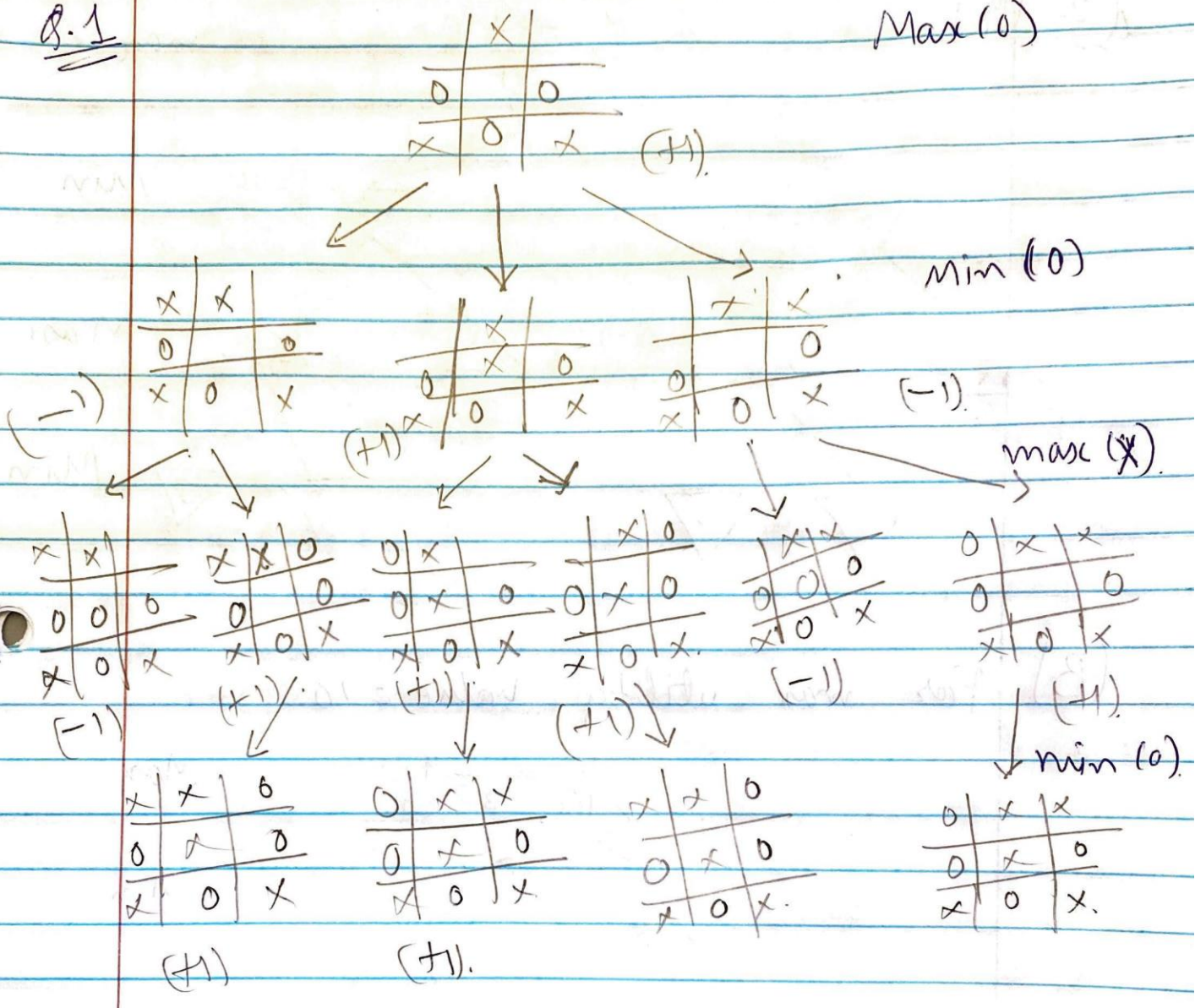
[redacted]

Assignment - 2.

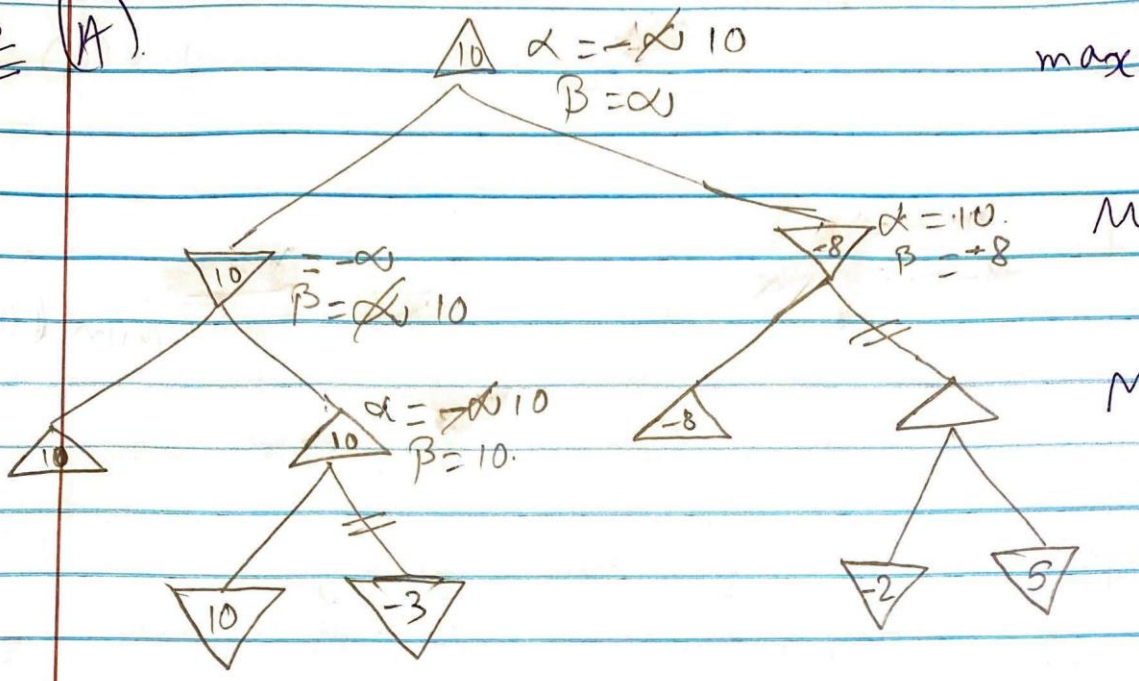
Q. 1



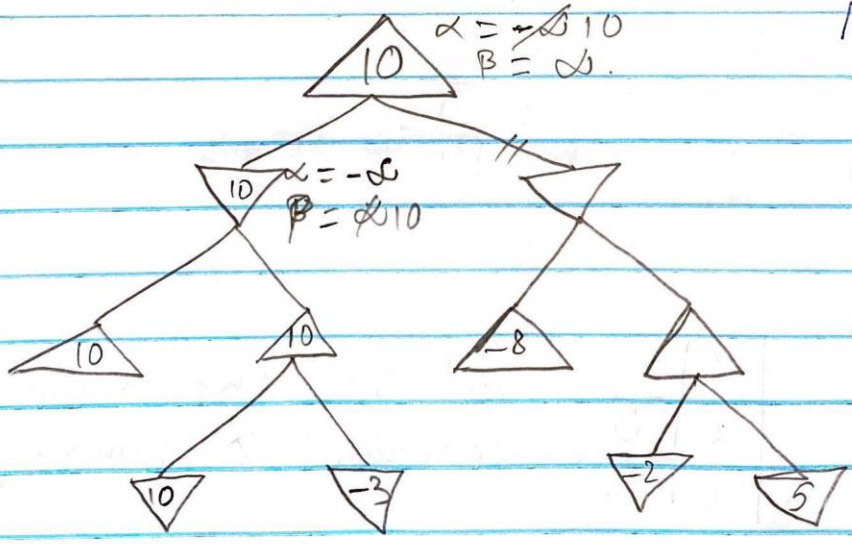
	X	
O	X	O
X	O	X

will be the ~~mode~~ move which generates X winning.

Q2 (A).



(B) For max. utility value = 10



We got the utility value as 10.
So, Now it will stop expanding further.

Q3 Supper Computer - Deepgreen is two player deterministic game of perfect information. and it don't use minmax algorithm.

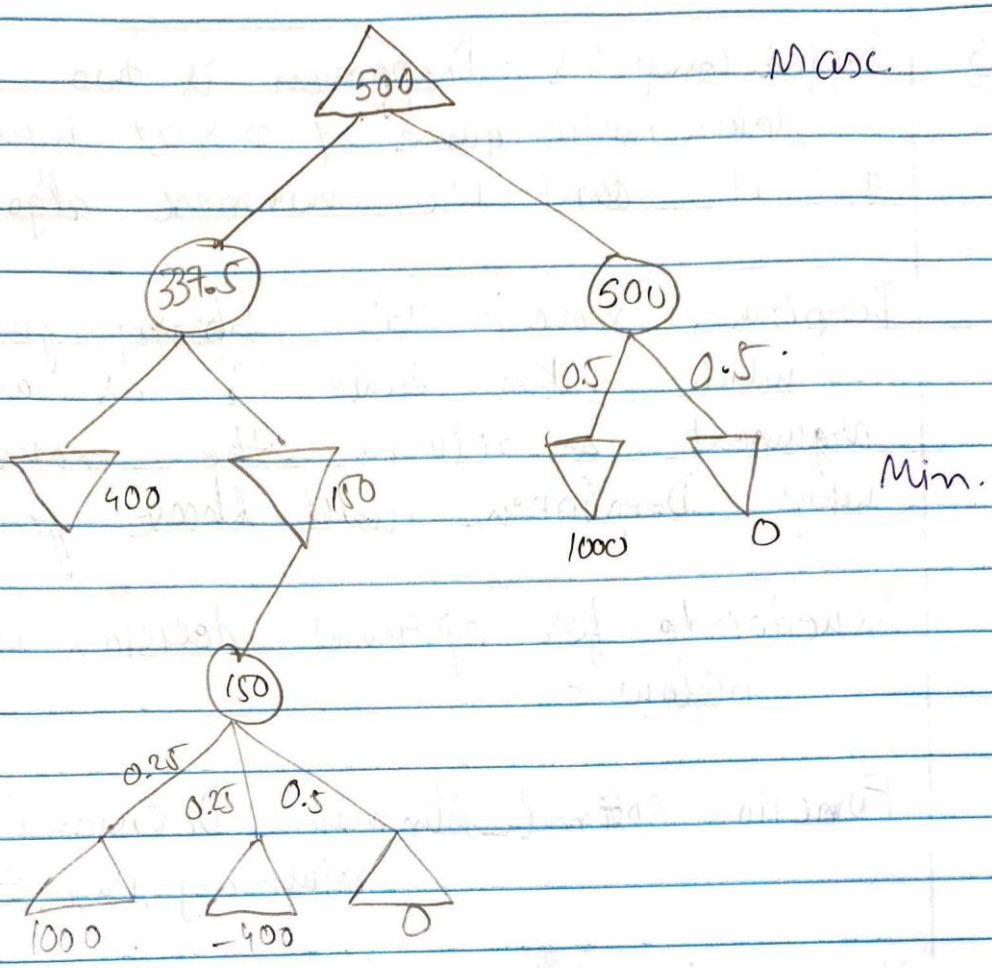
Deepgreen move :- It's Library function which takes State "S" as an argument & returns the move which Deepgreen will choose for "S".

Pseudocode for optimal decision is given below :-

Function Optimal_Minmax_Decision (State).
return arg Max ∈ Actions.

Min-value (Result (State, a))
function Max value (State) returns utility value.
if Terminal-test (State) then return utility (State) v ← -∞
for each a in Action (State) do.
v ← Max (v, min_value (Result (s, a)) return v.
function min value (State) returns a utility value
if Terminal_test (State) then return utility (State v ← ∞
for each a in Action (State) do.
v ← Min (v, max-value (Result (s, a))
if v ∈ Utility (Deepgreen move (state))
then return Max (v, Deepgreen move (state))
else return v.

Q.4

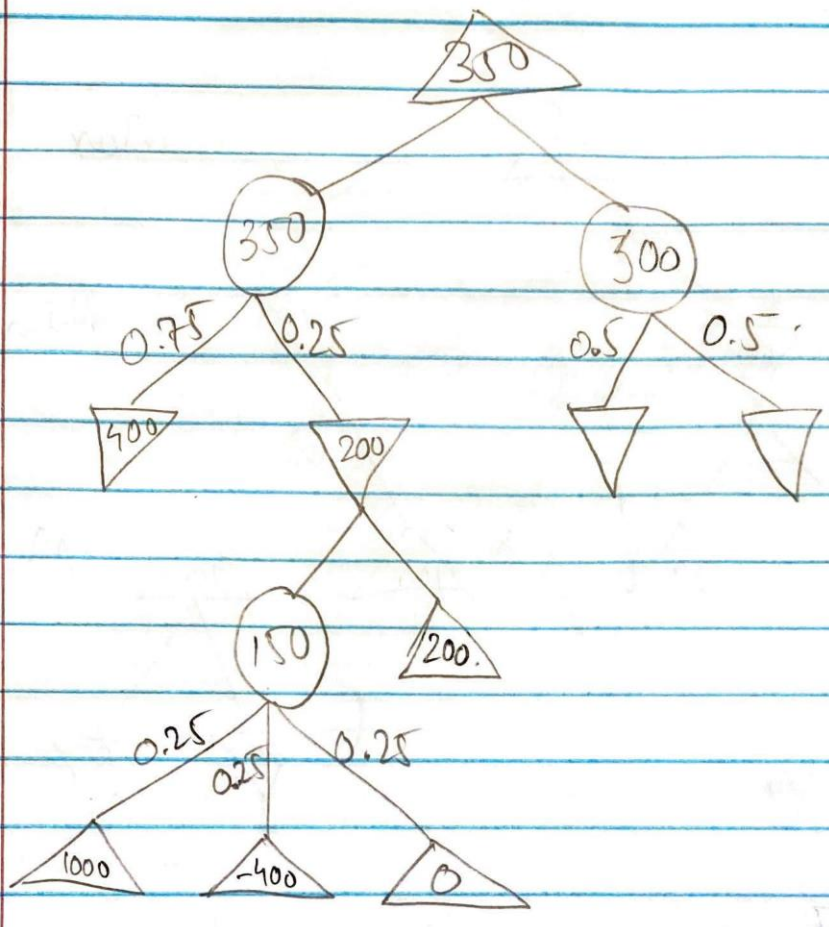


$0.25(1000) + 0.25(-400) + 0.5(0) = 150$

$400(0.75) + 0.25(150) = 337.5$

$1000(0.5) + 0.5(0) = 500.$

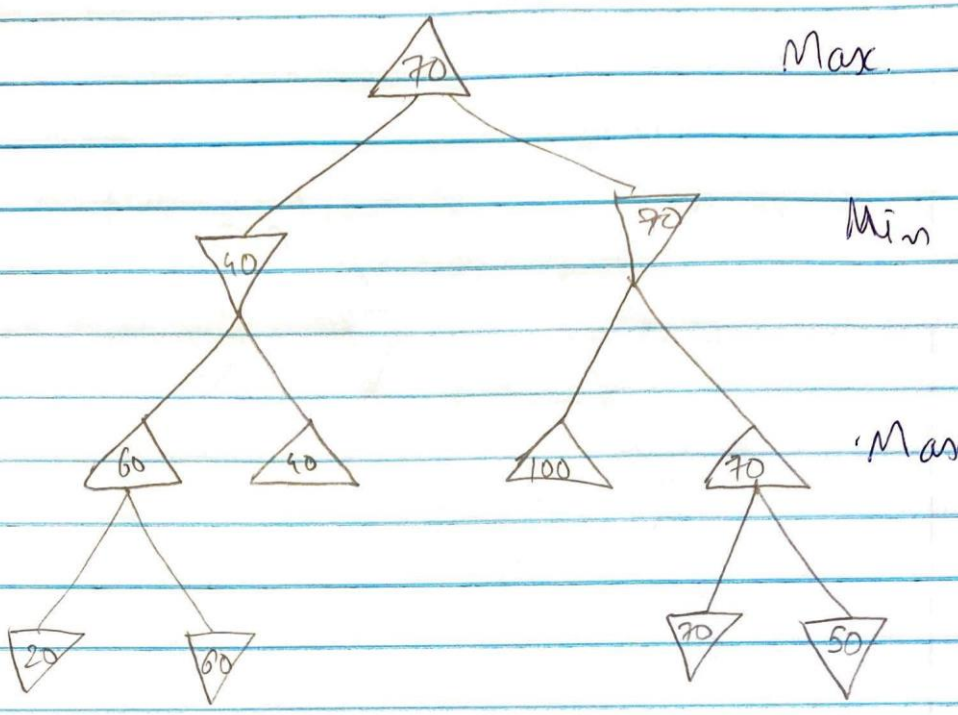
Max. value $\Rightarrow$ 500



$0.75(400) + 0.25(200) = 350$

Min. value $\Rightarrow 350$.

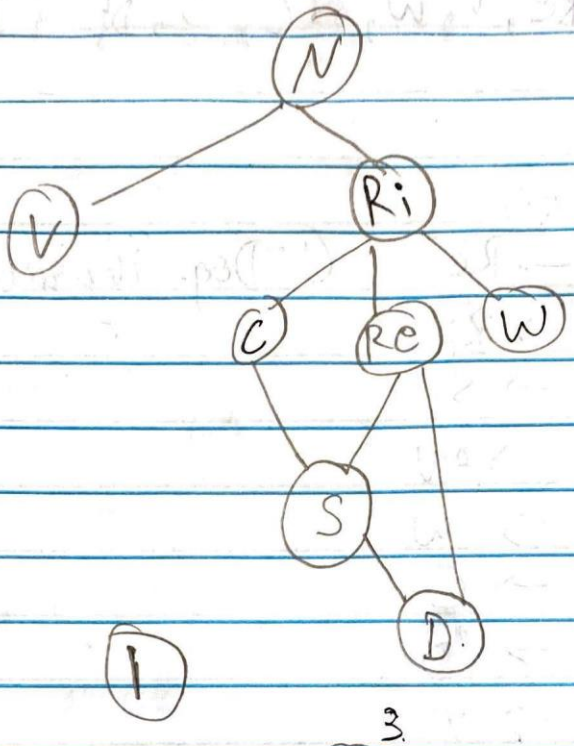
Q.5



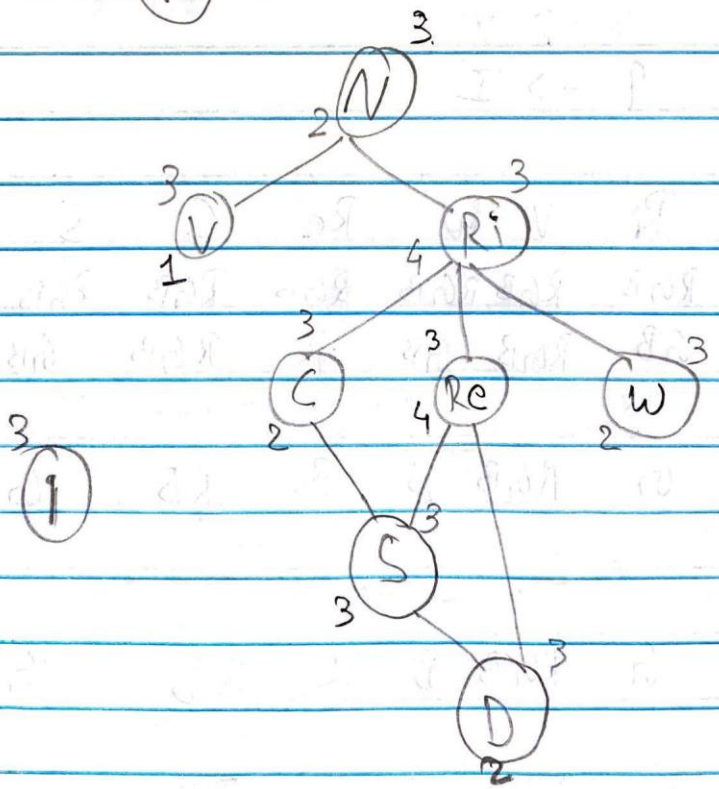
70 is max player value.
100 will be the value of max player if, min player doesnot play optimal.

Q.6

(A).



(B).



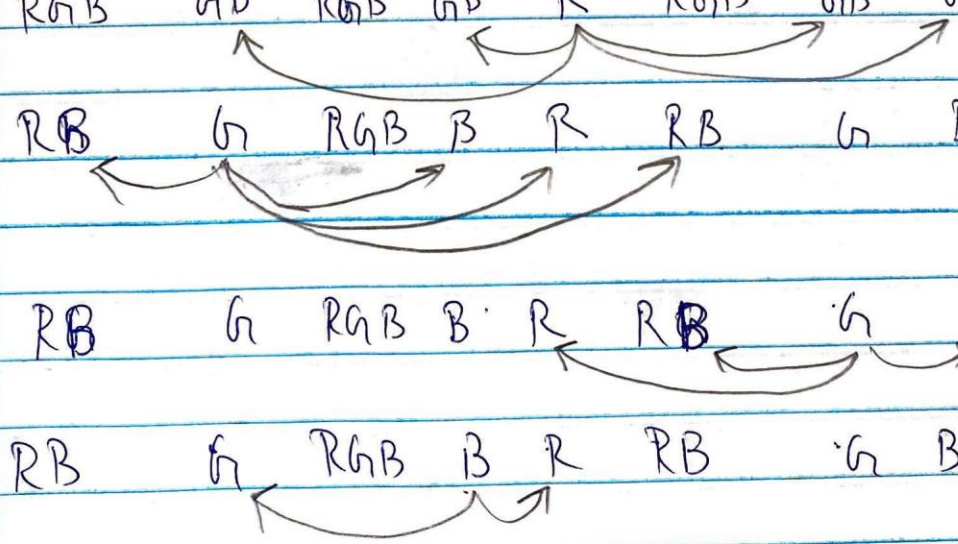
{N, Ri, Rc, V, W, C, S, D} {I}

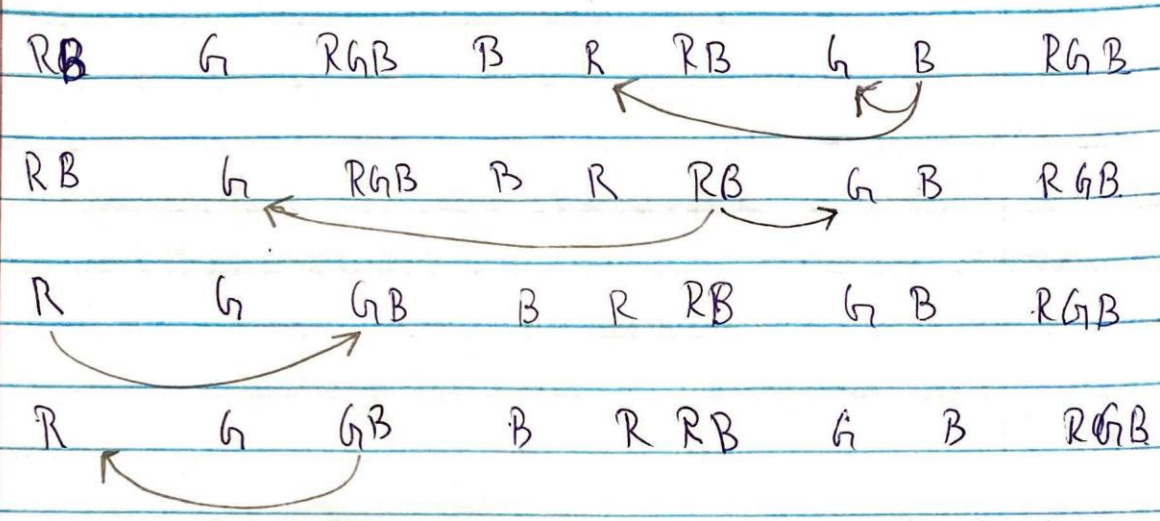
Steps :-

1 → Rc (∵ Deg. Heuristic is 4)
2 → Ri
3 → S
4 → N
5 → W
6 → C
7 → D
8 → V
9 → I

(C)

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





(D) Yes, it can be used to simplify process

(E)
N - R
Ri - G
V - G/B
W - B
Re - R
C - B/G
S - G
D - B
I - R/G/B