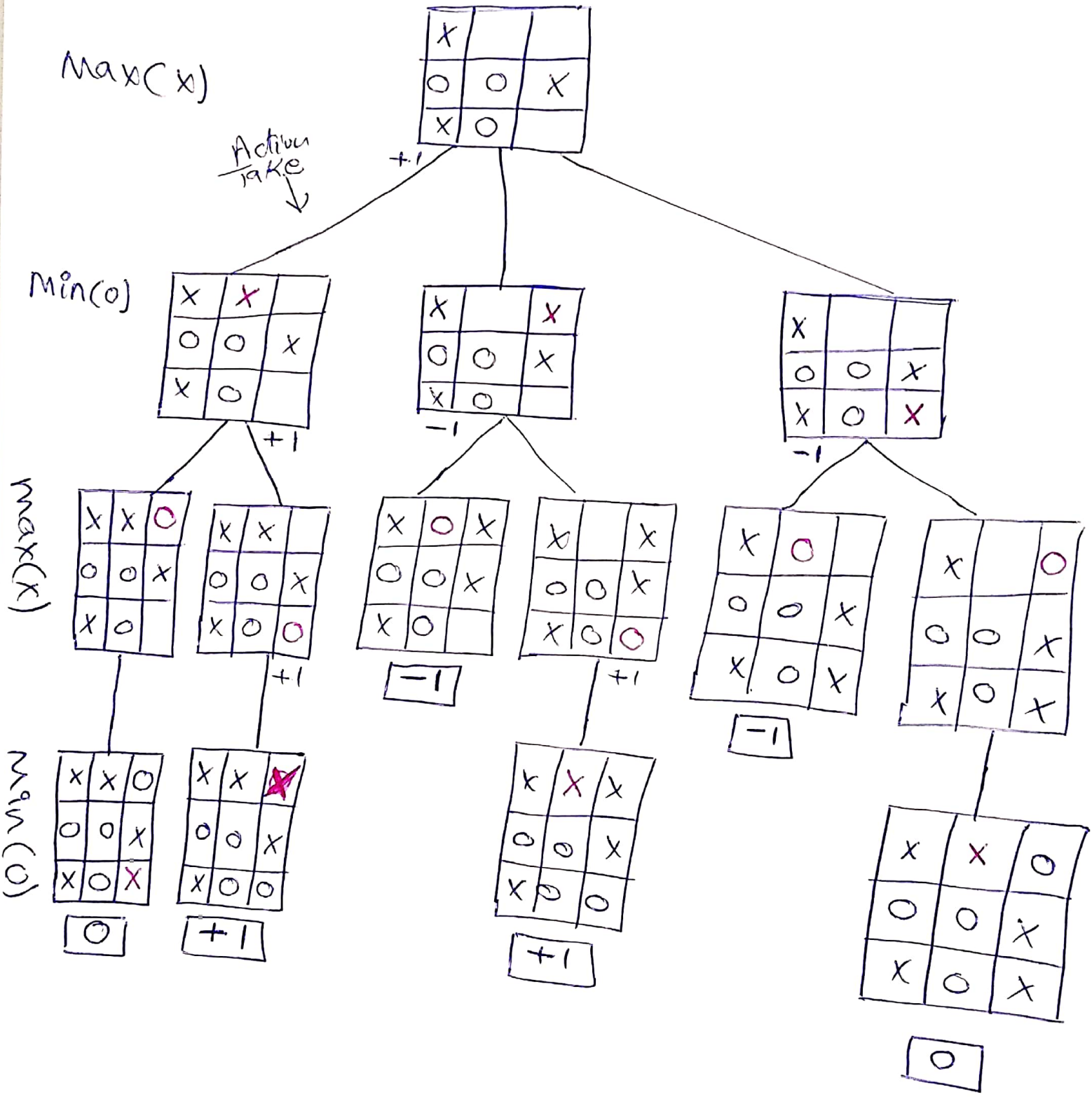
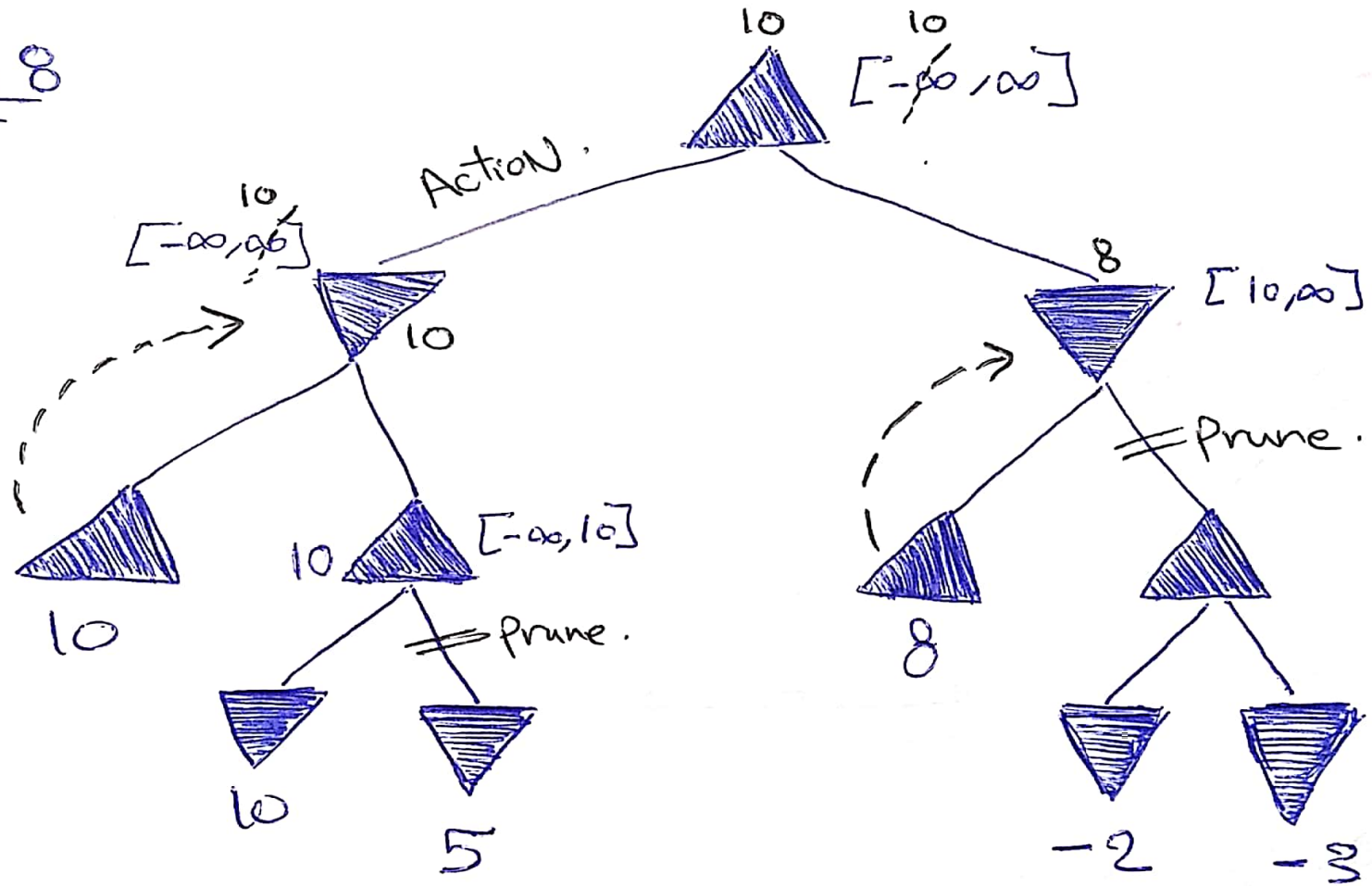
"Assignment 2"



Task 2 8

(a)



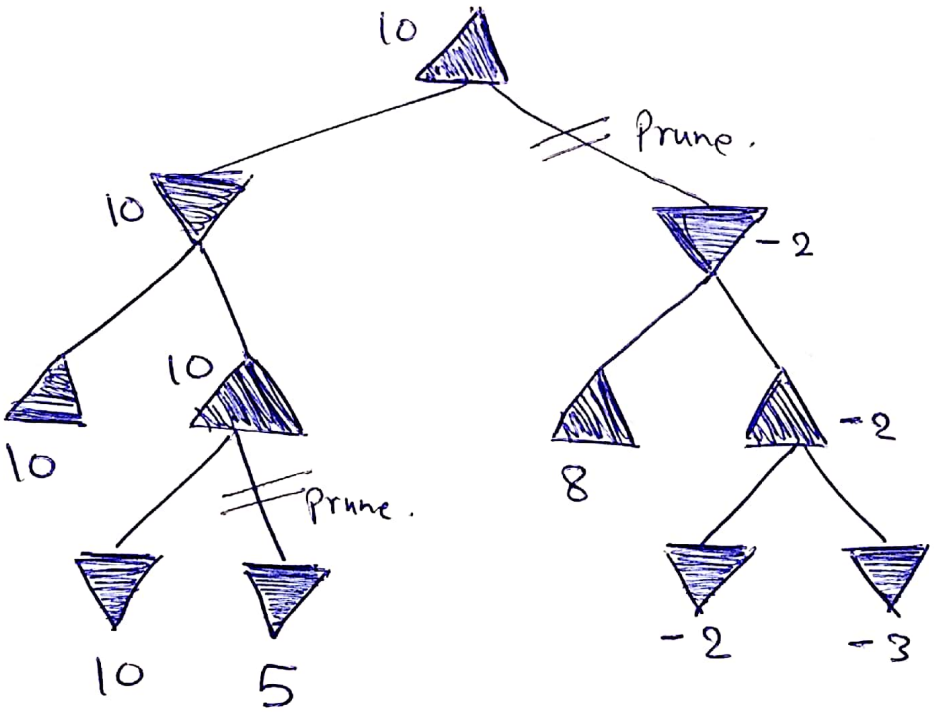
Task 2-b:

The knowledge given: Max utility value = 10

I know, Max value = 10

Min " = -3

If we know that Max utility is 10 then, we do not need to explore the other childs (successors) Because we know when we look to graph below we can figurout that left branch gives value of 10 to Max Node So, we don't need to explore the right branch and therfore will be pruned.



Task 3 8

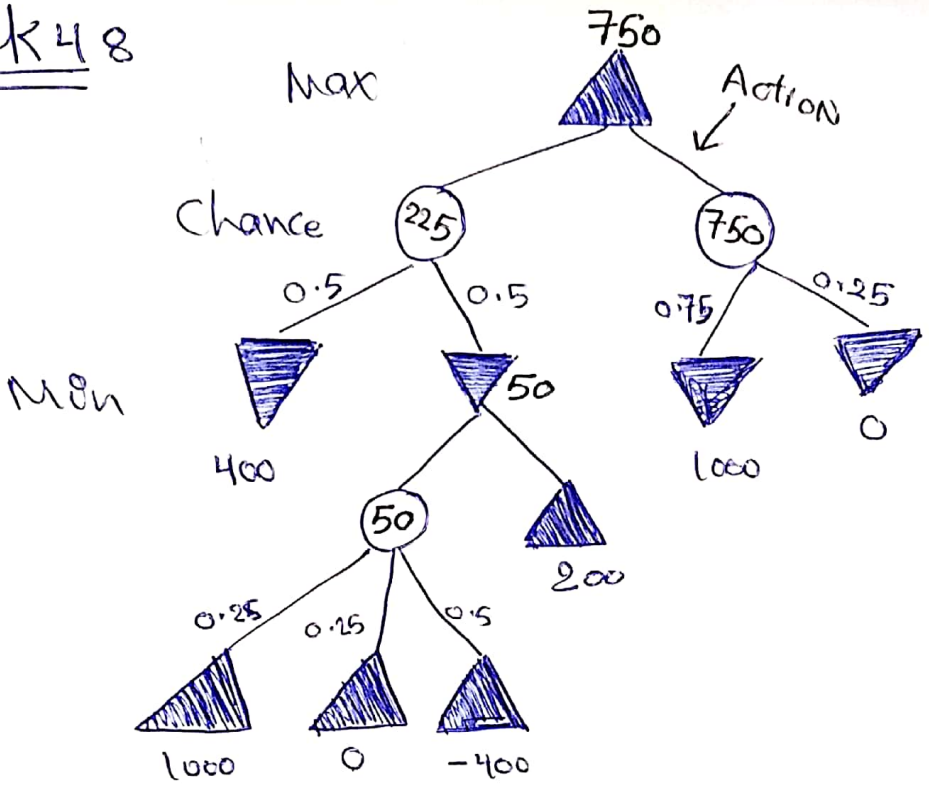
Pseduo Code:

```
function Min-Value(S) return utility value
    if terminal-test(S)
        return utility(S)
    return Max-Value(DeepGreenMove(S))
```

Since, I do have function that can return state of DeepGreen move then I can perform it to Search less Nodes whether the movement of Perfect or NOT.

if Deep Green play optimal Stratigy then will return MinMax tree, ;nother Hand, will visit fewer nodes if DeepGreen Plays Sub-optimal.

Task 4 8



$(0.25 * 1000) + (0.25 * 0) + (0.5 * -400) = 250 + 0 - 200 = 50$

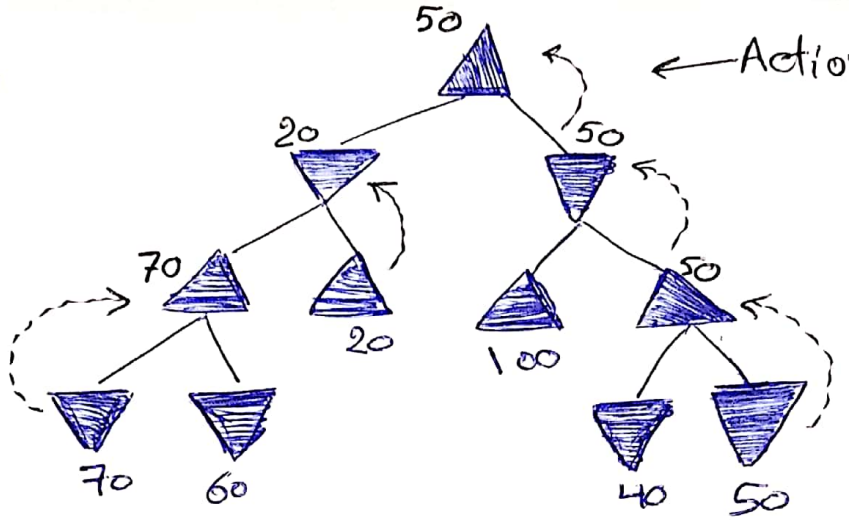
$(400 * 0.5) + (50 * 0.5) = 200 + 25 = 225$

$(1000 * 0.75) + (0 * 0.25) = 750$

∴ lowest outcome : [redacted] 0

highest " : 1000

Task 5 8



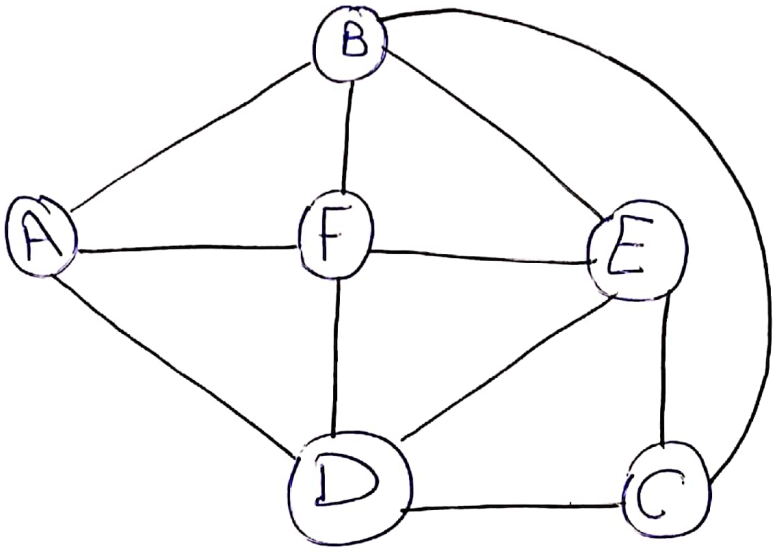
the Best possible outcome : 100
the worst " " : 50
} for Max player.

if the opponent play Random strategy.

but if opponent play optimal strategy
then
Best possible outcome : 50
worst possible outcome : 50

Task 6:

(a)



(b)

Nodes	RV	DH
D	3	4
E	2	3
F	1	2
B	1	2
A	1	0
C	1	0

(C)

initially set List & assign Reel → (D)

D	E	F	B	A	C
RBG	RBG	RBG	RBG	RBG	RBG

Step 1 :-

D	E	F	B	A	C
R	GB	GB	RGB	GB	GB

List Arcs

A → D ~~F → D~~ ~~E → D~~ ~~C → D~~

Step 2 :-

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

Arcs Checked

~~C → E~~ ~~D → E~~ ~~F → E~~ ~~B → E~~

~~E → C~~ ~~B → C~~ ~~D → C~~ ~~A → F~~ ~~B → F~~ ~~E → F~~

~~D → F~~ ~~C → B~~ ~~E → B~~ ~~A → B~~ ~~F → B~~

~~A → B~~ ~~E → B~~ ~~C → B~~ ~~F → B~~ ~~B → A~~

~~F → A~~ ~~D → A~~

Done !!

(e)

D → Red

E → Green

F → Blue

B → Reed

A → Green

C → Blue.