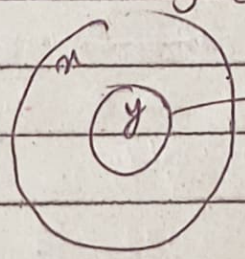
Existential quantifier — Universal quantifier

A method for converting a sentence with '$\exists$' to a sentence without '$\exists$' such that first sentence is satisfiable if and only if the 2nd is.

Case-I Both $\exists$ & $\forall$

To eliminate an $\exists$, replace each occurrence of its variable by a Sokolem function whose arguments are the variables of ~~uni~~ $\forall$ and whose scope includes the scope of $\exists$ being eliminated.

$\forall x, \exists y \longrightarrow f(x)$ Skolem function



$f(x)$ → Argument of x

$f(x)$ → scope of y

y

Ex: ① $\forall x, \forall y, \exists z : P(x, y, z)$

$\forall x, \forall y : P(x, y, f(x, y))$ → scope: z

② $\forall x, \exists y : Pred(x, y)$

$\forall x : Pred(x, succ(x))$ — scope: y

predecessor

$succ(x) = y$

successor (No relation with Pred)

③ Everyone has a brain.

$\forall x : Person(x) \rightarrow \exists y : Brain(y) \wedge has(x, y)$

$\forall x : Person(x) \rightarrow Brain(\underbrace{B(x)}_{y}) \wedge has(x, \underbrace{B(x)}_{y})$

Case - II

If the existential quantifier being ~~illuminated~~ eliminated is not within the scope of $\forall$, then the Skolem fn will have no argument that it is a constant.

Ex. ① $\exists_x P(x)$

$P(A)$

② $\exists x : crown(x) \wedge onhead(x, John)$

$\hookrightarrow crown(C_1) \wedge onhead(\cancel{John}\ C_1, John)$

C_1: Skolem constant.

2/10/2023
Monday

Date: Page:

ALGORITHMS :-

① Eliminate $\rightarrow$ or $\rightleftarrows$

$a \rightarrow b \approx \neg a \vee b$

$a \rightleftarrows b \approx (a \rightarrow b) \wedge (b \rightarrow a)$

$(\exists x)(P(x) \wedge (\forall y)(f(y) \rightarrow h(x,y)))$

$\hookrightarrow (\exists x)(P(x) \wedge (\forall y)(\neg f(y) \vee h(x,y)))$

② Eliminate negations or reduce the scope of negation to one literal.

$\neg(\neg P) \approx P$

$\neg(P \wedge Q) \approx \neg P \vee \neg Q$

$\neg(\forall x \in S, P(x)) \approx (\exists x \in S, \neg P(x))$

or

$\neg(\exists x \in S, P(x)) \approx (\forall x \in S, \neg P(x))$

New

effect on quantifier

③ Standardize variables within a well-defined formed formula so that the bound or free variables of each quantifier have unique names.

$(\exists x)(\neg P(x) \vee (\forall x)(P(x))) \approx (\exists x)(\neg P(x) \vee (\forall y)(P(y)))$

unique

④ Advanced Clause form step

If there $\exists$ quantifier, eliminate them using Skolem function

$(\exists x)(P(x)) \approx P(A)$

$(\forall x)(\exists y)\, k(x,y) \approx (\forall x)\, k(x, f(x))$ f: Skolem f^n

scope y

⑤ Convert to PNF (Prenex Normal Form)

$(\exists x)(\neg P(x) \vee (\forall y)(P(y))) \approx$

$(\exists x)(\forall y)(\neg P(x) \vee P(y))$

(Bring quantifiers to LHS)

⑥ Convert to CNF

$P(x) \wedge Q(x) \approx$ $P(x)$ $Q(x)$

2 different.

⑦ Drop the prefix

ALGORITHM IN ~~PRED~~ RESOLUTION in PREDICATE LOGIC

Given set : F : predicate form Prove : P

⓪ Convert the English statements to predicate form] X

① Convert all statements of F in ~~Predicate~~ Clause form

② Negate P & convert the result to clause form & add it to the set of clauses obtained in step ①

③ Repeat until either a contradiction is formed or no progress is made. or a predetermined amt. of efforts has been ~~spent~~ expended.

a) Contradiction □ empty "means opp. was true"

b) No progress Opp. can't be found.

c) No of ~~to~~ iterations or depth of search

(4) a) Select 2 clauses, call these parent clause C_1 & C_2

b) Resolve them together. The resolvement will be the disjoint of all the clause of both parent clause with appropriate substitution performed with following exceptions.

If there is one pair of literal T_1 & $\neg T_2$ such that one of the parent clause contains T_1 & other contains T_2 & if ~~$T_2 \leftarrow T$~~ T_1 & T_2 are ~~unifiable~~ unifiable, then neither T_1 nor T_2 should appear in the resolvent. T_1 & T_2 are called as complementary literals.

different algorithm

(unification calls ~~on~~ for one variable)

Use the substitution produced by the ~~notification~~ unification to create the resolvement.

∴ Call Unification fn multiple time)

c) If the resolvent is the empty clause, then a contradiction has been formed, if it is not then add it to the set of clauses available to the productions

b) $a \vee \neg b$ $\quad$ $b \vee c$

$a \vee c$

Unification

Man (Marcus) $\quad$ $\neg$ man (x)

x / Marcus

" "

man (marcus) ∨ ¬ruler (caesar) ¬ man (x) ∨ ruler (y)

x / marcus

¬ruler (caesar) ∨ ruler (y)

y / caesar

" "

Unification is called for only one at a time
∴ twice calling

ALGO : Unify (L_1 , L_2)

① If L_1 & L_2 are both variables or constants then:

5 a) If L_1 & L_2 are identical then return NIL

b) Else if L_1 is a variable then if L_1 occurs in L_2 then return fail. {FAIL}
else return { L_2 / L_1 }

c) Else if L_2 is a variable then if L_2 occurs in L_1 then return {FAIL}
else return { L_1 / L_2 }

d) Else return {FAIL}

1

② If initial predicate symbol of L_1 & L_2 are not identical then return {FAIL}

2

③ If L_1 & L_2 have different no. of arguments then return {FAIL

substitution set

Date: Page:

3

(4) Set SUBST to NIL

4

(5) For i ← 1 to no. of ~~ra~~ arguments of L1

a) Call unify with the ith arg. of L1 & its arg. of L2 putting the result in S.

b) If S contains FAIL then return {FAIL}

c) If S is not equal to NIL then

i) Apply S to the remainder of both L1 & L2

ii) SUBST := APPEND (S, SUBST)

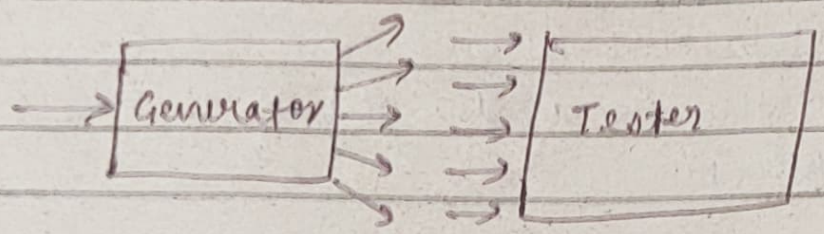
Return SUBST

22.08.2023
Tuesday.

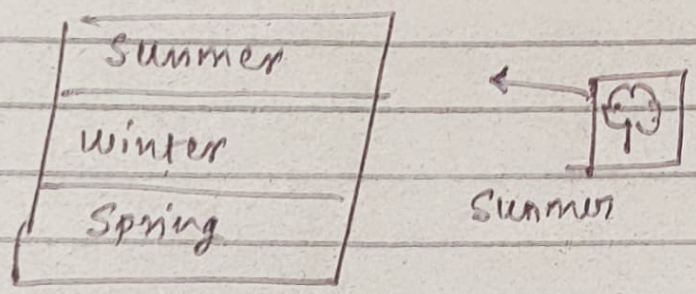
SEARCH ALGORITHMS

GENERATE AND TEST

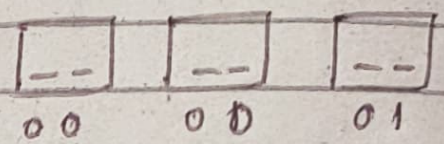
1000000
300
3333.3 h



Ex. i) Textbook



ii) Locker



$(100)^3 = 1M$ generates

& tests

Infomation : each no. is prime

prime < 100 ≈ 25

$(25)^3$ generative patterns

less testings.

Good Generator

1. Informed
2. Systematic
3. Non Redundant

Heuristic

Infomation, Judgmental based to decrease no. of generatives

Production Systems

Date: Page:

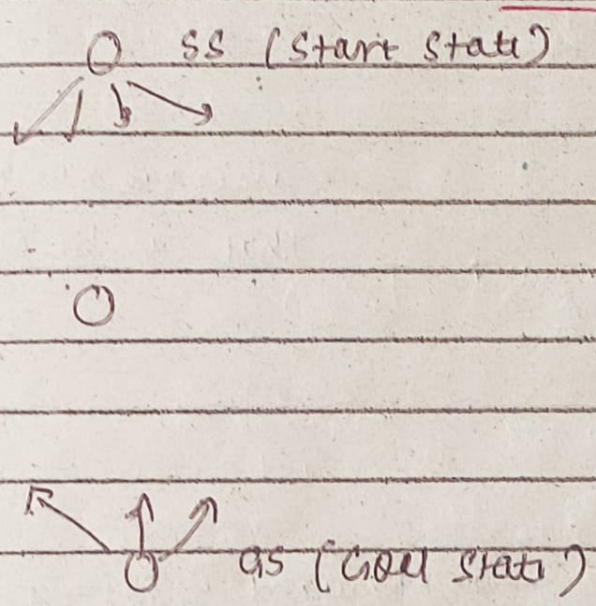
Search Algos

Uniformed / Blind	Informed / Heuristic
BFS	Best F.S.
DFS	A* search
Depth Limited Search	AO* search
Iterative Deepening DFS	Hill Climbing
Bidirectional Search	Generate and Test
Uniform Cost Search	

AI

Bidirection Search

BIDIRECTIONAL SEARCH



Start from SS & GS simultaneously & where these meet is the point where we know we meet G.S.

Production Rule

LHS → RHS

⋮

Factors to apply ① or ②
which ?

(i) Find no. of IS & G.S

Go from low to high

(unknown है वह अभी Easy) → Not Rule

① Forward Search
Forward Reasoning
Data Driven Search

IS
↓
RHS
⋮
GS

When we start from initial state
Search this in LHS
then RHS is Next State

② Backward
Backward
Goal Driven

G.S
↓
LHS
⋮
IS

(ii) Branching factor :

$(bf)_{\text{forward dir}^n}$ $(bf)_{\text{backward dir}^n}$

average no. of nodes that can be emerged.

Take lesser one.

(iii)

New facts आ सकते हैं तो → Forward Reasoning

Ex: Cricket (New cond^n आ सकती है।

only answer

(iv) Let new decide which to choose.

Cricket game

i) N(IS) < N(G.S) — FS

Win Tie Lost Draw

ii) $(bf)_{\text{forward dir}^n}$ < $(bf)_{\text{backward dir}^n}$ — FS

iii) Cricket # New facts are there — FS

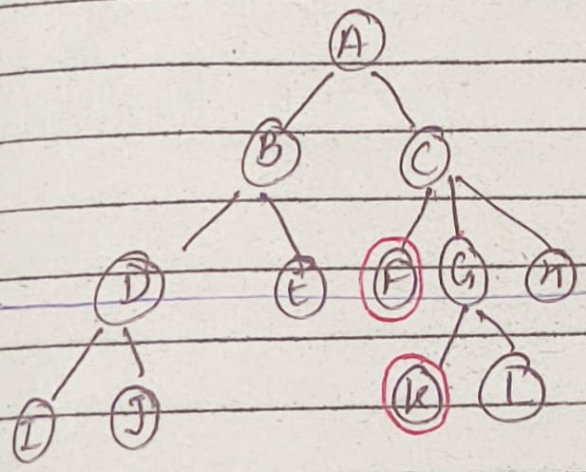
iv) Your → Forward search (FS)

Why against i) ii) & iii)

DEPTH-LIMITED DFS

predefined depth and apply DFS upto that depth

(can be rule to look one down the depth or any)



Depth limited DFS

depth = 3

○ G.S.

Search till this only.

⊙ continue till max-depth given predefined or we reach Goal state

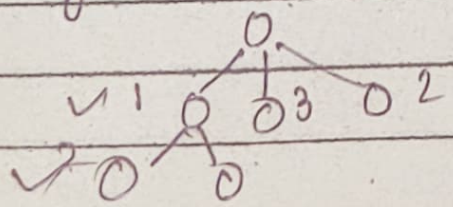
UNIFORM COST SEARCH ALGO

Select lower cost Node

& explore it.

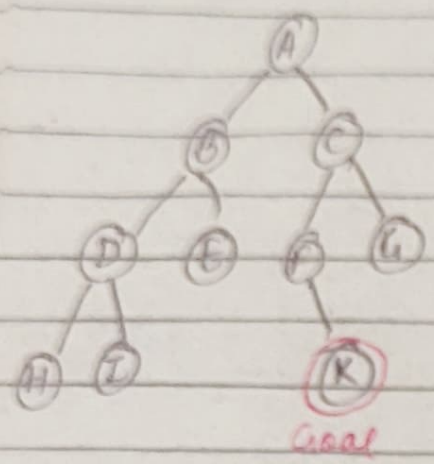
$\boxed{\text{Priority Queue}}$

If cost is same → use BFS over it same.



21.08.2023
Monday.

ITERATIVE DEEPENING A* SEARCH



Iteration	Nodes order
1	A
2	A, B, C
3	A, B, D, E, C, F, G
4	A, B, D, H, I, E, C, F, K, G

Disadvantage:-
many repetitions of nodes in different iterations.

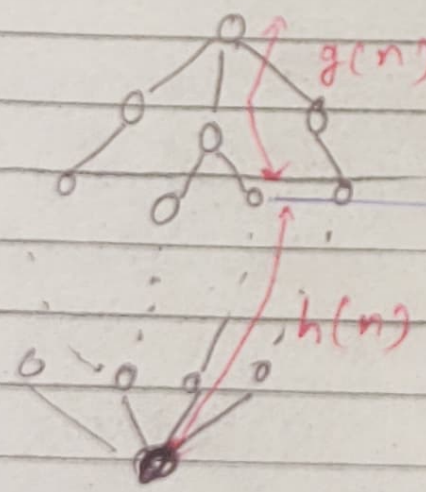
A* Algorithm

Informed Search

h : heuristic function

f(n) : fitness value of Node 'n' / Evaluation function value

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



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

$g(n)$ → Cost till Start to nth State

$h(n)$ → estimated cost from nth to Goal State

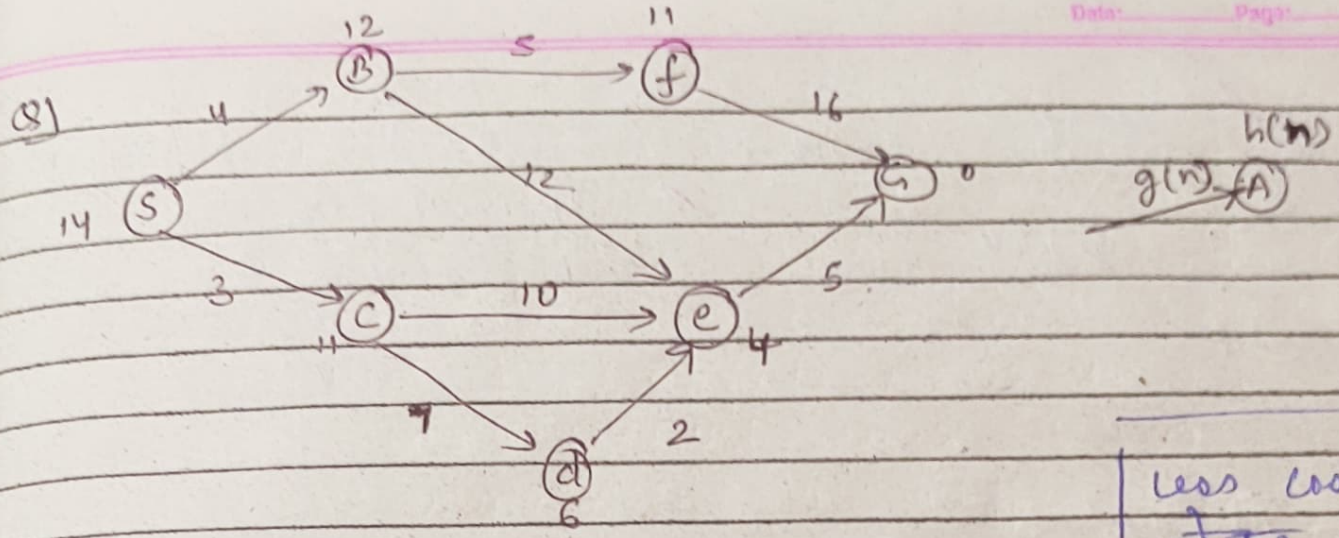
Exact

Overestimate
ज्यादा माना cost

Underestimate
कम माना cost

S : Start
G : Goal (Assume)

Q)



Less cost वाला लेकर आगे बढ़ो

A* will ~~written~~ return most optimal value if there exists solution } Admissibility (Property of A*)

Step ①

S → B ~~4+12~~ S → B 4 + 12 = 16
S → C 3 + 11 = 14 ✓

Step ②

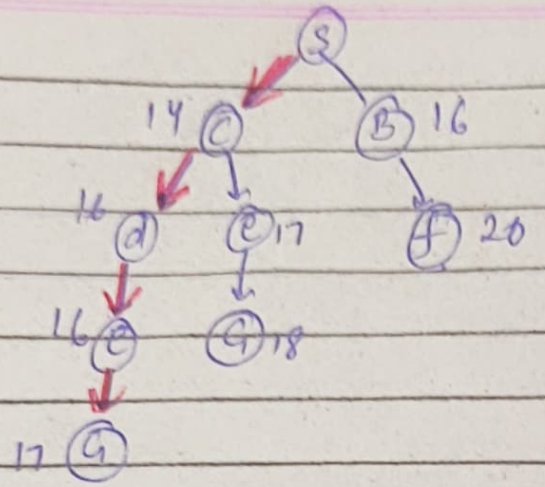
S → C → e 3 + 10 + 4 = 17
S → C → d 3 + 7 + 6 = 16 ✓

Step ③

S → C → d → e 3 + 7 + 2 + 4 = 16 ✓
S → b = 16
∴ S → b → f = 4 + 5 + 11 = 20

Same

Step ④ S → C → d → e → g 3 + 7 + 2 + 5 + 0 = 17
S → C → e = 17
∴, S → C → e → G = 3 + 10 + 5 + 0 = 18



Graph.

$s \to c \to d \to e \to g$ (17)

Optimal path
(A* properties)

Algorithm : A*

undirected graph

① Place the initial node in the OPEN LIST.

② Check if the OPEN LIST ~~is~~ is empty or not, if the list is empty then return ~~falet~~ failure & stop.

③ Select the node from the OPEN LIST which has the smallest value of estimation function $f(n) = g(n) + h(n)$ of node 'n', if 'n' is goal return & quit.

④ Expand node 'n' & generate all of its successor & put 'n' into the CLOSED LIST. For each successor 'n', check whether n' is already in the OPEN ~~LIST~~ or CLOSED LIST, if not then ~~compute~~ compute the estimation function for n' & place it into OPEN list.

⑤ Else if node n' is already in OPEN & CLOSED, then it ~~is~~ should be attached to the back pointer which reflects the lowest of (n') value.

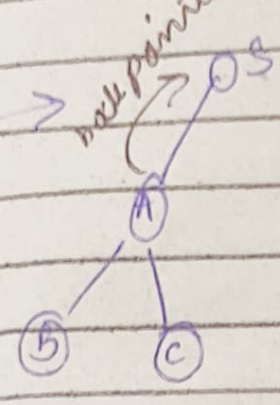
⑥ Return to step 2.

open में Fitness value के

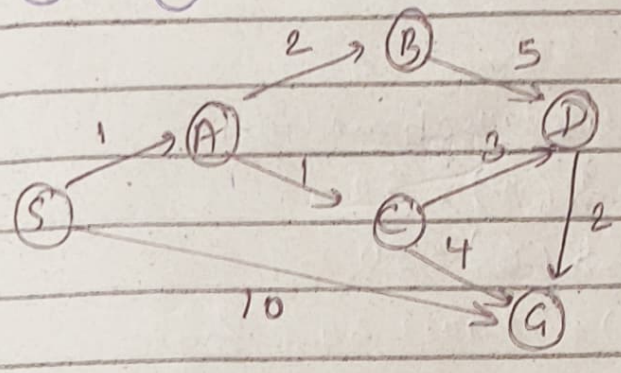
(A)

Once सारे successor Nodes किसी किसी Nodes के Explore हो जाये तब ० अ. उस Node (A) को Closed list में डाल दो।

Estimation function == Fitness function



Q)



Node	h(n)
S	5
A	3
B	4
C	2
D	6
G	0

(S,5) OPEN : S CLOSED :

Step:1 S→A : 4 ✓
S→G : 10

OPEN : A, G → order matter
CLOSED : S → Not matter

Step:2 S→A→C : 4 ✓
S→A→B : 7

OPEN : C, B, G
CLOSED : A, S

Step:3 S→A→C→G : 6 ✓
S→A→C→D : 11

OPEN : B, G, D
CLOSED : A, S, C

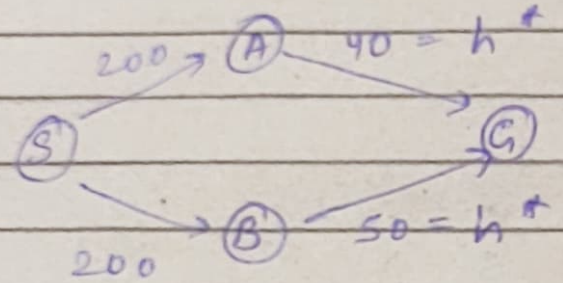
04.09.2023
Monday

Underestimation में Admissibility maintain रहती।

$h(n)$ - estimated
$h^*(n)$ - actual

$h(n) \leq h^*(n)$
Underestimation

$h(n) \geq h^*(n)$
Overestimation
ज्यादा Price बताकर
कम निकला Price



$g(A) = 200$
$g(B) = 200$

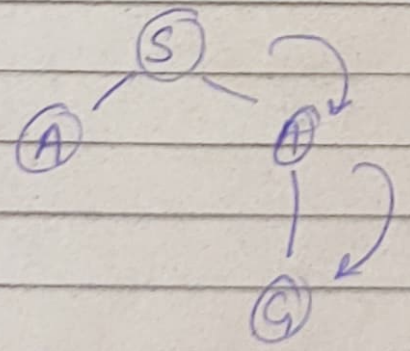
Overestimation

$h(A) = 80$
$h(B) = 70$

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

$f(A) = 200 + 80 = 280$
$f(B) = 200 + 70 = 270$ ✓

$f(G) = 250 + 0 = 250$ ✓

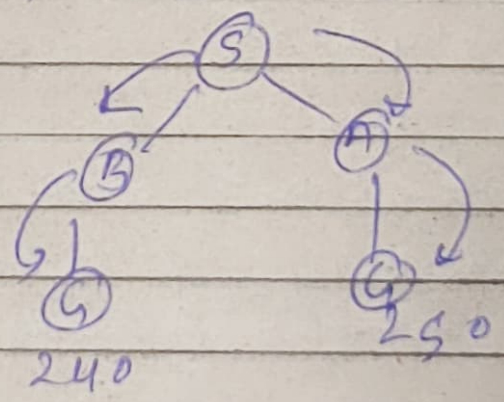


Underestimation

$h(A) = 30$
$h(B) = 20$

$f(A) = 200 + 30 = 230$
$f(B) = 200 + 20 = 220$ ✓

$f(G) = 250 + 0 = 250$
$f(G) = 240 + 0 = 240$ ✓



$h(n)_{\text{goal state}} = 0$ always

Take min. $h(n)$

8 PUZZLE PROBLEM

$h(n)$ = No. of tiles not in correct position. → heuristic f^n: Hamming dist.

board 2

	1	2	3
		5	8
4	4	7	6

1	2	3
4	5	6
7	8	

up

	1	2	3
	4		8
3	7	5	6

left

	1	2	3
	4	5	8
3		7	6

right

	1	2	3
	4	5	8
2	7	6	

HILL CLIMBING

Initial State → IS & GS should → Goal State

① dist. betn IS & GS reduce

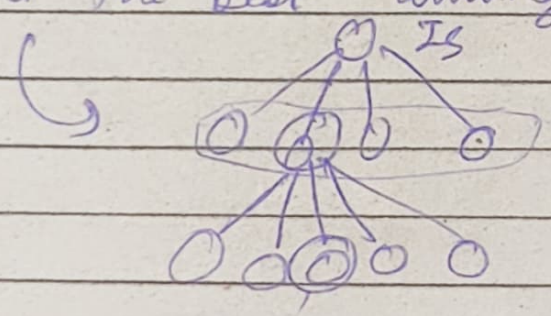
② उपर कदम रख रुके बिना पहुंचना है।

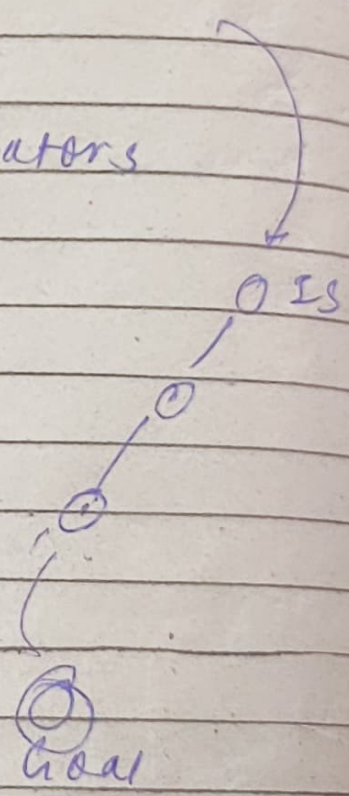
① SIMPLE HILL CLIMBING

② STEEPEST ASCEND CLIMBING

① operator apply करके Next step from current
↳ while (you don't get)

② Get all successors through operators
& get the best among them.





It can be
Cost
(less)

Profit
(most)

Goal

05.09.2023
Tuesday

a kind of dfs search

SIMPLE HILL CLIMBING

{NOT necessarily optimum sol^n}

Algo:

1. Evaluate the I.S. If it is G.S. then return it & Quit ~~otherwise continue with the IS as C.S~~
2. Loop until a solution is formed or until there are no new operation left to be applied to the counter state.

a) Select an operator that has not yet been applied to the current state & apply it to ~~the~~ produce a new state

b) Evaluate the new state
   i) if it is G.S. then return & Quit.
   ii) if it is not a G.S. ~~then~~ but it is better than the current state, then make it the current state.
   iii) if it is not better then the current state then continue in the Loop.

a kind of bfs search

IS: Initial state
CS: Current state

Steepest HILL ASCEND CLIMBING:

a kind of bfs

① Evaluate the I.S. If it is G.S. then return it & Quit. Otherwise continue with I.S. as C.S
② Loop until a sol^n is found or until there are no new operators left to be applied to C.S.

a) Let Succ (SUCC) be a state such that

any possible successor of the C.S. will be better than SUCC.

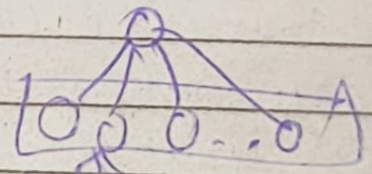
b) For each operator applies to the current state do;

(i) Apply the operator & generate the new state.

(ii) Evaluate the new state: If it is a G.S. then return it & Quit.
If not, compare it to SUCC. If it is better, then set SUCC to this state.
If it is not better, leave SUCC as it is.

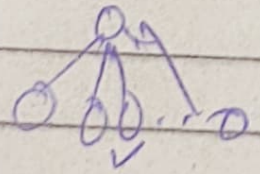
c) If the SUCC is better than C.S. to SUCC.

Sense of better: Distance betn C.S. to G.S
↳ कम है तो बेहतर



Succ = min. value.

Space complexity is less than A* & AO* Algo
↓
Reason



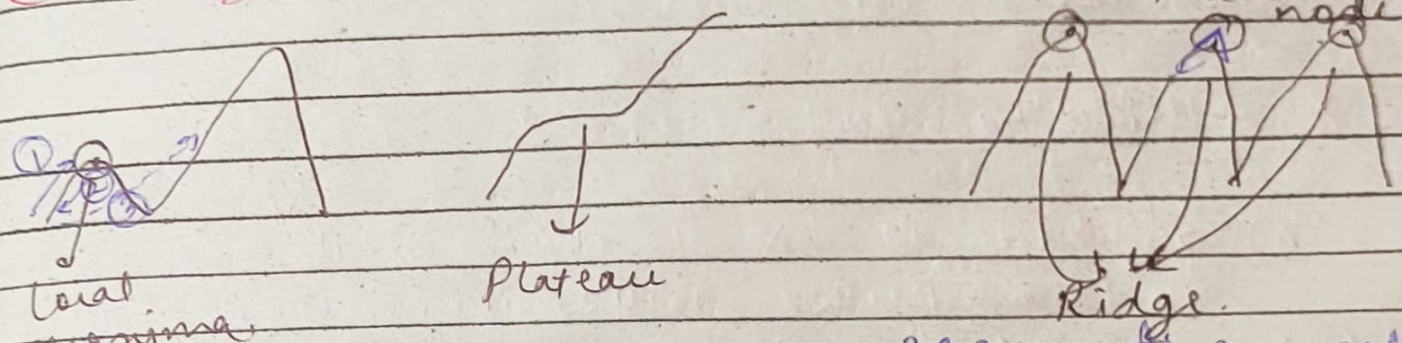
One got better discard everyone else of that level.
Thus saving space.

Disadvantages of Hill Climbing —

① Local Maxima : maxima है पर Global ~~or~~ can't be decided.

② Plateau : flat area of ~~an~~ search space.

③ Ridge : Several adjoining nodes have the higher values than the surrounded node



किसी collage को RGPV में 200 colleges के First Runner करते इनमें First है।

Deal with Disadvantages

① Local Maxima : Backtrack to some earlier nodes & try going in different directions

② Plateau : Make a big jump in some dir^n & try to go a new section of a search space.

③ Ridge : Try different paths at same time.

Ex. of real life:

① music system ~~of sound~~ Tuning parameters pleasant to all.

② Setting up the carburetor
First high rate
& then keeping constant.

O - X Tic Tac Toe.

$h(n) \rightarrow$ (कितने पर X पर win करता है -
कितने पर O पर win करता है)

$h(n) = X(n) - O(n)$
or $E(n)$

$f(n) = g(n) + h(n)$
↳ No. of levels explored

Q) ① 15 Puzzle (same as 8 Puzzle)

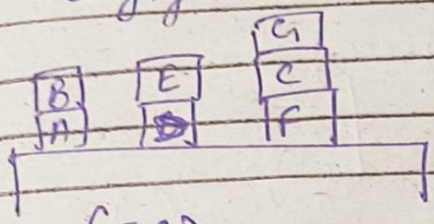
② Travelling salesman problem (TSP)
↳ $h(n)$: sum of all cities ~~at~~ covered so far (small value better)

11.09.2023
Monday

Date: Page:

BLOCK WORLD PROBLEM :-

Initial configuration



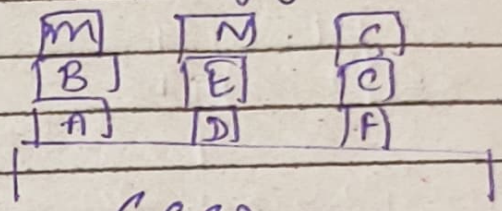
(IS)

Robot arm is given

↓

it can put block on table or put on some other block heap.

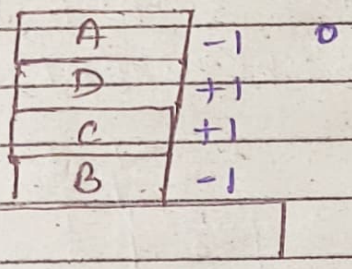
Goal configuration



(GS)

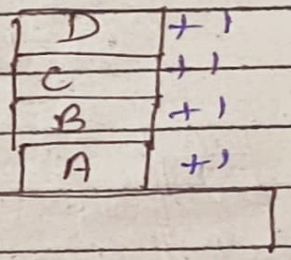
Local & Global heuristic ~~problem~~ functions :-

IS :



0

GS :



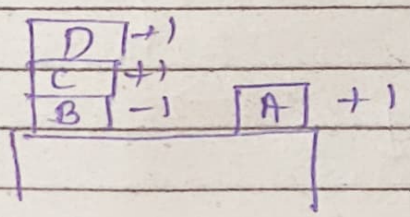
(1)

+1 : for each block that is residing on the thing it is supposed to be residing on.

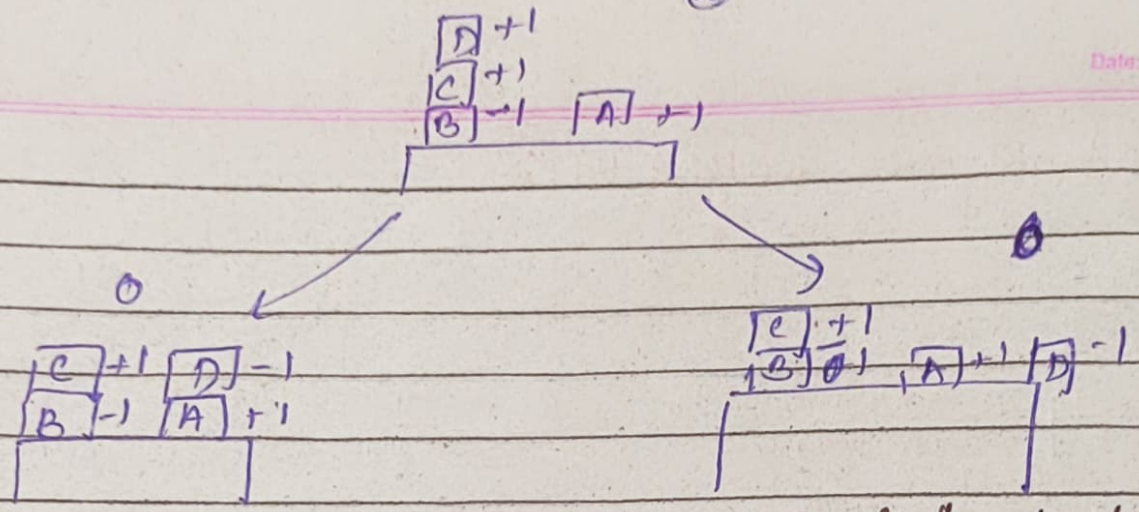
-1 : for each block that is residing on a wrong thing

Local Heuristic

(2)



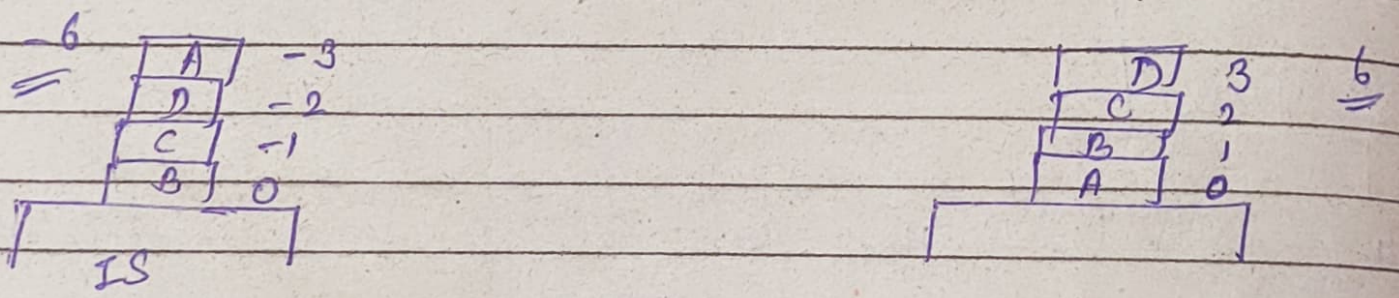
(Next page)

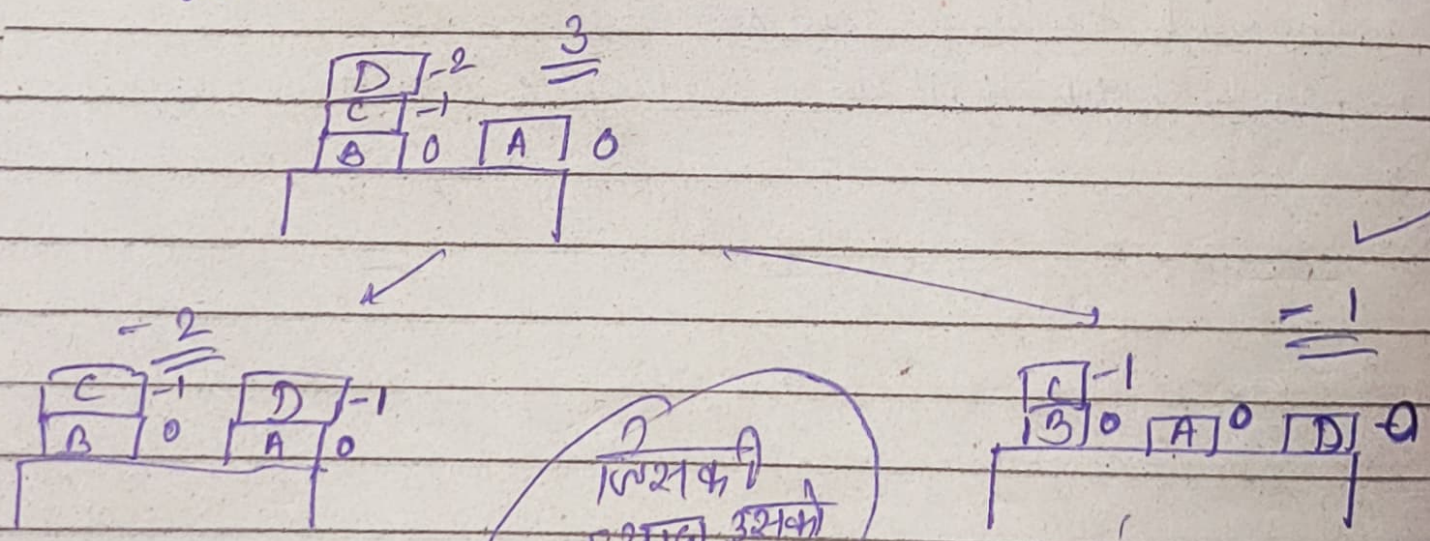


Global heuristic :- (local heuristic fails to decide) local maxima problem

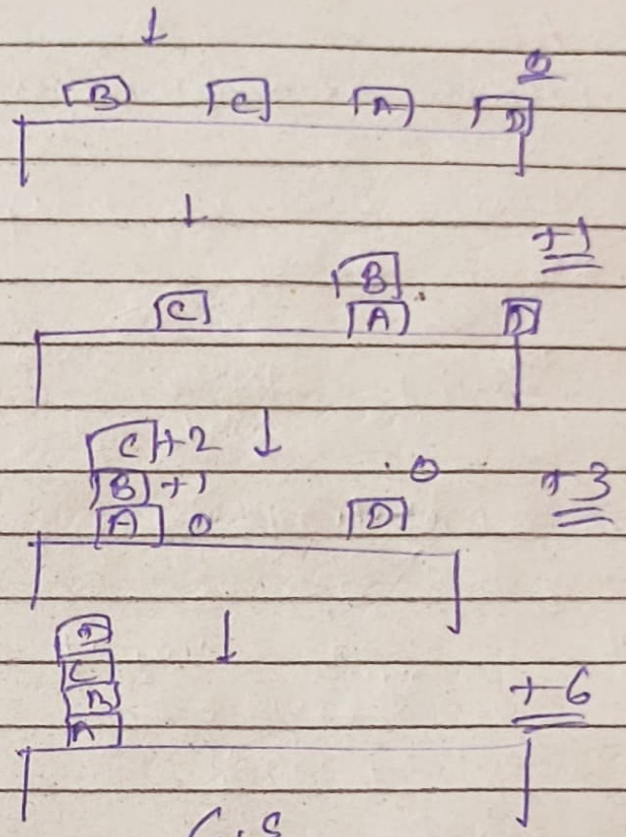
→ For each ~~prob~~ block, that has a correct support structure +1 to every block in the support structure

→ For each block that has a wrong support structure -1 to every block in the support structure





जिसकी जरूरत उसको आगे बढ़ाओ

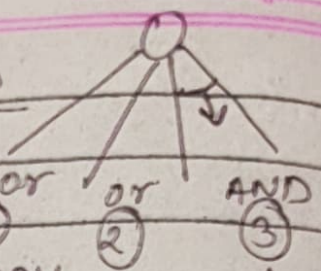


G.S

AO*

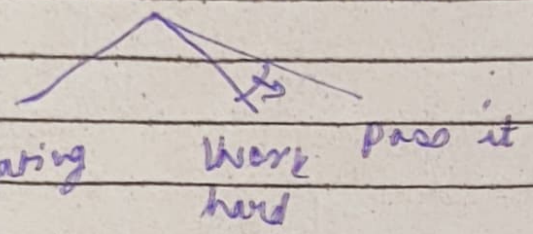
And-or Graph

Problem Decomposition

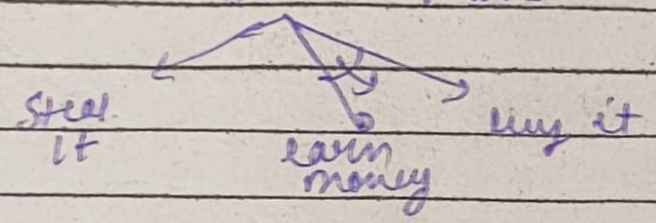


3 ~~choose~~ किए हो दोनों 212 ने तो।

Want to pass in exam

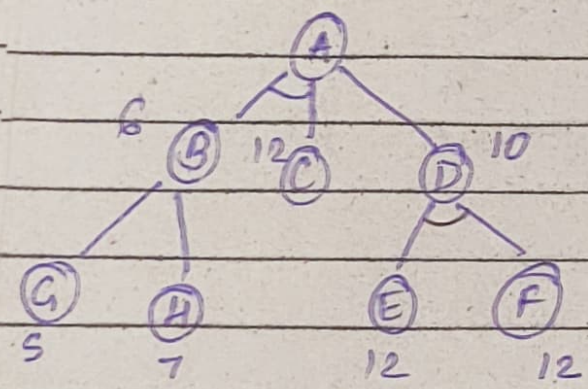


Mobile Phone



A* : gives optimal solⁿ always under underestimation

↳ diff. betⁿ A* & AO*



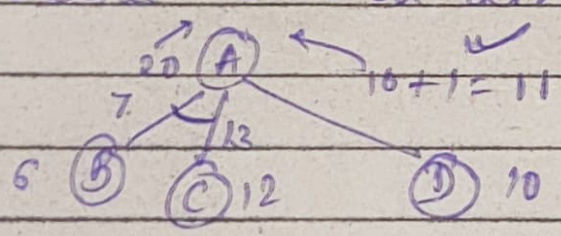
~~heur~~

Heuristic value = Cost + No. of nodes in And Arc

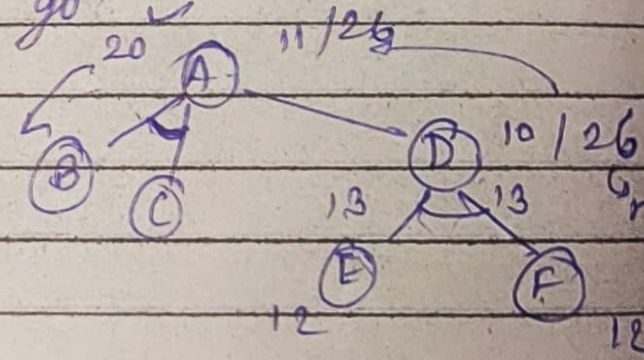
Takes minimum level wise

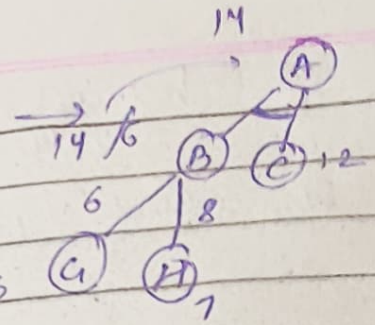
refer it upside

E2 level ₹2 Estimise करके upward जाएगा And decide which dirⁿ to go!



→





Algorithm : Best first search

Step.1 Place the starting node into OPEN LIST.

Step.2 If the OPEN LIST is empty, stop & return.

Step.3 Remove the node 'n' from OL which has the lowest value h(n) & place it in the CLOSE LIST (CL)

Step.4. Expand node 'n' & generate all of its successor of node 'n'. For each successor of node 'n' find out whether any node is Goal node or not.

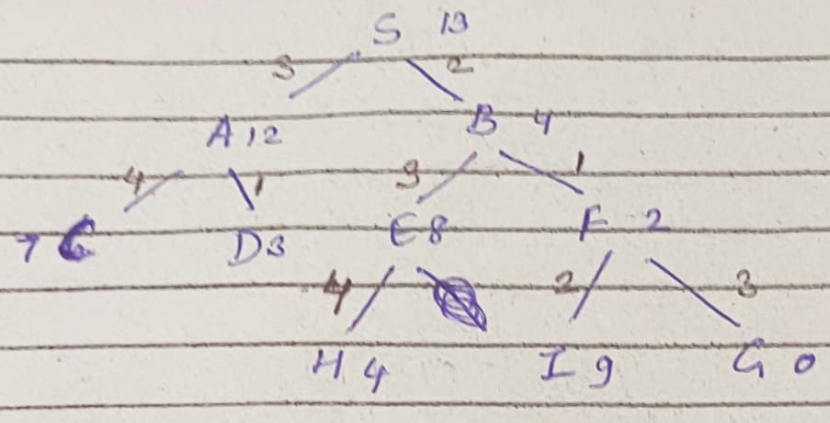
Step.5. If any successor node is a Goal node then return it & terminate the search with SUCCESS.

else go to Step.6.

Step.6. ~~If can~~ For each successor node evaluate "f(n)" & check if the node is already in ~~OPE~~ OL or CL.

If the Node is ~~in~~ not in any list then place it in OL

Step.7. Return to Step 2.



→ heuristic value

→ cost of path

min of OL & CL

1.
OL : S
CL : 2

2.
OL : A B
CL : S 2

3.
OL : A E F
CL : S B 2

4.
OL : A E I G
CL : S B F 2

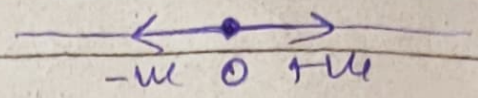
5. OL : A E I
CL : S B F G ✓

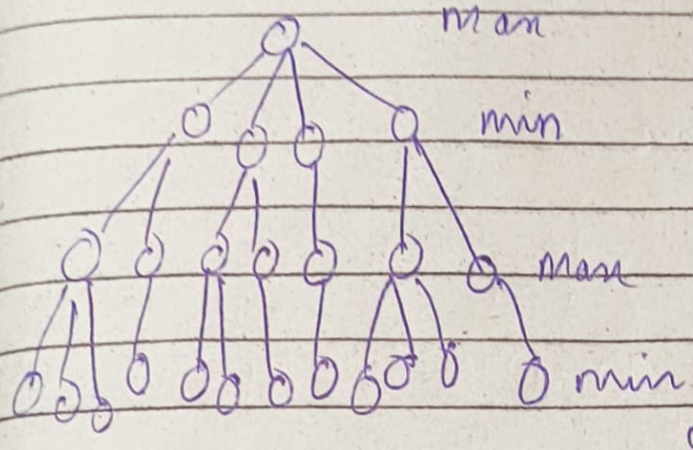
Goal ∴ Terminate

$f(F) = h(F) + \text{cost of path till } F$
$= 2 + (1 + 2) = 5$

12.09.2023
Tuesday

(Opponent) (me)
min max





deep-enough = T

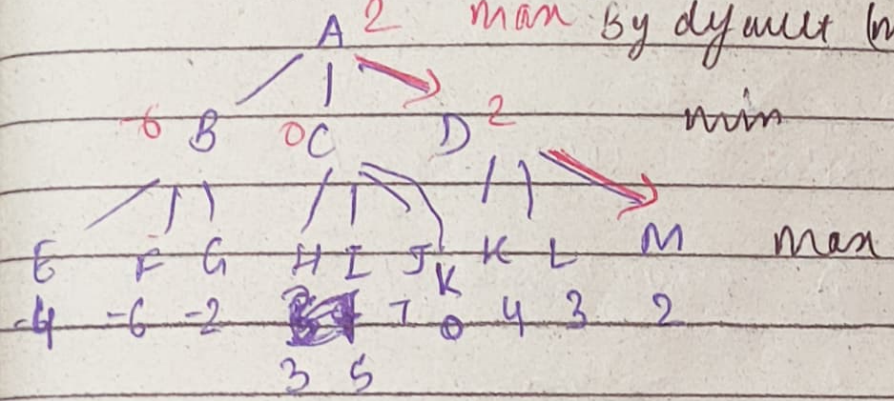
evaluation function

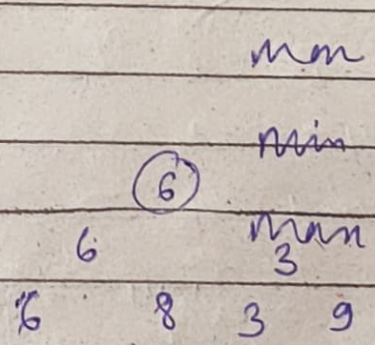
cutoff branches initially { α-cutoff, β-cutoff

↳ search would be reduced

left-right α-β cutoff
right-left α-β cutoff.

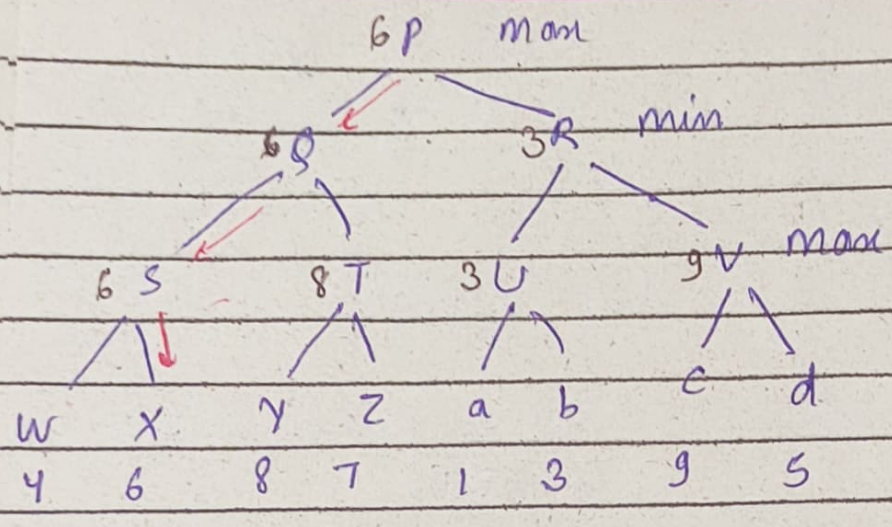
Consider the following tree. Find out value of root using mini-max search Algo.





Date: Page:

P = 6



β will come at minimizing level

β min upperbound

β cutoff / β Pruning

α max lower-bound

α cutoff / α Pruning

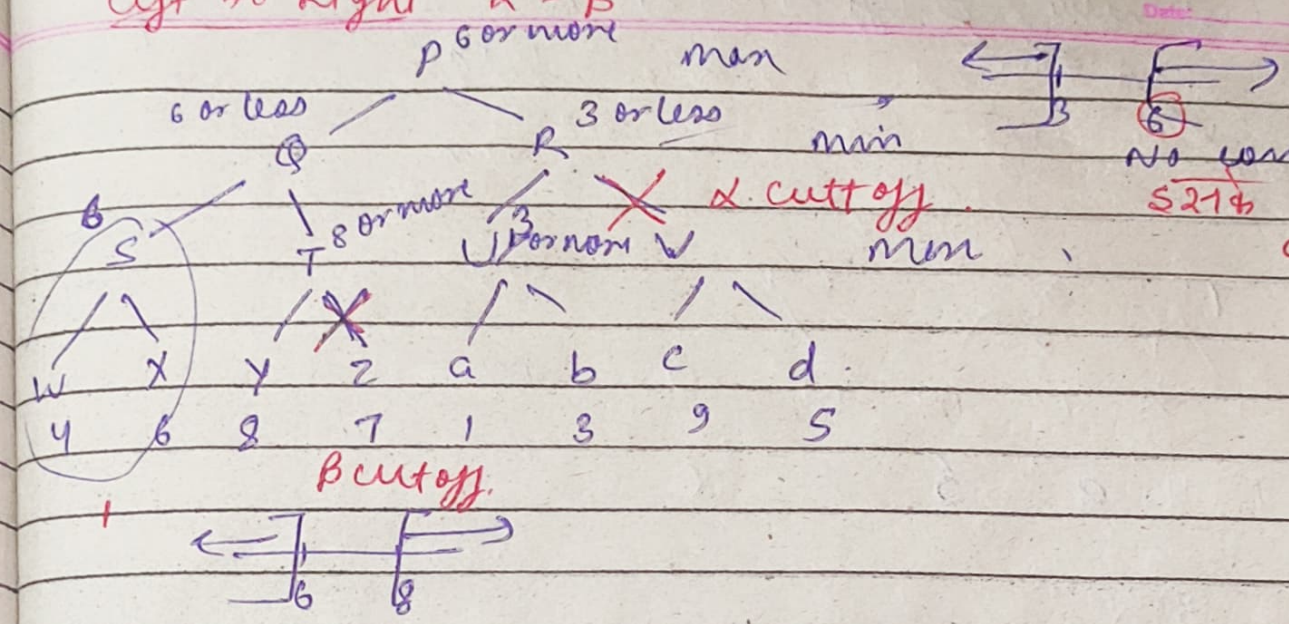
Same for β.

① lowerbound at minimizing level is ~~β~~ α value

② cutoff occurring due to α value is α - cutoff.

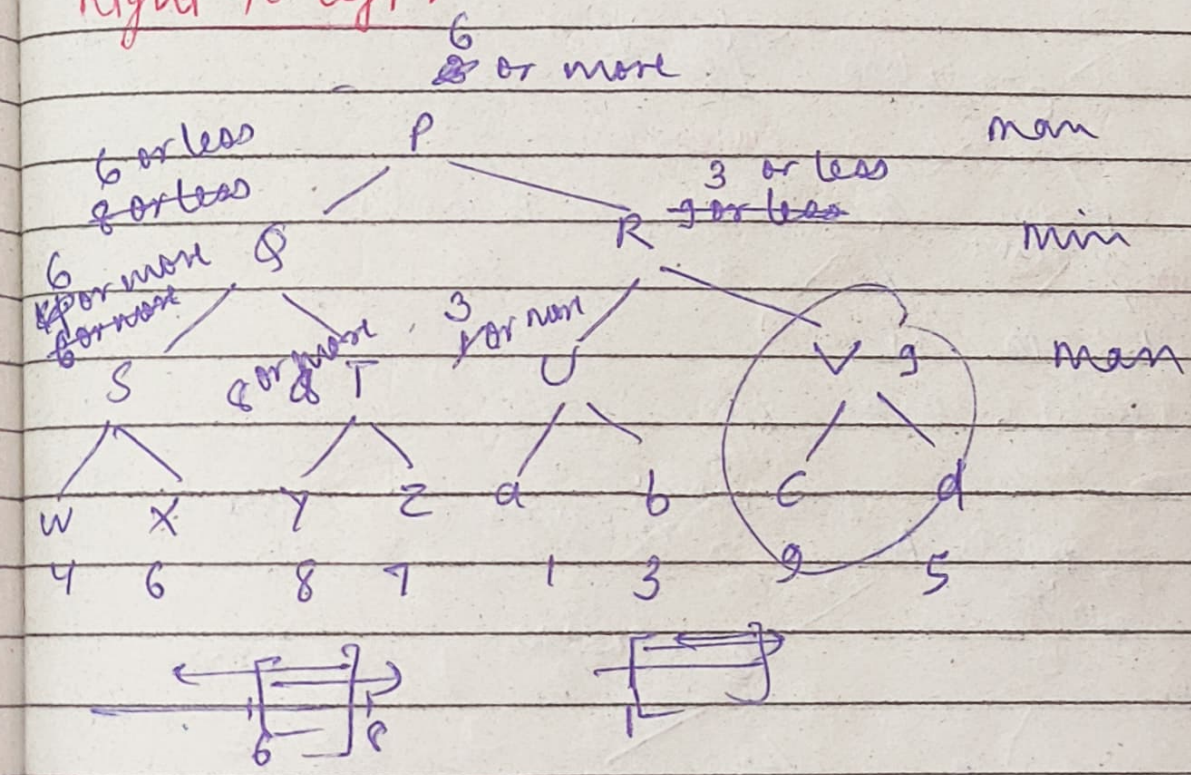
Range lim बता के करो।

Left to Right α − β



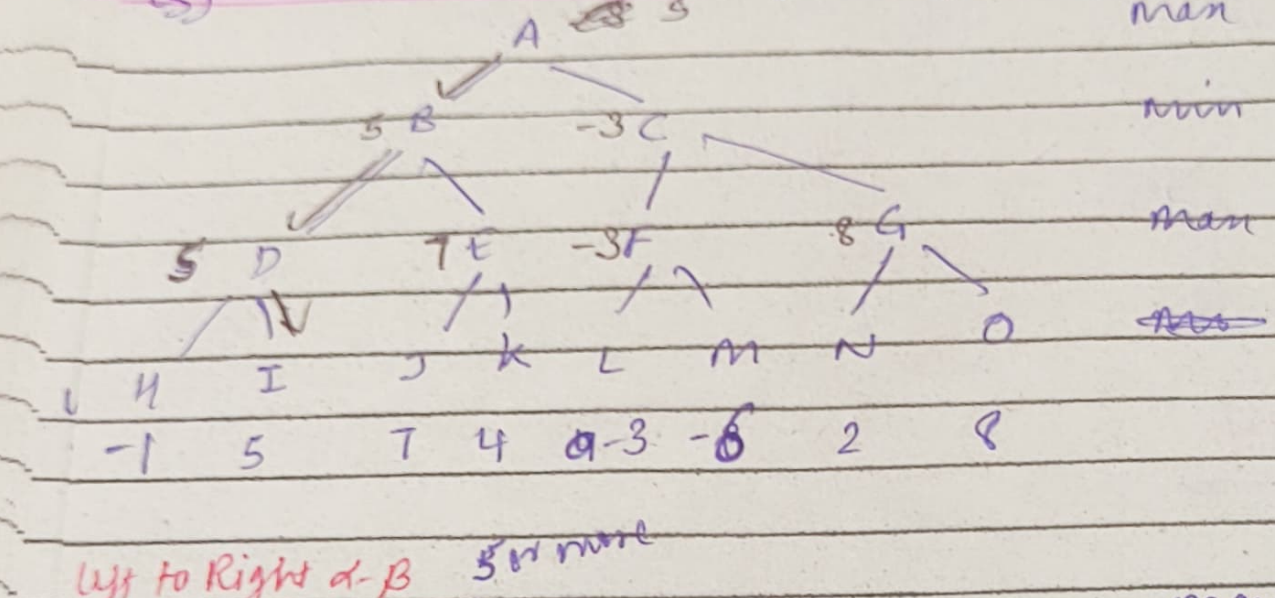
No common
5 21 6 (12 0)
काटा

Right to Left:

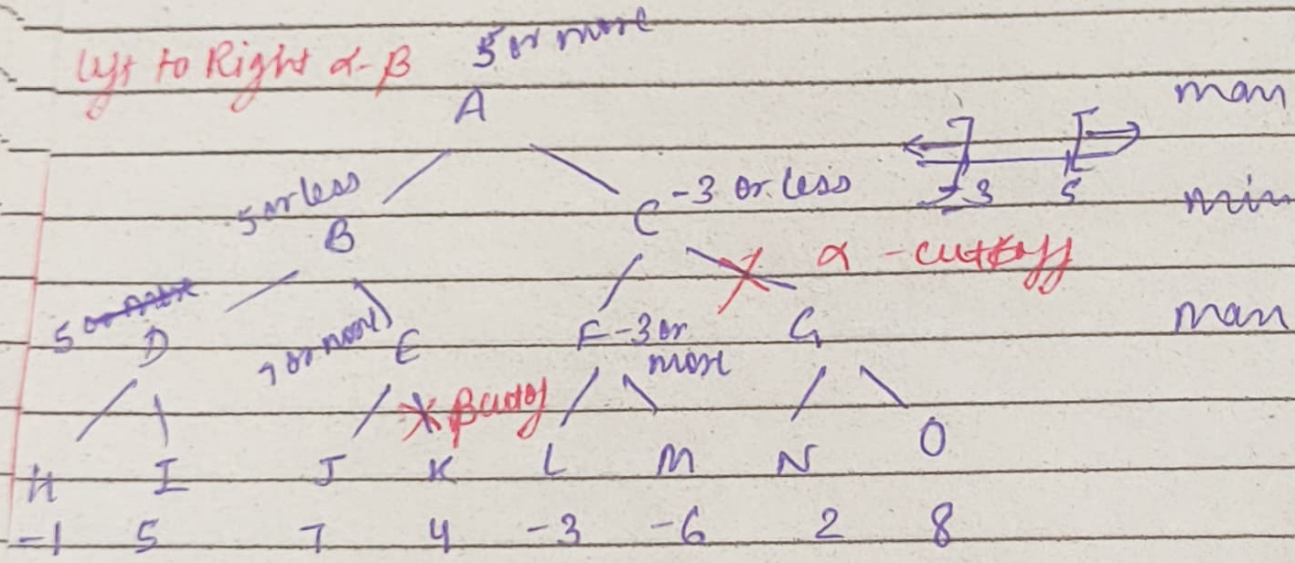


No cutoff

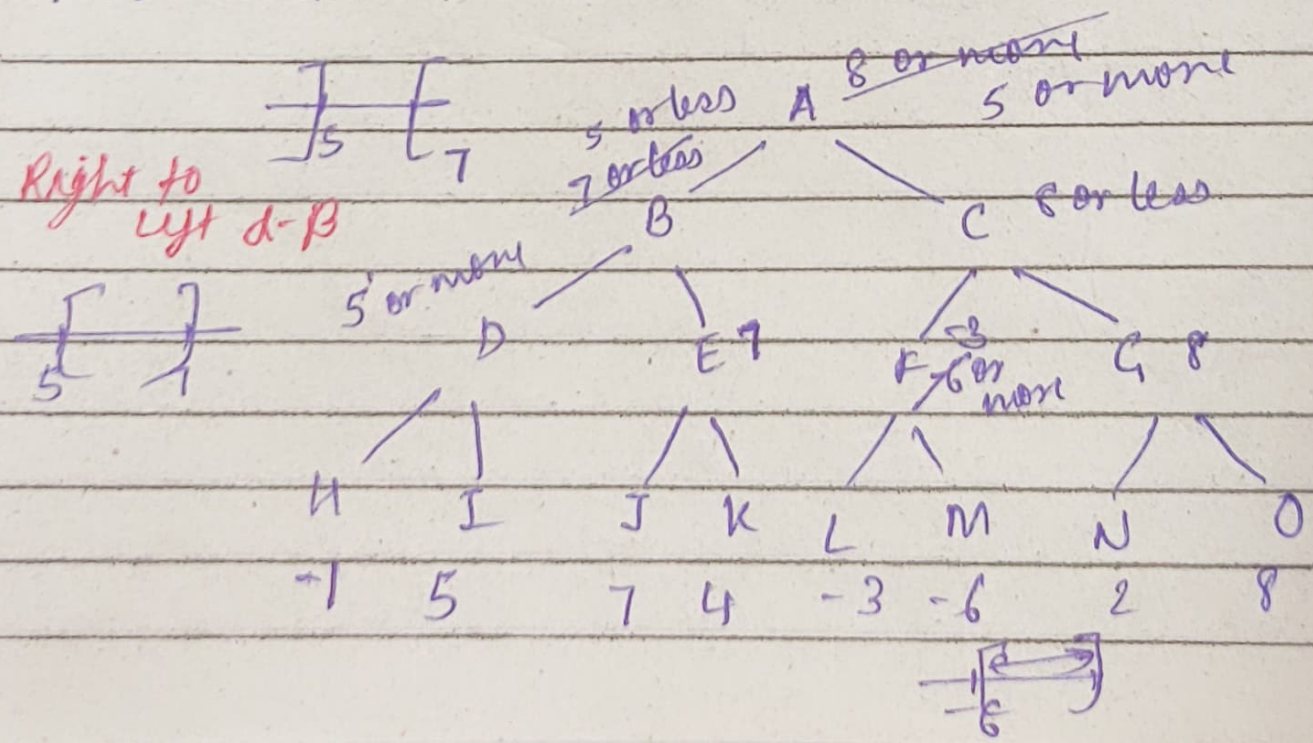
Q) Find Root value, Left to Right α-β, Right to Left α-β



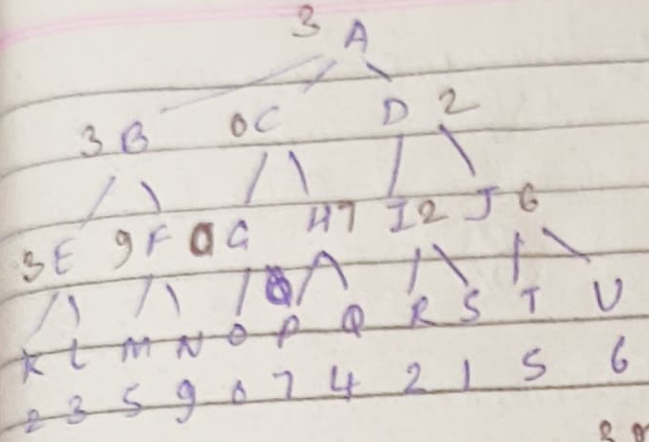
Left to Right α-β



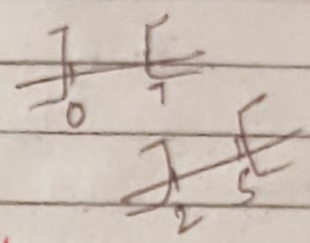
Right to Left α-β

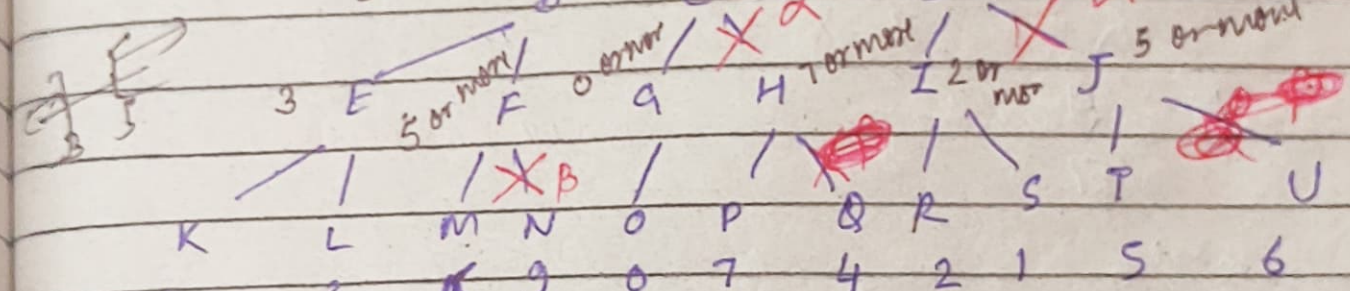


Tic Tac Toe using α-β



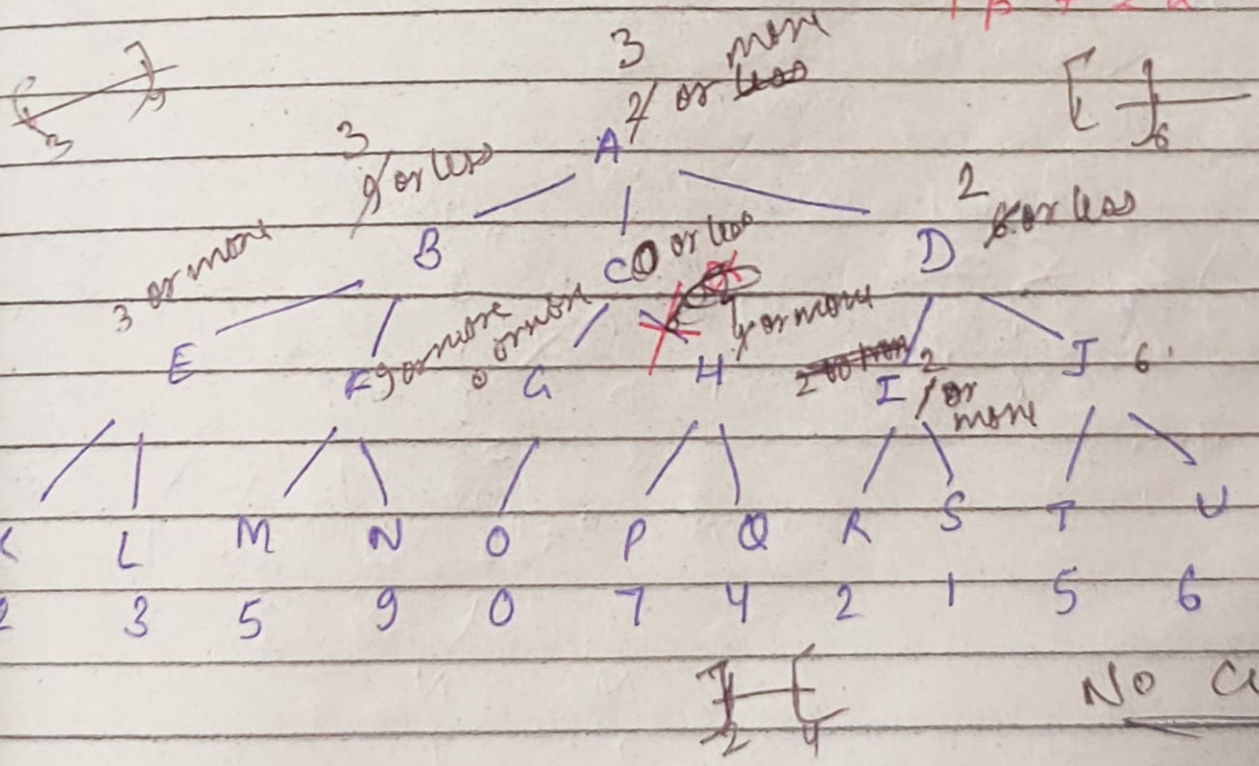
max Man
min
max





Man
min
max

1β + 2α cut off



Man
min
max

No cut off

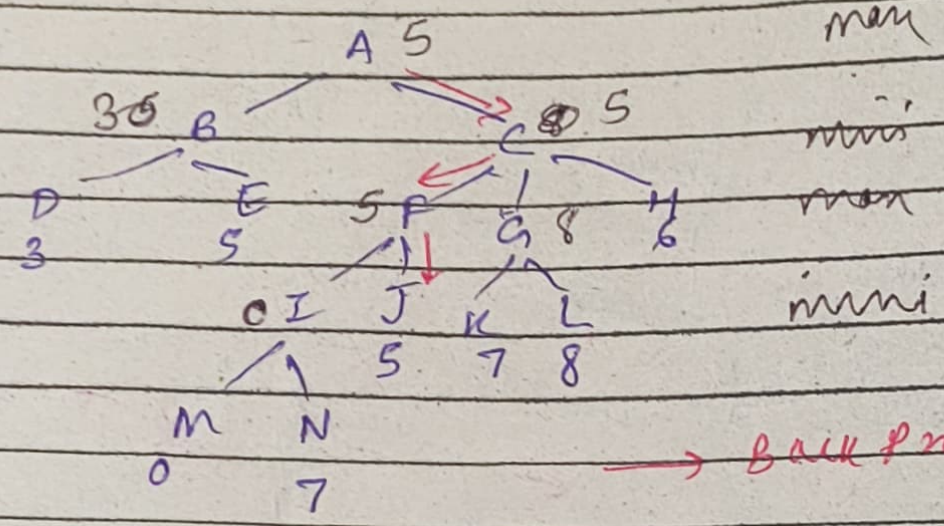
14.09.2023
Thursday

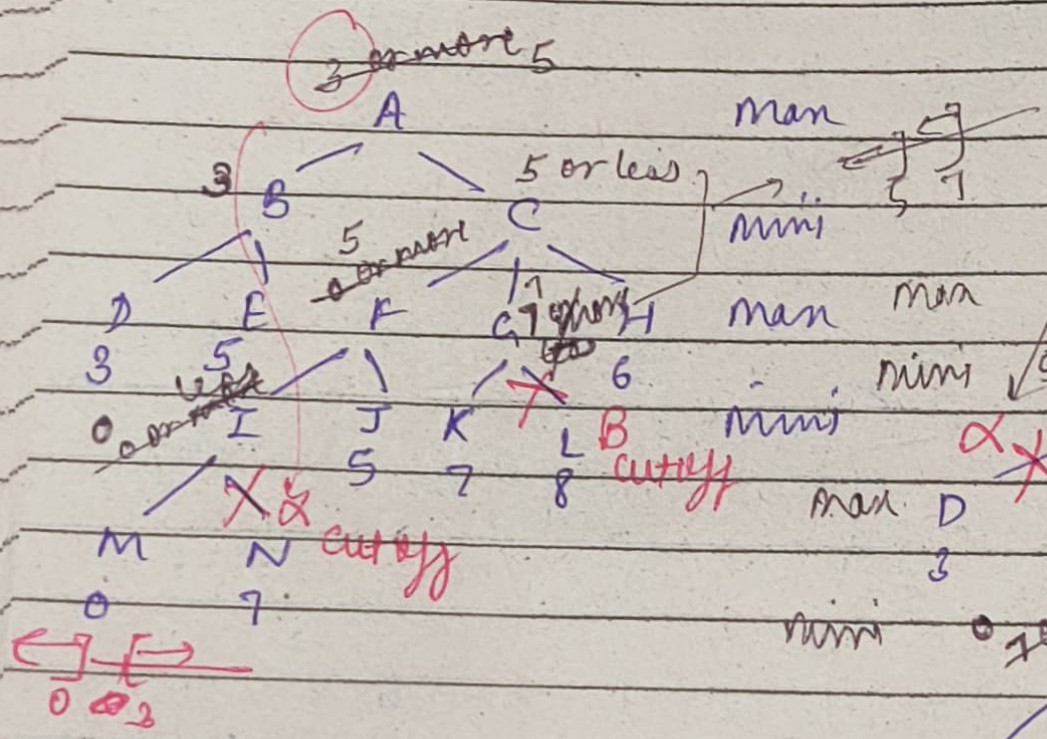
max. Node पर α

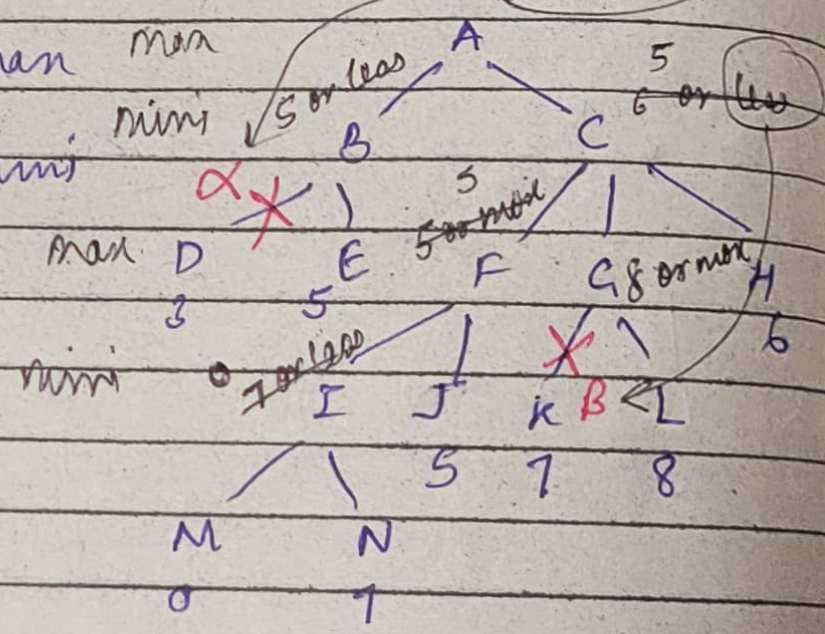
Date: Page:

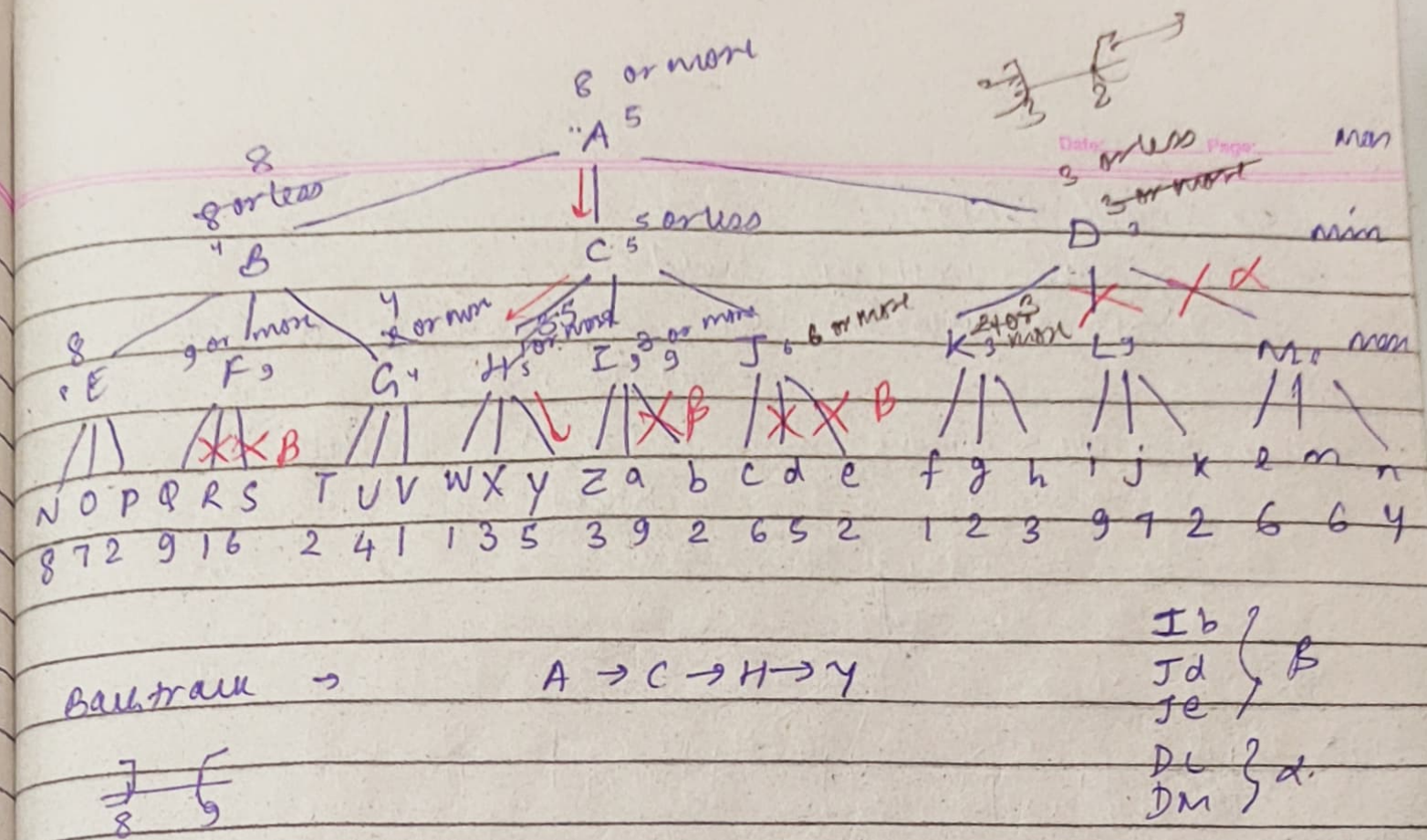
Q) Consider the following game tree (1) Using min max find. (2) Left to Right α-β cutoff.
Using Right to left α-β pruning (cutoff)

Sol:





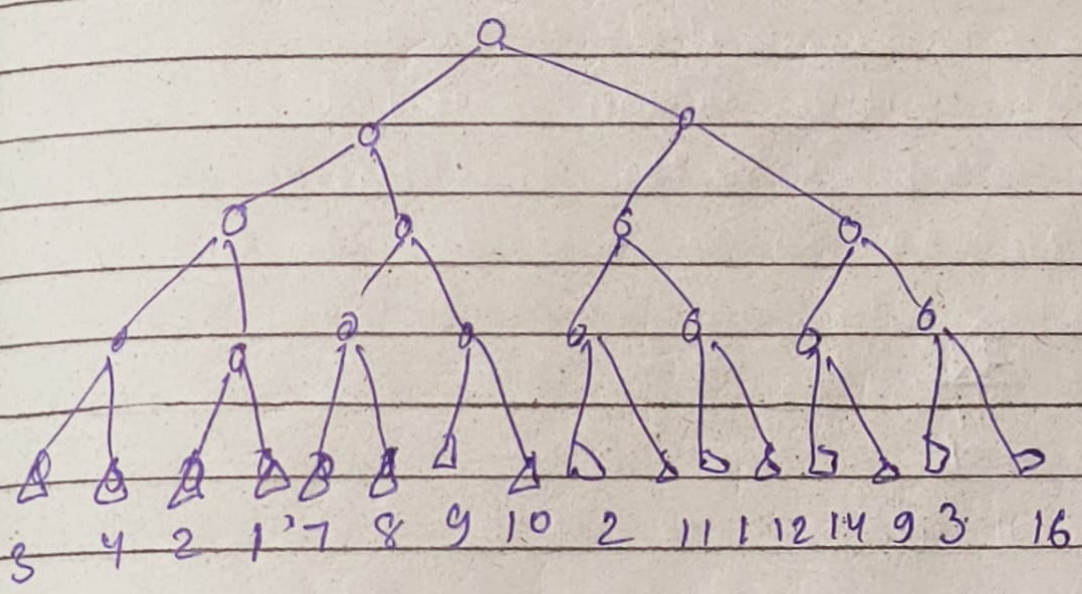




Backtrack → A → C → H → Y

Ib
Jd
Je } β

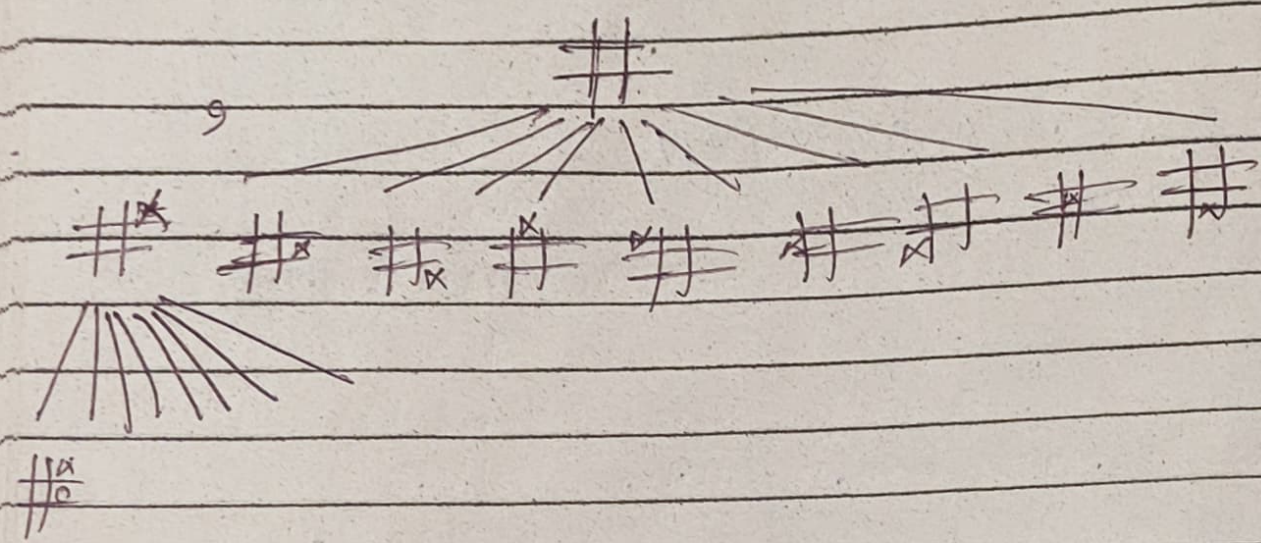
DL
DM } α

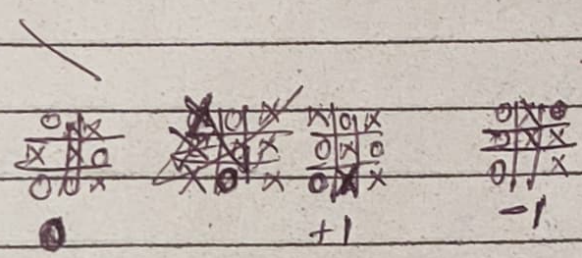


Date: Page:

Back Tracking Algo.

- Best move strategy used
  - Max will try to maximize its utility (Best move)
- Min will try to minimize its utility (Worst move).





No one wins — I wins (X wins) — II wins

Plausible Move Generator → All possible children are generated

Static Evaluation function (utility value) → Evaluates value of the path.

20.09.2023

Rich & Knights

MinMax :-

① Set FINAL-VALUE to be as minimum as possible.
② If limit of records has been reached then FINAL-VALUE = GOOD-VALUE of the current position.
③ Else do
   1. Generate the successor of the position
   2. Recursively call MINMAX again with the present position with depth incremented by UNITY.
   3. Evaluate the GOOD-VALUE. → minimizer है तो minimum
④ If GOOD-VALUE > FINAL-VALUE then update FINAL-VALUE
   अगर maximizer है तो maximum।

MIN-MAX :-

① ~~If the~~ Determine if the limit of search has been reached or if the level is minimizing level or if the level is maximizing level.
(a) If the limit of the search has been reached compute the static values of the current position relative to the appropriate player. Report the result. (Leaf Nodes at value पर ही रहेगी)
(b) If the level is minimizing level, use MINI-MAX on the children of current position. Report the minimum of result. (minimum value)
(c) If the level is maximizing level, use Mini-Max on the children position. Report the maximizing of result. (Maximum value)

Date: Page:

α-β Cut off :-

① Determine if the level is top level or if the limit of the search has been reached or if the level is minimizing level or if the level is maximizing level.

(a) If the level is top level, let $\alpha = -\infty$ & $\beta = +\infty$

(b) If the limit of the search has been reached, compute the static values of current position relative to the appropriate player. Report the result.

(c) If the level is MINIMIZING level.

(i) Until all children are examined with MINIMAX or $\alpha \geq \beta$

(A) Set β to the smaller of the given β-values & smallest value so far reported by ~~MINIMAX~~ MINIMAX working on the children.

(B) Use MINIMAX on the next ~~children~~ of the current α & β.

(ii) Report β.

(d) If the level is MAXIMIZING level,

(i) Until all children are examined with MINIMAX or $\alpha \geq \beta$

(A) Set α to the larger of the given α-values & biggest value so far reported by MINIMAX working on the children.

AI engineers → knowledge engineers - who create simulation.

How can we extend MIN-MAX to more than 2 players.

(B) Use MINIMAX on the next child of the current position & handling this new application of MINIMAX on the current α & β.

(ii) Report α.

Q) What is the reason, why game playing plays a major role in AI?

① The rules of the games are limited. Extensive knowledge of game specific.

② Many humans desist in developing of problems

Knowledge Engineers → goto → Domain experts (multiple) — Chess players, internet

(knowledge acquires)

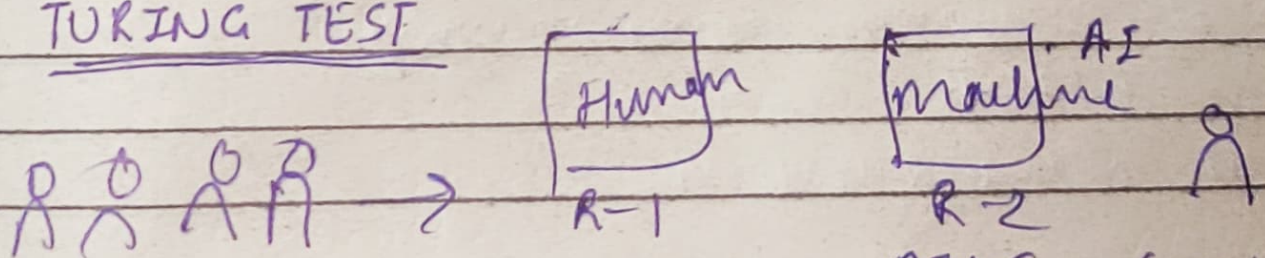
Production rules are created.

LHS → RHS

(many to many)

Expert System (only when machine can replace human)

TURING TEST



who has done go... giving prescri... on

Date: Page:

~~(5) Games provide a~~

(1) The rules of the game are limited. Hence extensive amt. of domain specific knowledge are needed.

(2) Many human experts exist to assist in the developing of the program. Hence the problem of finding the variety of human experts does not arrise.

(3) Games provide a structured task ~~when~~ where success of failure ~~of~~ can be measured with least effort.

(4) For the human expert, it is easy to explain the rationale ~~of~~ for a move unlike other domains → ~~reason~~

(5) Games picture real life ~~pictu~~ situation in a constricted ~~man~~ fashions. The logical reasoning ability of the human in a normal cond^n & under stress is clearly exhibited in game playing. Game playing permits one to simulate real life situations.

(IOT ~~solutn~~ devices use of AI)
(Medical + use of AI)

~~(6) Cor~~.

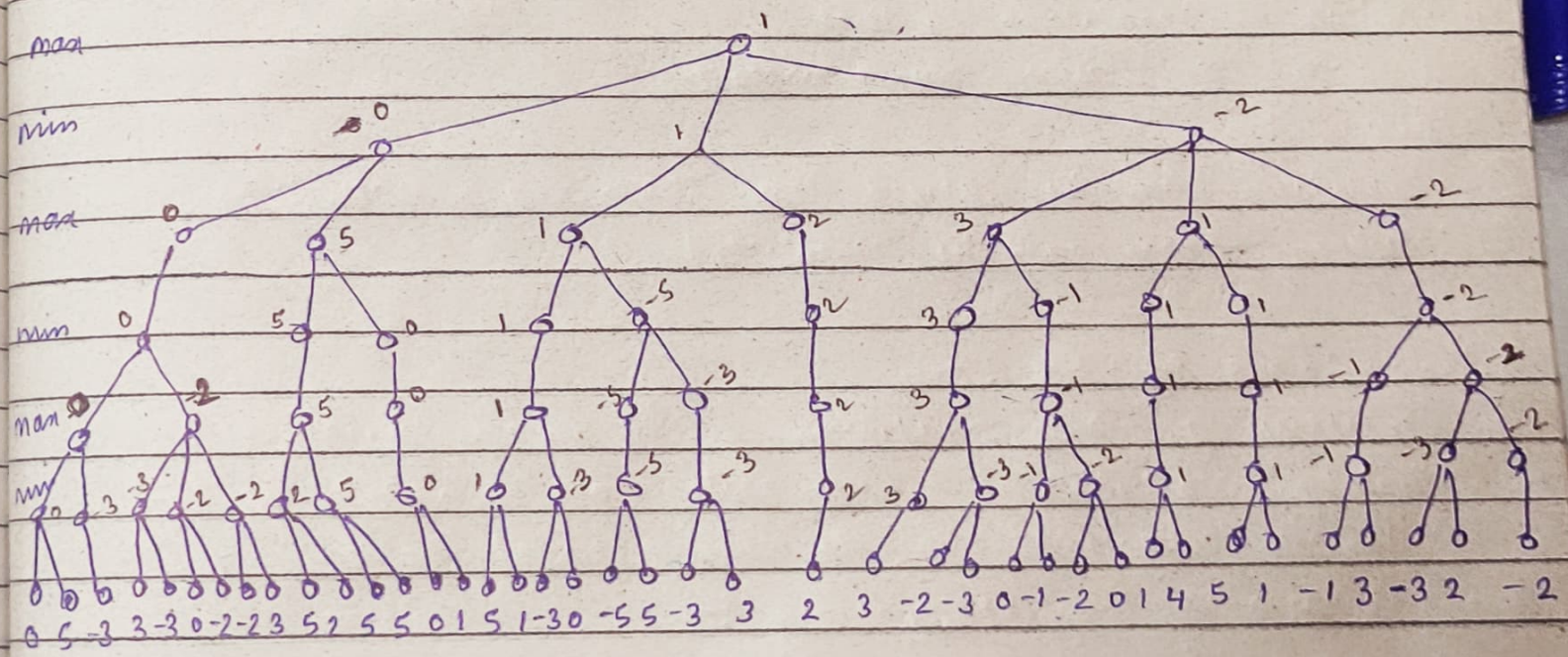
Q) Problems of computer game playing.

Sol: (1) Horizon Effect.
(2) Optimal move Question.

programs

① Horizon Effect :- Computer Chess playing have to be deal with the major problem known as Horizon effect. These 2 programs do look ahead search, if it is possible for program to postpone the danger by dragging the situation into the branch of the tree which the algorithm explores then the situation is out of the view of computer program or horizon.

② Optimal Move Question :- Second major defect is that these program expect the ~~comp~~ opponent to make most optimal move which can't be expected in real.



21.09.2023
Thursday

Date: Page:

```
   P A S
 x R B Q
 -------
   S B K W
 A S A A x
S E P B x x
-----------
S Q S K A W
```

K + A = A

K = 0

B + A + B = K → 0 or

2B + A = 0 or 10 or 20

Date: Page:

PRODUCTION SYSTEM: (PS)

(Search system according to AI)

PS in AI helps in structuring AI programs ~~in a way~~ that facilitates describing & performing the search process.

PS consists of

① Set of Rules ⇐ LHS → RHS (Forward Strategy Search)

② Knowledge Base {Database in AI}

③ Control Strategy {When more than 1 rule matches then ~~that is~~ which ~~to take~~}

④ Rule Applier.

Perform the first
Perform the best
Perform the most recent
↳ जितना Recent उतना updated.

Steps required to solve the problem –

① Reduce the problem so that it can be shown in a precise statement

② Problem can be solved by scanning a path thro space

③ Solving process can be modelled.

Advantages of PS :-

① Structuring AI problem (PS is an excellent tool)

② Highly modular (new rule can be added easily)

③ Rules are expressed in its own natural forms.

Characteristics of Control Strategy :-

① It should cause motion

② It should be systematic.

Classes of PS | Types of PS | Characteristics of PS :-

1. Monotonic P.S.
2. Non - Monotonic P.S.
3. Commutative P.S.
4. Partially Commutative P.S.

1. Monotonic PS (system which never prevents backtrack)

MPS is a PS in which ~~PS~~ application of a rule never prevents the latter application of another rule which could be applied at the time the first rule was selected.

8 Puzzle

2	1	6
4	5	8
3	7	

①

2	1	6
4	5	
3	7	8

②

③

2. Non Monotonic PS (prevents backtrack)

NPS is a PS in which upper one true.

Chess ~~Jump~~ more Backtrack etc.

4. Partially Commutative PS : (PCPS) :

PCPS is a PS with a property that if the application of a particular ~~system of~~ sequence of rule, transform state ~~X~~ into Y. then any ~~transform~~ permutation of those rules that are allowable also transform state x into state Y

$x \rightarrow y$

1 4 3 2
4 1 2 3

} all permutation give same results.

③ Commutative PS : (CPS)

Monotonic + PCPS

S:

①

②

③

A

①

②

③

C

①

B