Name :- Kunal G. Bachute

Class :- B.E / IT

Roll.No :- 02

Subject :- IS lab.

DOP	DOA	Remark	Sign.

Min Max with Alpha Beta for Tic Tac Toe

→ The goal of Tic-Tac-Toe is to be the first player to get three in a row on 3x3 grid.

→ "x" always goes first.

→ players alternate placing "x"s & "O"s on board until either:

① one player has three in a row horizontally vertically or diagonally.

② All nine squares are filled.

→ programmer Created in "Winningstate" named set Containing a list of all possible win Conditions inside "properties.py", if a player places 'x's or "O"s is any of the list, they are declared winner.

The winning States are:

Winning states = ([0,1,2], [3,4,5], [6,7,8], [0,3,6], [1,4,7], [2,5,8], [0,4,8] [2,4,6])

→ programmer has Created a dummy bot which chooses positions randomly as DummyBot. The GameBoard initializes the free spaces to None. (List of Nones).

→ programmer also created a minmax bot which uses min Max Algorithm with Alpha Beta pruning to decide the best move.

Q2.

a).

→

The lowest path cost g(n) Can be the cost to reach the goal configuration in least steps.

In our Case, we Can reach the final Configuration in at least 4 moves ; up, up, LEFT, LEFT.

Since all moves are equally Costly, we Compute g(n) as.

$$g(n) = 1 + 1 + 1 + 1.$$

$$g(n) = 4$$

Consider the following 8 puzzle instance!

8	7	6
2	1	5
-	3	4

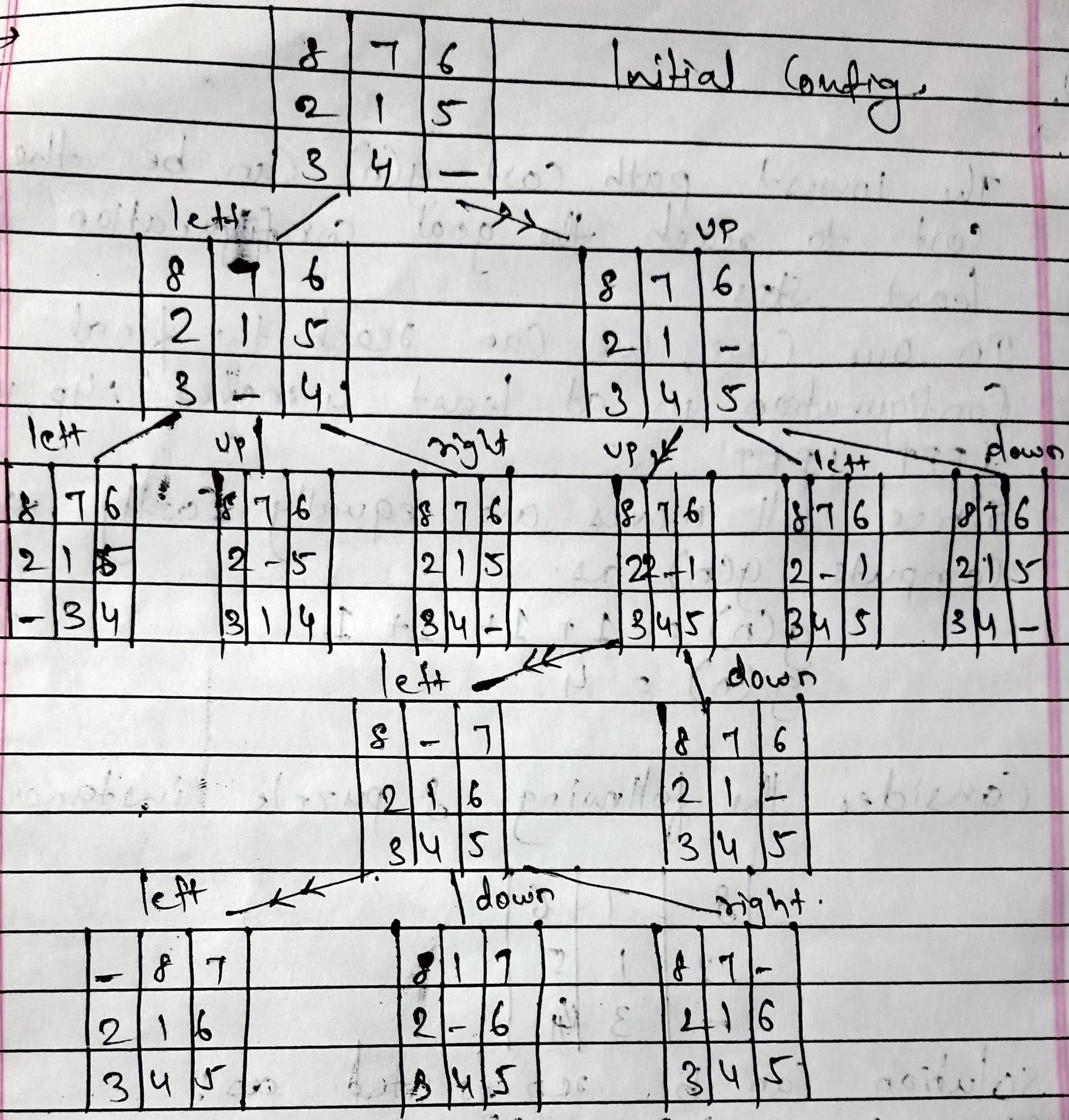
Solution Can be represented as:

{{8,7,6} {2,1,5} {-,3,4}} → {{8,7,6} {2,1,5} {3,-,4}} → {{8,7,6} {2,1,5} {3,4,-}} → {{8,7,6} {2,1,-} {3,4,5}} → {{8,7,6} {2,1,5} {3,4,5}} → {{8,7,-} {2,1,6} {3,4,5}} → {{-,8,7}, {2,1,6}, {3,4,5}}

Since all the moves are equally costly the Cost would be

$$g(n) = 6$$

c) →



8	7	6
2	1	5
3	4	-

left

8	7	6
2	1	5
3	-	4

up

8	7	6
2	1	-
3	4	5

left

8	7	6
2	1	5
-	3	4

up

8	7	6
2	-	5
3	1	4

right

8	7	6
2	1	5
3	4	-

up

8	7	6
22	-	1
3	4	5

left

8	7	6
2	-	1
3	4	5

down

8	7	6
2	1	5
3	4	-

left

8	-	7
2	1	6
3	4	5

down

8	7	6
2	1	-
3	4	5

left

-	8	7
2	1	6
3	4	5

down

8	1	7
2	-	6
3	4	5

right

8	7	-
2	1	6
3	4	5

Final Configuration.

→ The main.py starts by initialization of two objects named of MinMax Bot & Dummy Bot the code then creates a variable Judge which called Tic Tac Toe Judge, to which both objects are passed, the TicTacToe Judge.py decides the winner.

→ programmer also created a Helper method, Helper.py which gets the opponents position to bot and gets the available moves to play imports properties.py mentioned earlier.

* Inputs: → No inputs from user.
(as both the bots, Dummy Bot & minmax Bot play the Game)

* Output: i) Winner Name which can be:
(a) Bot One (minmax Bot).
(b) Bot Two (Dummy Bot)
(c) Draw (when all positions are filled with no winner).

The winner is decided if the bot's position is in the set of list of Winning states().