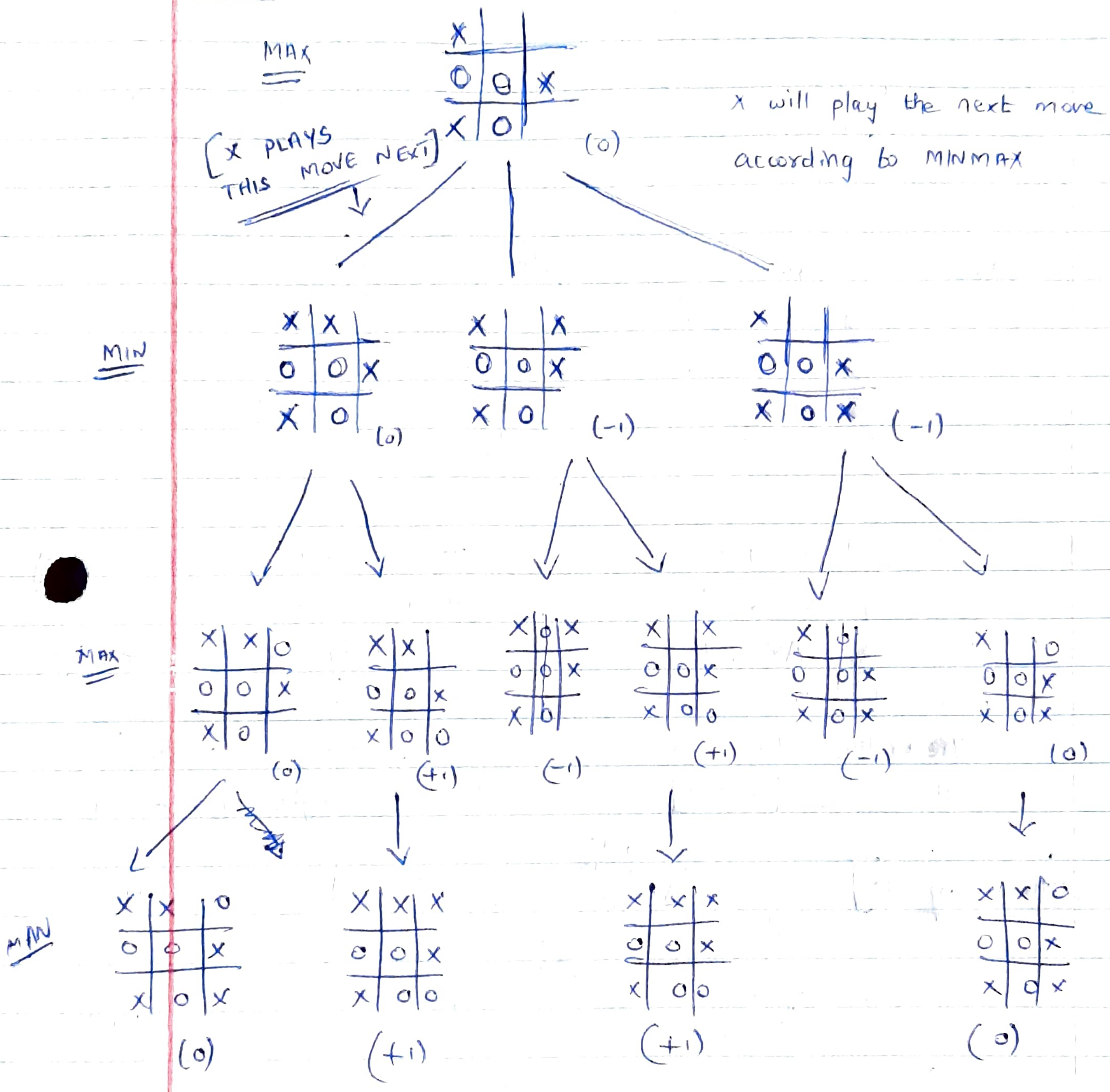
SUDHARSAN SRINIVASAN
1001755919

ASSIGNMENT - 2

TASK 1:



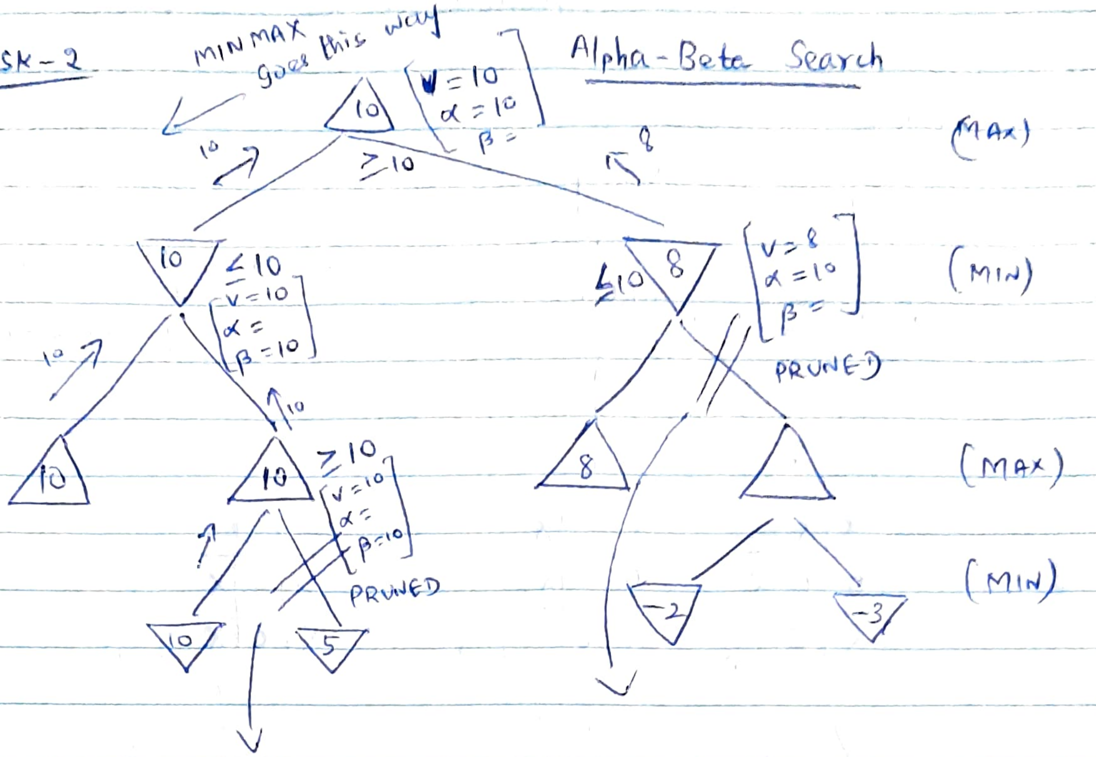
X will play the next move according to MINMAX

The above game, for X it plays the move indicated on the left sub-tree

TASK-2

Alpha-Beta Search

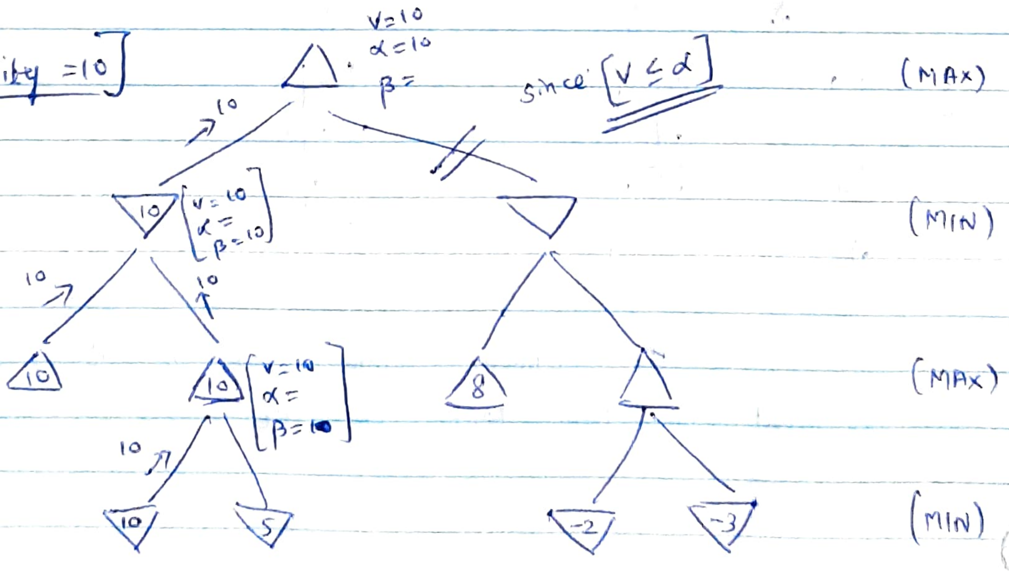
a)



Here, it is PRUNED because if it is less than 10, MAX will ignore it and if it is greater than 10, MIN will ignore. [v ≤ α]

Here, the below nodes will be PRUNED because the ublity 8 is already lesser than root ubility 10 and further expansion towards right [v ≤ α]

b) [Max utility =10]



Since we already have 10, entire right subtree is PRUNED

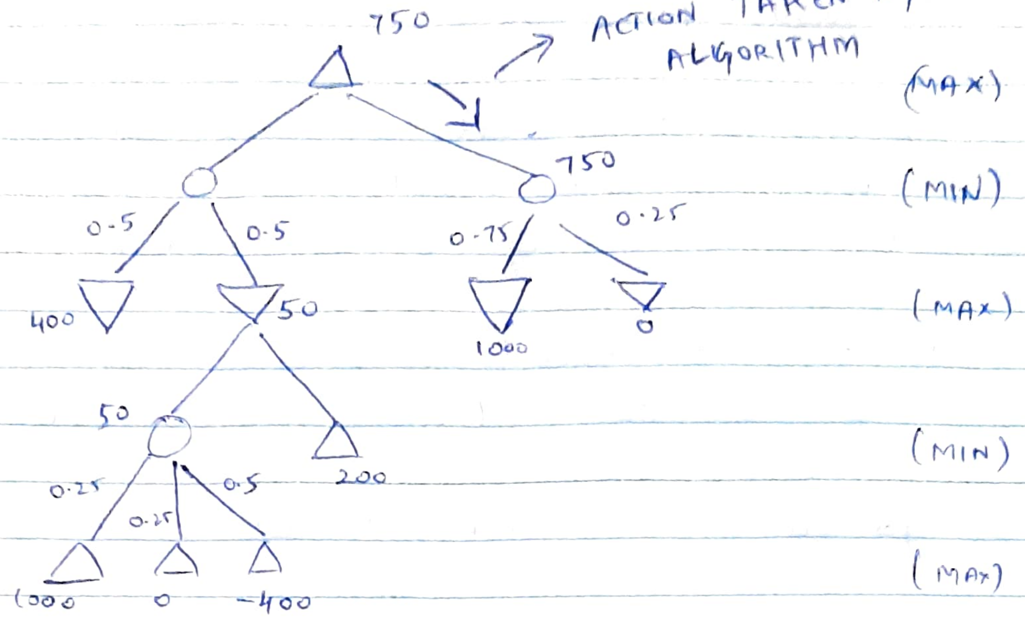
TASK-3:

Pseudocode for 2-player deterministic game:

```
function MINIMAX_DECISION (STATE) return action
  inputs : state, current state in the game.
  return a in Actions (STATE) maximizing MIN_VALUE (Result,
                                                    (a, state))

function MAX_VALUE (state) returns a UTILITY VALUE
    if TERMINAL_TEST(state) then return UTILITY (state)
        $v \leftarrow -\infty$
  for a, s in SUCCESSOR (state) do $v \leftarrow$ MAX(v, MIN_VALUE(s))

  return v

function MIN_VALUE (state) returns a UTILITY VALUE.
    if TERMINAL_TEST (state) then return UTILITY (state)
        $v \leftarrow +\infty$
  for a, s in SUCCESSOR (state) do $v \leftarrow$ MIN (v, MAX_VALUE(
                                                  DeepGreen Move (s)))

  return v
```

Here, when MIN_VALUE is changed, rather than exploring all states, states returned by DeepGree Move () are only considered. This reduces number of states being replaced thus optimizing the solution.

TASK - 4 EXPECTIMINMAX TREE



Here $[((1000 \times 0.25) + (0 \times 0.25) + (0.5 \times -400)] = 50$

$(400 \times 0.5) + (50 \times 0.5) = 225$

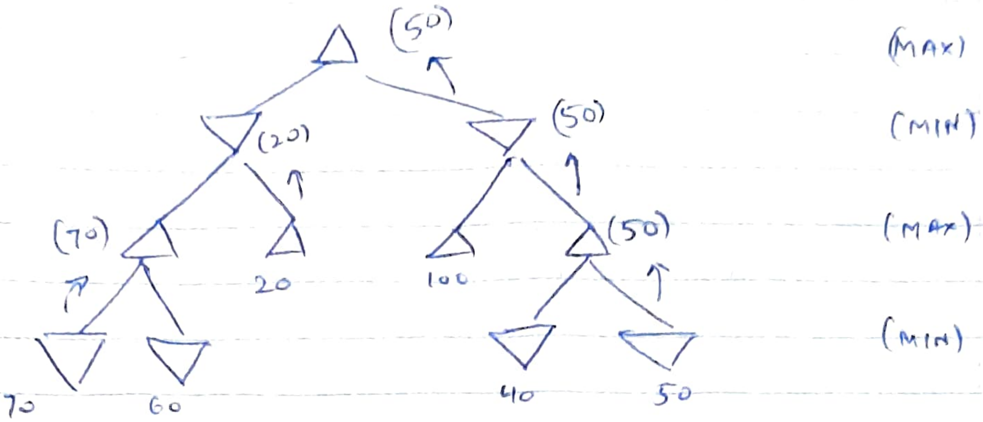
$(1000 \times 0.75) + 0 = 750$

Lowest possible outcome is 0 with 0.25 probability.

Highest possible outcome is 1000 with 0.75 probability.

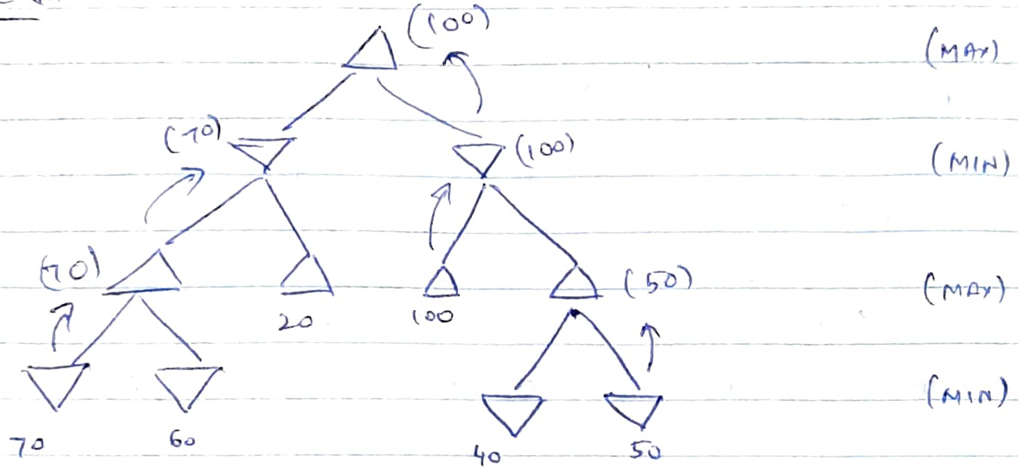
TASK - 5 MINIMAX TREE

Case(i):



Here the possible outume for MAX is 50//

Case (ii)



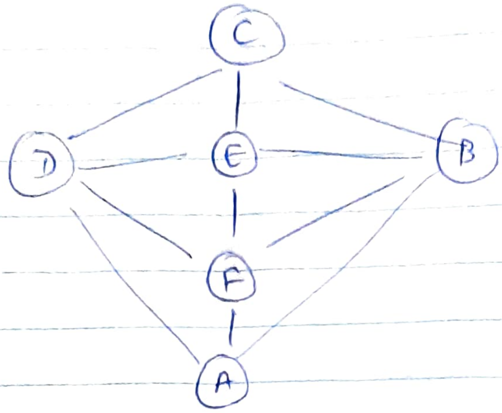
Here, the possible outcome is 100//

BEST POSSIBLE OUTCOME: 100 if the MIN-VALUE works opposite of how it usually works.

WORST- POSSIBLE OUTCOME : 50 if the MIN-MAX works the usually way it does.

TASK-6 GRAPH COLORING

a) Constraint Graph:



b)

Nodes:	A	B	C	D	E	F
MRV :	3	3	3	3	3	3
Degree :	3	4	3	4	4	4

Choose the node with the lowest MRV and maximum degree. Here we can pick either B, D, E (or) F

We pick B. Now the MRV & degree for remaining nodes are as follows.

Nodes	A	C	D	E	F
MRV	2	2	3	2	2
Degree	2	2	4	3	3

Now we pick E

Nodes	A	C	D	F
MRV	2	1	2	1
Degree	2	1	3	2

Now we pick F

Nodes	A	C	D
MRV	1	1	1
Degree	1	1	2

Now, we pick D

Nodes	A	C
MRV	0	0
Degree	0	0

pick Either A (or) C

∴ Final selection of variables is

B → E → F → D → C → A

c) Order of Variables:

B E F D C A

Now, we assign the color "Red" to B

If B is Red, then its neighbors can't be assigned Red. Therefore using Arc consistency,

B	R
E	~~R~~
F	~~R~~
D	R
C	~~R~~
A	~~R~~

Now we assign "Green" to A

B	R
E	~~R~~
F	~~R~~ ~~G~~
D	R
C	~~R~~
A	~~R~~ G

Since A is "Green", its neighbor F can't be Green or Red, so F is "Blue"

B	R
E	~~R~~ ~~B~~
F	~~R~~ ~~G~~ B
D	R
C	~~R~~
A	~~R~~ G

Since F is "Blue", its neighbor E can't be Blue or Red, so E is Green

B	R
E	~~R~~ ~~B~~ G
F	~~R~~ ~~G~~ B
D	R
C	~~R~~
A	~~R~~ G

→ Finally C is assigned Blue

B	R
E	~~R~~ ~~B~~ G
F	~~R~~ ~~G~~ B
D	R
C	~~R~~ ~~G~~ B
A	~~R~~ G

d) One valid solution:

A	→	GREEN
B	→	RED
C	→	BLUE
D	→	RED
E	→	GREEN
F	→	BLUE