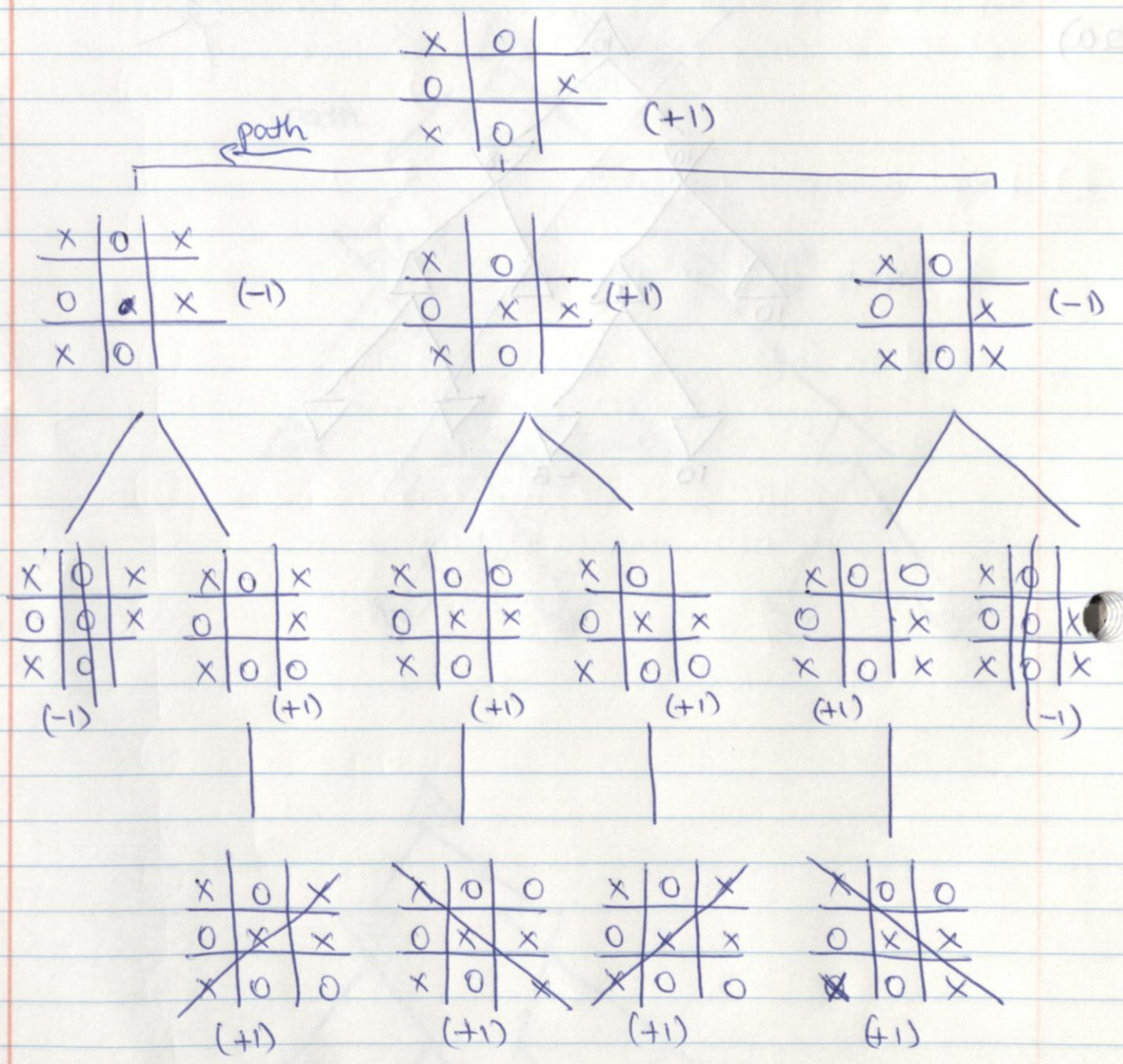
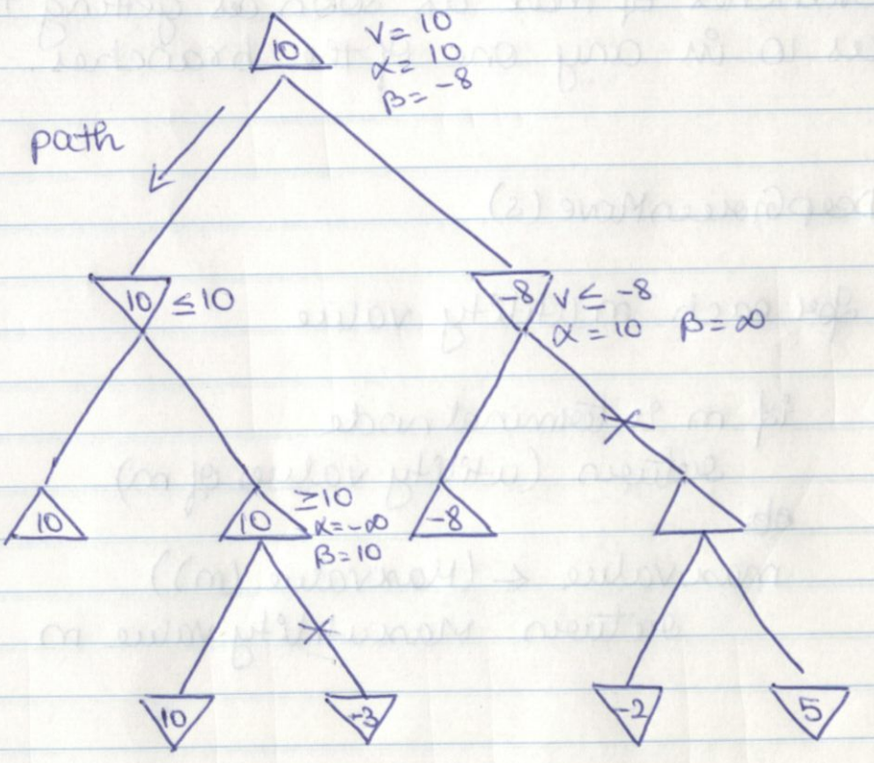
Aarchitha Haarinath
1001657433

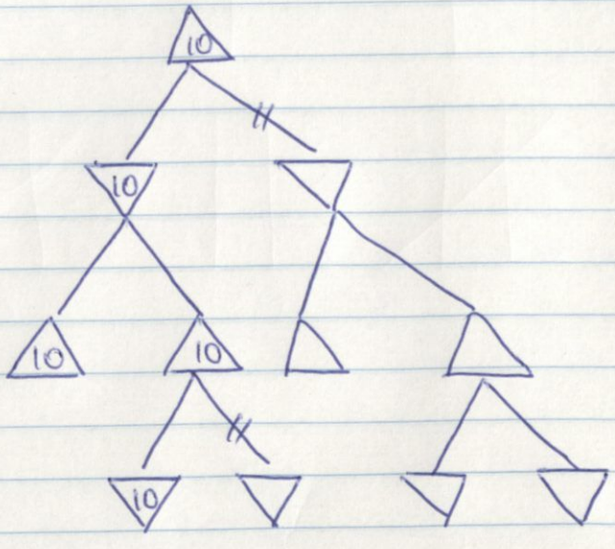
1)



2 a)



b)



Since they have given that maximum utility value as 10, we can stop traversing through the other

2)b branches of max as soon as getting the utility value as 10 in any one of the branches.

6) DeepGreenMove(s)

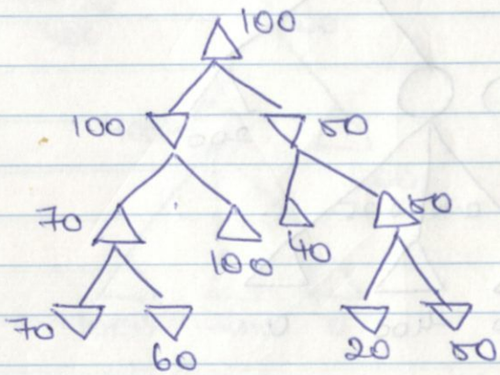
for each m utility value

if m is terminal node
return (utility value of m)

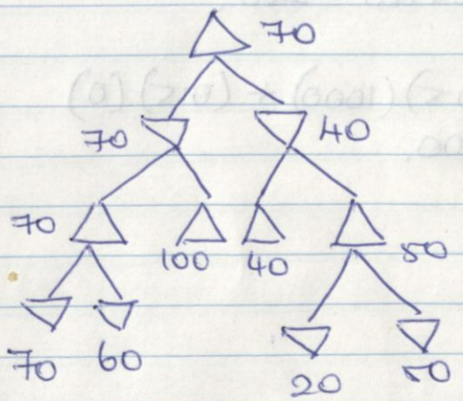
max value ← (Maxvalue(m))
return Maxutility value m

3)

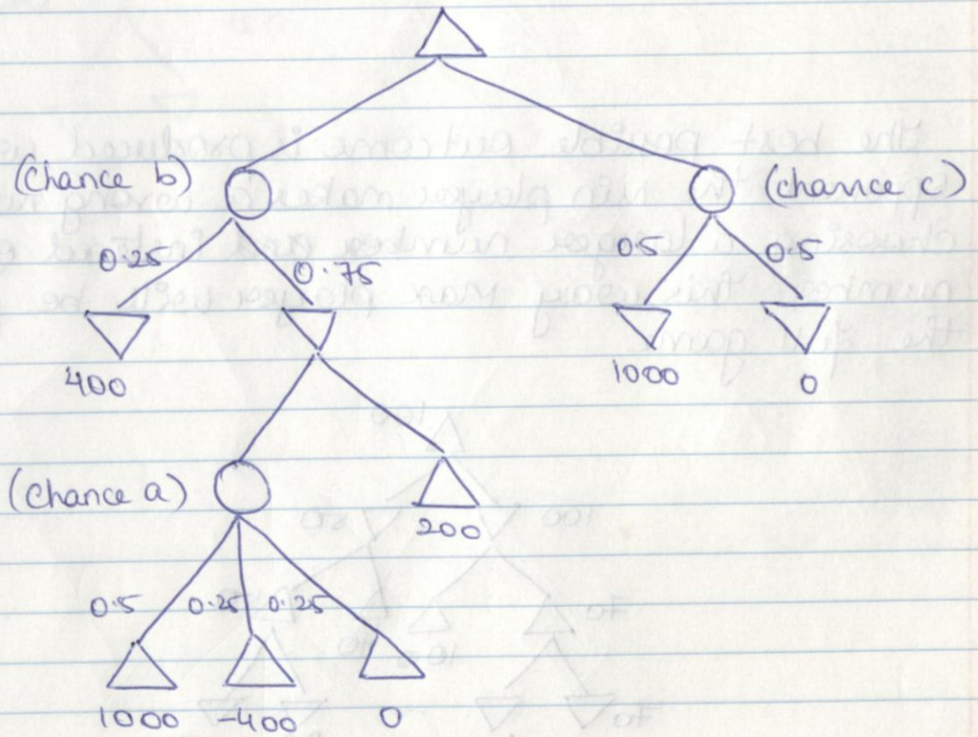
i) The best possible outcome is produced when the opponent the min player makes a wrong move by choosing a largeer number and instead of small number. This way Max player will be playing the full game.



ii) The max player loses or gets the worst outcome when the min player knows the strategy to play (i.e) choose a min value all the way till the end.



4)



chance a $= (0.5)(1000) + (-400)(0.25) + (0)(0.25)$

$= 400$

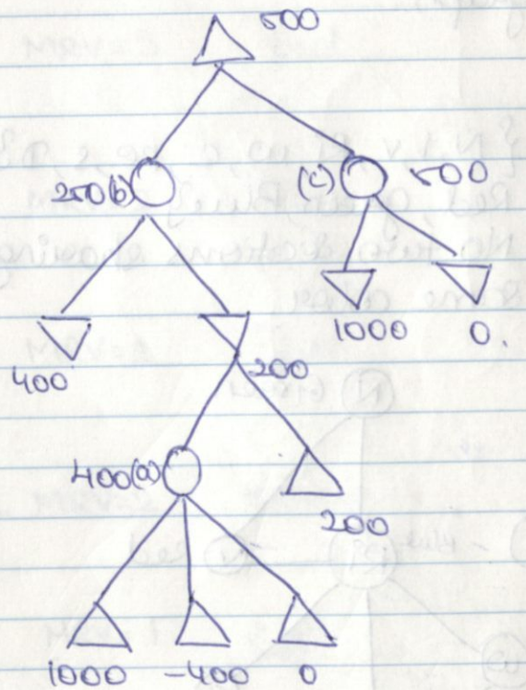
chance b $= (0.25)(400) + (0.75)(200)$

$= 100 + 150 = 250.$

chance c $= (0.5)(1000) + (0.5)(0)$

$= 500.$

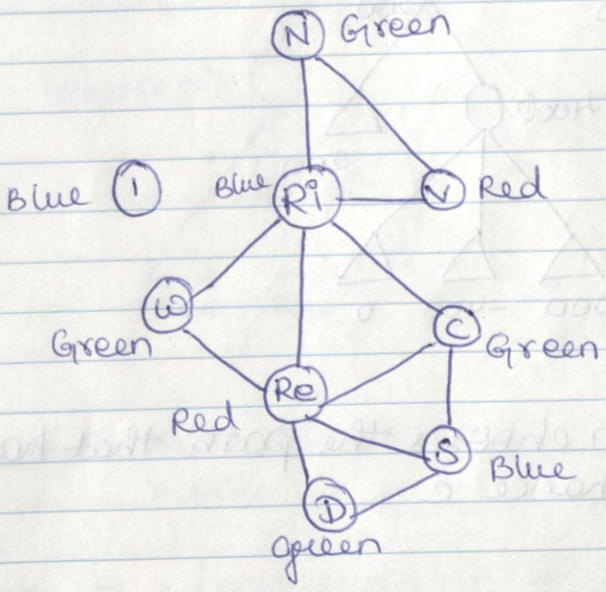
Solution:



The algorithm chooses the path that has the value of 500 (ie) chance c

5) constraint graph.

a) variables = {N, I, V, Ri, W, C, Re, S, D}
Domain = {Red, green, Blue}
Constraint = No two sections sharing a border should have same color.

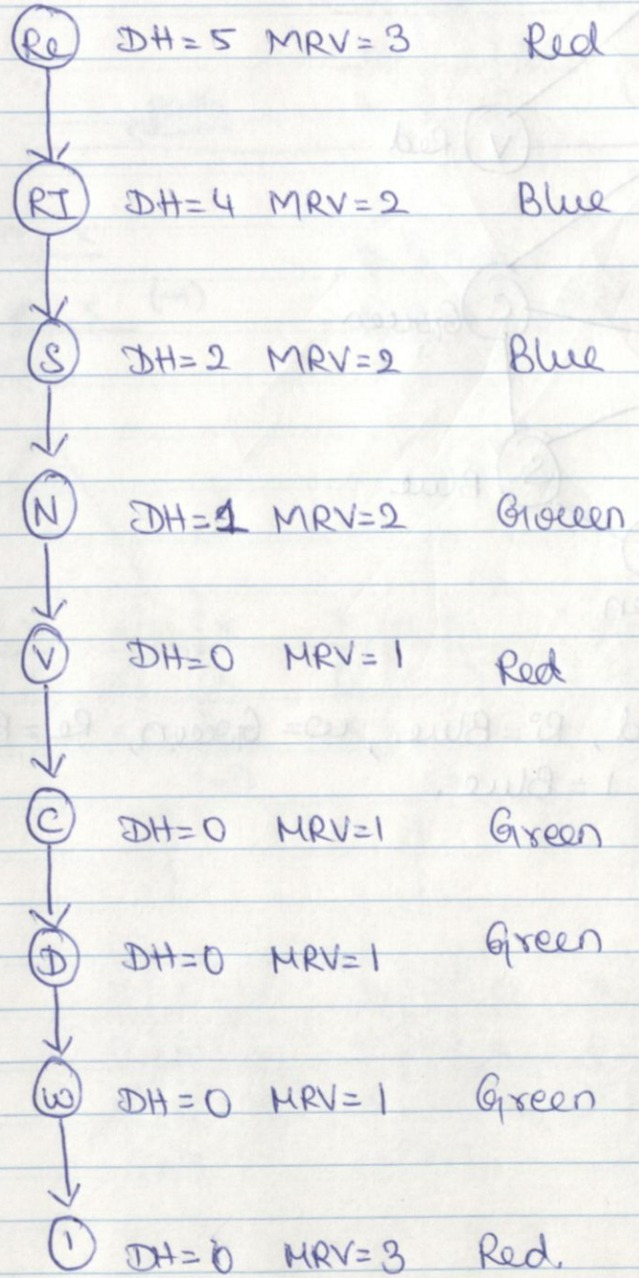


The colors assigned to the variables are

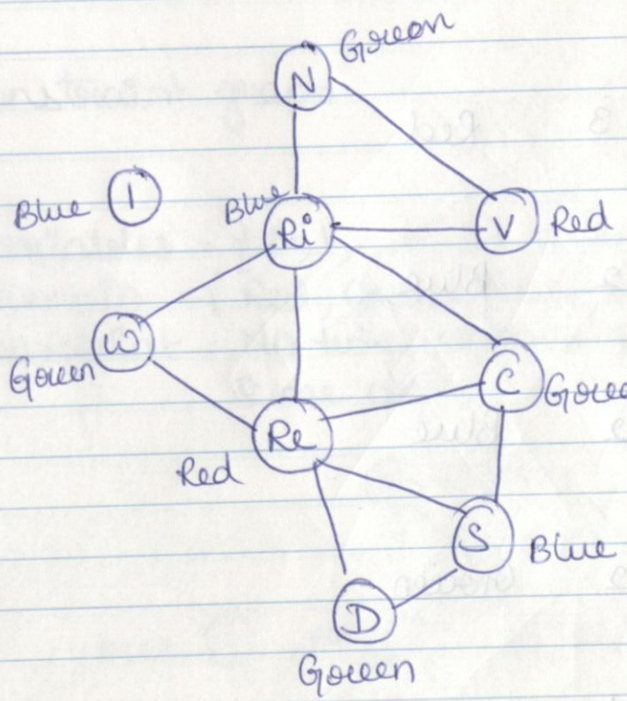
N = Green, I = Blue, Ri = Blue, V = Red, Re = Red, W = Green, C = Green, S = Blue, D = green.

yes we can use this information to simplify the problem

5) b)


Re DH=5 MRV=3 Red
RI DH=4 MRV=2 Blue
S DH=2 MRV=2 Blue
N DH=1 MRV=2 Green
V DH=0 MRV=1 Red
C DH=0 MRV=1 Green
D DH=0 MRV=1 Green
W DH=0 MRV=1 Green
T DH=0 MRV=3 Red.

5c)



N = Green, V = Red, Ri = Blue, W = Green, Re = Red, C = Green, S = Blue, D = Green, I = Blue.