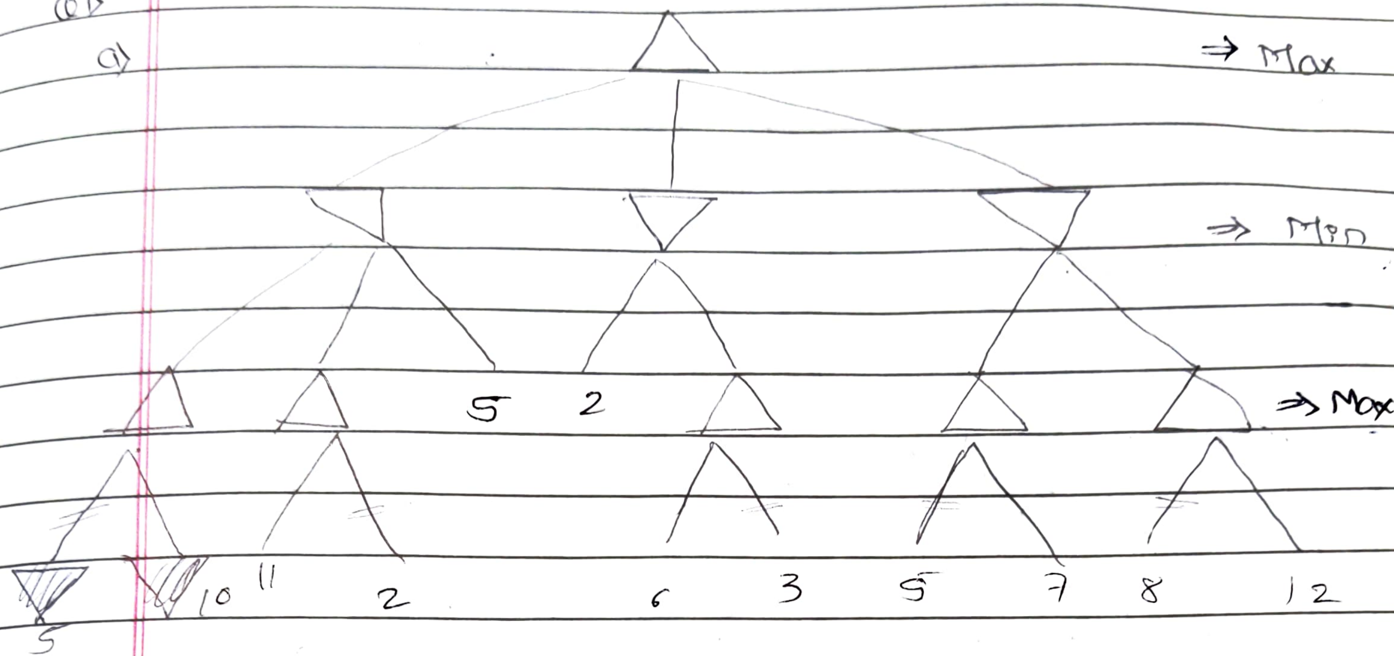
Assignment - 2

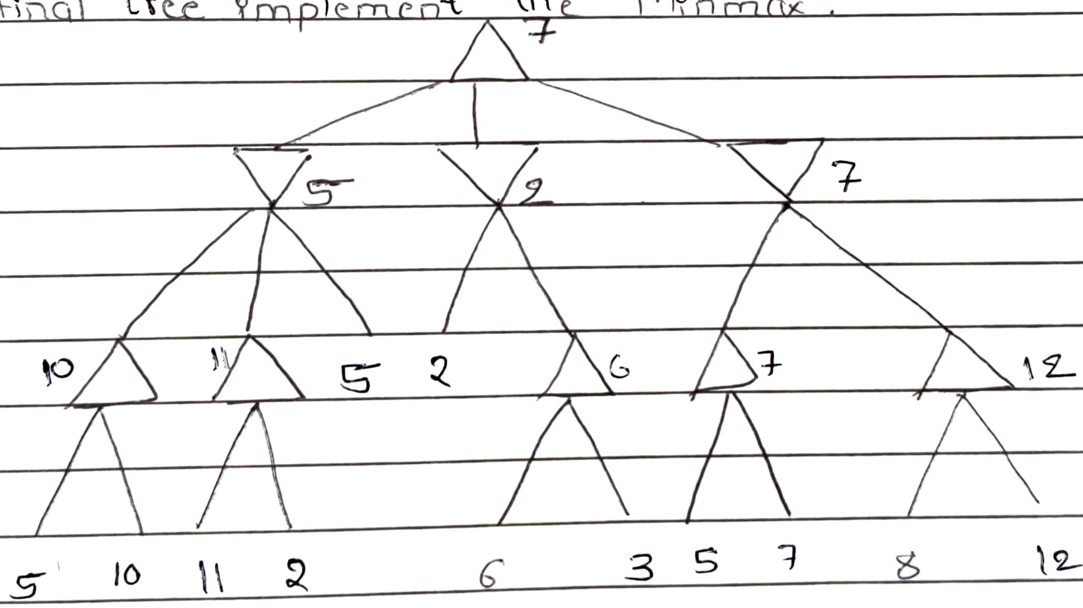
Q1)

a)

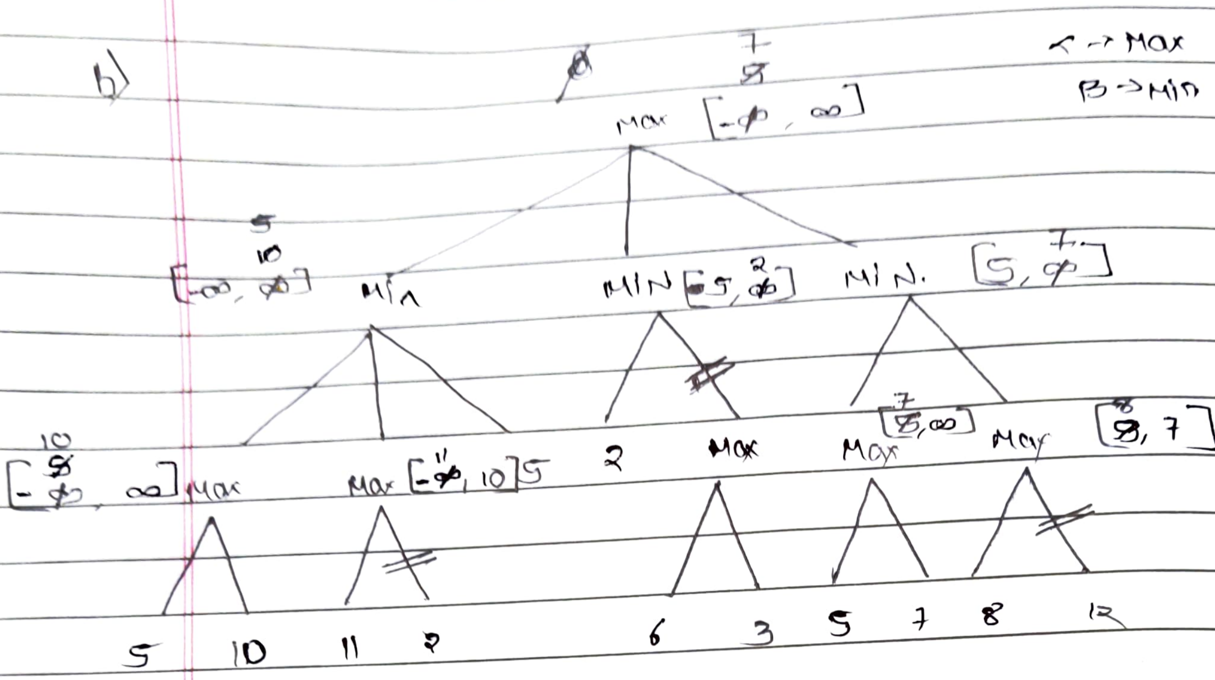


= denotes pruning

Final tree implement the Minmax.



b)



c) for α-β pruning,
we maximise α & minimise β
also if $\alpha \geq \beta$ we prune,
give $\alpha \leq 12$
$\beta \geq 2$

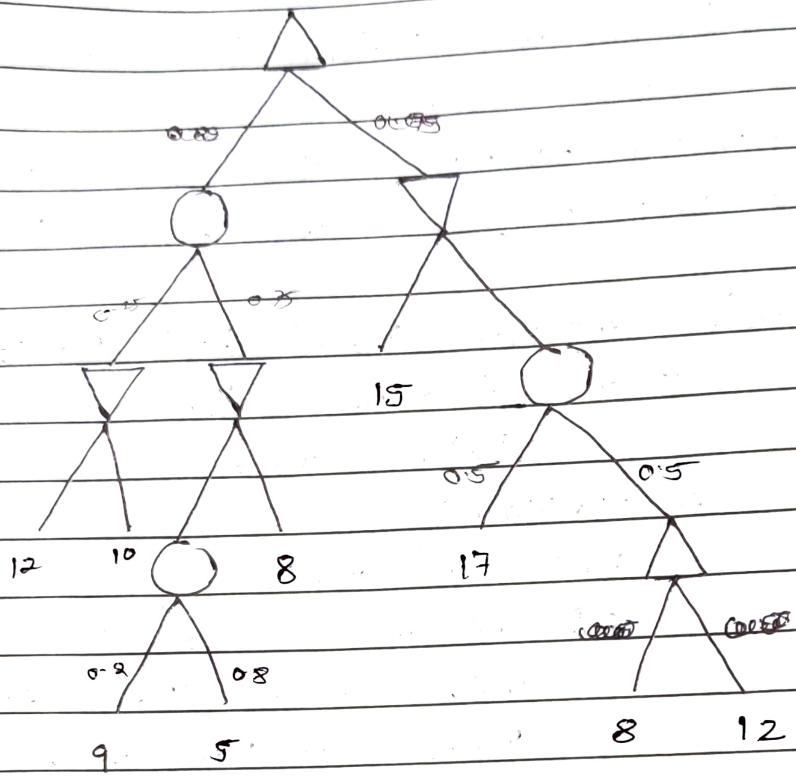
Q2)

⇒ DeepGreen will Return the best possible move from state S.

By using minmax function, we'll check for all possible ways & will return the best way.

```
function minmax (board, depth, MaxPlay):
    if current state is terminal state :
            Return value.
    if MaxPlay :
        Best = - infinity
        for each move in board:
            value = minmax (board, depth+1, false)
            Best = min (Best, value)
    Return Best.
   else :
        Best = + infinity
        for each move in board:
            value = minmax ( board, depth+1, true)
            Best = min ( Best, value)
    Return Best.
```

Q3)

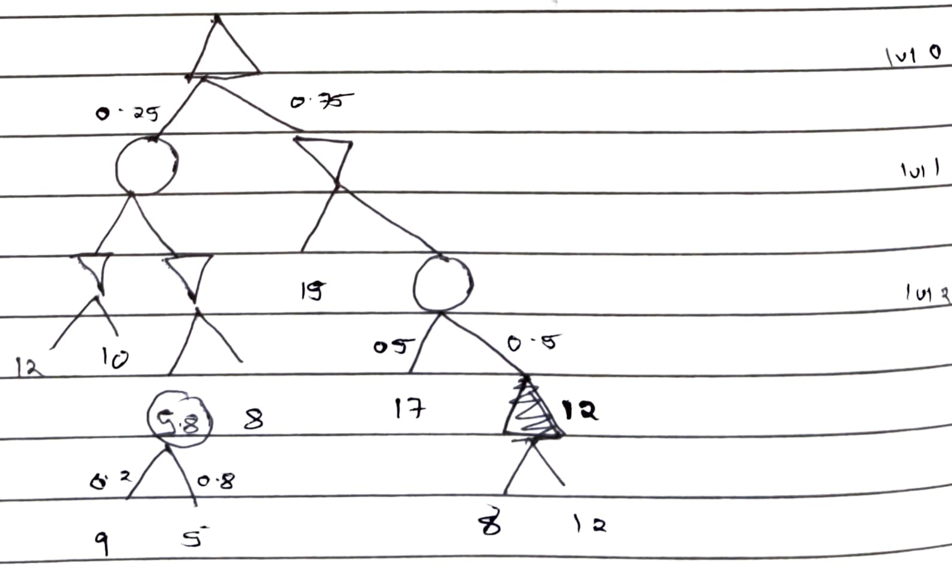


P) At level 3:

Value $= 0.2 \times 9 + 0.8 \times 5 = 1.8 + 4$
$= 5.8$ } for left node

&

$\max(8, 12) = 12$ for right node.



At level 2 :

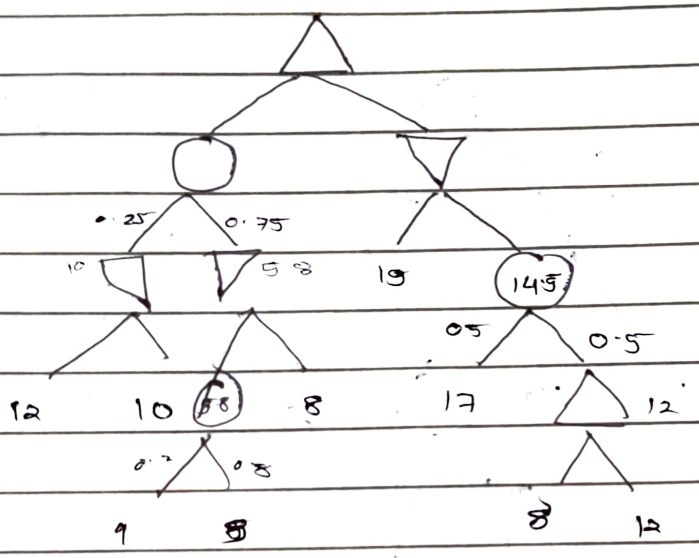
min (12, 10) = 10

min (5.8, 8) = 5.8

value = 0.5 × 17 + 0.5 × 12

= 8.5 + 6

= 14.5

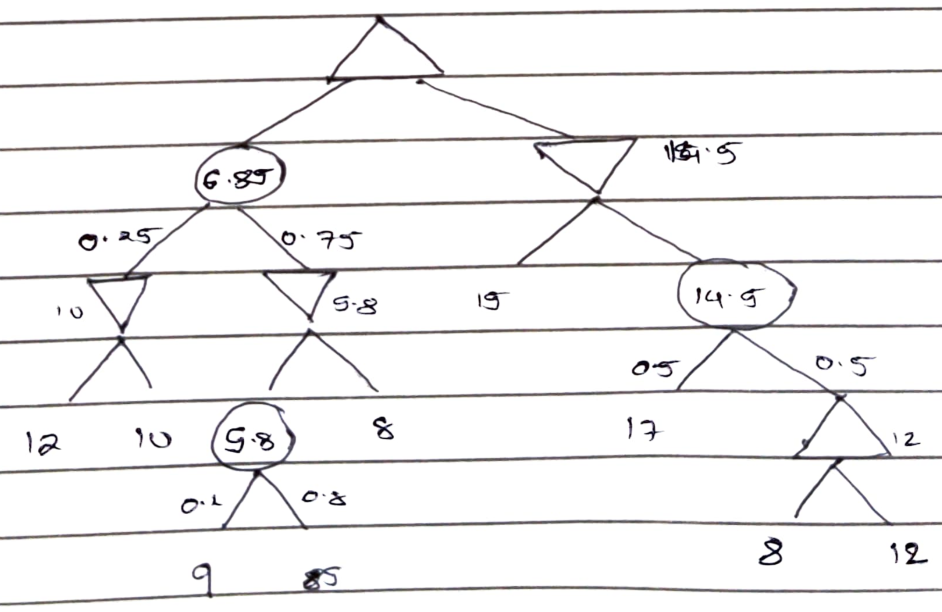


At level 1

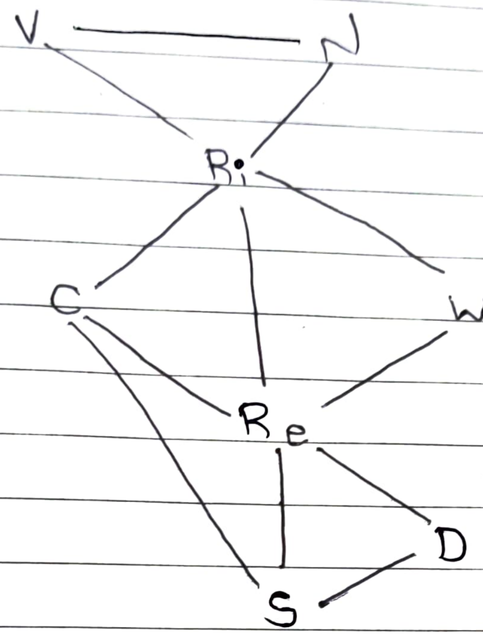
value = 0.25 × 10 + 0.75 × 5.8 = 2.5 + 4.35

6.85

max (19, 14.5) = ~~19~~ 14.5



Q4 Part a)



Part b)

Taking Ri with DH = 5

coloring red

Ri - Red

Re - blue (with DH = 4)

W - Green (with DH = 0 & MRV = 1)

C (MRV = 1 & DH = 2) : Green

S : Red (MRV = 1 & DH = 2)

D : Green (DH = 1)

N : DH = 1, MRV = 2, Blue

V : Green, (DH = 1, MRV = 1)

Part c)

V	N	Ri	Re	W	D	S	C	I
G	B	R	GB	GB	RGB	RGB	GB	RGB

color for N assigning Blue to N

~~V → N~~

~~Ri → N~~

~~N → V~~

~~Ri → V~~

V : G

N : B

Ri : R

color for S assigning B to S

Ri	R	W	D	S	C	I
R	GB	~~G~~ B	RG~~B~~	B	~~GB~~	RGB

Re → S

D → S

C → S

Ri → Re

W → Re

C → Re

S → Re

D → Re

Re → D

S → D

Re → C

S → C

Ri → C

Ri → W

Re → W

As there is No value in C. Hence no blue for S

Assign R to S

Ri	Re	W	D	S	C	I
R	GB	GB	~~R~~GB	R	GB	RGB

~~Re → S~~

~~D → S~~

~~C → S~~

~~Re → D~~

~~S → D~~

S will have Red, Ri will have red,
∴ V: Green & N: Blue.

Ri	Re	W	D	S	C	I
Red	G	~~G~~B	~~G~~B	R	~~G~~B	RGB

Now, for Green for Re

~~Ri → Re~~

~~W → Re~~

~~C → Re~~

~~B → Re~~

~~D → Re~~

~~Ri → W~~

~~Re → W~~

~~Re → C~~

~~Ri → C~~

~~S → C~~

~~Re → D~~

~~S → D~~

Assign I with any RGB Taking R with I

V → Green

N → Blue

Ri → Red

S → Red

W → Blue

Re → Green

D → Blue

S → Red

C: Blue

I: Red.

Part D)

Remaining are Ri & Re

V → N   W   D – S – C

Ri — Re

Making N as root

N — V

RGB   RGB

consider N is R & G or B to V

N : R

V : G

W can be R

Making C as Node

C → S → D

RGB   RGB   RGB

assign R to C

C → S → D

R   G   B

Re & V

N — Ri – C – S – D

W

Ri → N   W   C → S → D

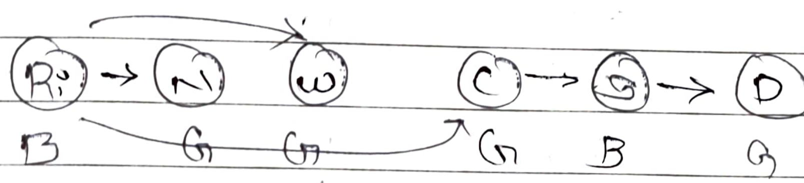
GB   RGB   RGB   RGB   RGB   RGB

V is Red , Re is Red

~~Red~~

V	N	Ri	Re	W	D	S	C
R	GB	GB	R	~~R~~GB	~~R~~GB	~~R~~GB	~~R~~GB

Ri → Re　　S → C　　Ri → N
D → Re　　Re → C
C → Re　　Ri → C
S → Re　　D → S
W → Re　　Re → S
S → D　　C → S
Re → D　　Re → W



Part ↄ

V : Red
N : Green
Ri : Blue
Re : Red
W : Green
D : Green
S : Blue
C : Green

Page No.
Date / /

Q5)

a) Give the above information, does KB entail S1?
Truth table for KB & S1 is:

KB	S1
T	T
F	T
T	T
F	T
F	F
F	F
T	T
F	F

In given Truth table it can be seen that KB entails S1 means if KB is true S1 is also True.
Hence, KB entail S1 is also True.

b)

(Not) KB	Not S1
F	F
T	F
F	F
T	F
T	T
T	T
F	F
T	T

(Not) KB does not entail S1, i.e when (Not) KB is true (Not)S1 is not true.
Hence (Not KB) does not entail (Not) S1 i.e false.

Q6)

→ First case:

A is true, B is false, C is true, d is true.

Second case:

A → false, B → false, C → true, D → false.

	A	B	C	D	KB
fc	T	f	T	T	f
sc	F	F	T	F	f
Remaining cases					T

For getting in CNF form, we'll negate all false rows & get conjuction between them.

Result:

$\neg (A \land \neg B \land C \land D) \land \neg (\neg A \land \neg B \land C \land \neg D)$

~~CNF: (¬A ∧ B ∧ ¬C ∧ ¬D) ∧ (A ∧ B ∧ ¬C ∧ D)~~

CNF: $(\neg A \lor B \lor \neg C \lor \neg D) \land (A \lor B \lor \neg C \lor D)$

Q7

⇒ KB ⇒ A → C

B ⇔ C

D ⇒ A

B and E ⇒ G

B ⇒ F

E

D

i) Forward chaining

MP to D ⇒ A, D add A to KB

MP to A ⇒ C, A add C to KB

MP to C ⇒ B, C add B to KB

MP to B & E ⇒ G, B, E add G to KB

∴ KB ⊭ G.

Explored list is [D, A, C, B, E, G]

ii) Backward chaining

let goal state be 'GT'

GT	G	B & E ⇒ G
GT	B E G	C ⇒ B

GT

C
B
E
G

$A \Rightarrow C$

GT

A
C
B
E
G

$D \Rightarrow A$

GT

D
A
C
B
E
G

D is True.

Hence,

MP, $D \Rightarrow A$, i.e. A = True

$A \Rightarrow C$, C = True

$C \Rightarrow B$, B = True

$B \& E \Rightarrow G$, G = True

$\therefore KB \neq G$ where KB entails G

iii) <u>Resolution</u>

$(A \rightarrow C) \cap (B \Rightarrow C) \cap (D \Rightarrow A) \cap [(B \cap E) \rightarrow G] \cap$
$(B \Rightarrow F) \cap E \cap D \cap \neg G$

Converting to CNF, eliminate '$\Rightarrow$'

$X \Rightarrow Y = \neg X \cup Y$

$(\neg A \cup C) \cap (\neg B \cup C) \cap (\neg D \cup A) \cap [\neg(B \cap E) \cup G] \cap$
$(\neg B \cup F) \cap E \cap D \cap \neg G$

By De-morgan's law;

$(\neg A \cup C) \cap (\neg B \cup C) \cap (\neg C \cup B) \cap (\neg D \cup A) \cap$
$(\neg B \cup \neg E \cup G) \cap (\neg B \cup F) \cap E \cap D \cap \neg G$

$\neg \cancel{G} \qquad \neg B \cup \neg E \cup \cancel{G}$

$\neg B \cup \neg E$

$\neg B \cup \neg E \qquad \neg C \cup B$

$\neg E \cup \neg C$

$\neg E \cup \cancel{\neg C} \qquad \neg A \cup \cancel{C}$

$\neg E \cup \neg A$.

$\neg E \cup \neg A \qquad E$

$\neg A$

$\neg \cancel{A} \qquad \neg D \cup \cancel{A}$

$\neg D$

$$\frac{\neg D \qquad D}{\square}$$

$\square \rightarrow$ Empty clause

So K.B $\neq$ G

Q8)

→ Part a)

Contract:

If it Rains on Monday, John must give check of \$100 to Mary.

If john gives Mary check of \$ 100, Mary must mow the ~~lawn~~ lawn. on ~~the~~ wednesday.

→ Part b)

→ It didn't Rain on Monday.

→ John gave check of \$100 to Mary on Tuesday.

→ Mary mowed the lawn on wednesday.

→ Part c) Defining symbols.

R(M)	Rain on Monday
c(x, y, z)	x will give check to y on day z
M(y, z)	y will mow the lawn on day z

Converting to Propositional logic

⇒ R(Monday) → c(John, Mary, Tuesday)

c(John, Mary, Tuesday) → M(Mary, wednesday)

But it didn't Rained so acc to contract;

¬R(Monday)

c(John, Mary, Tuesday)

M(Mary, wednesday).

Date / /

Part d)

No, contract wasn't violated, because it says given check to Mary if it rains. Question didn't mention that does not give check if it does not Rain.