Minimax - Algorithm.
Module - 3

Name:- Bhavesh Santosh Ainkar

Roll No:- 1

Class:- BE/IT

Sem:- VII

Sub:- ISLAB

Batch:- I-1

Mini-Max Algorithm (Module-3)
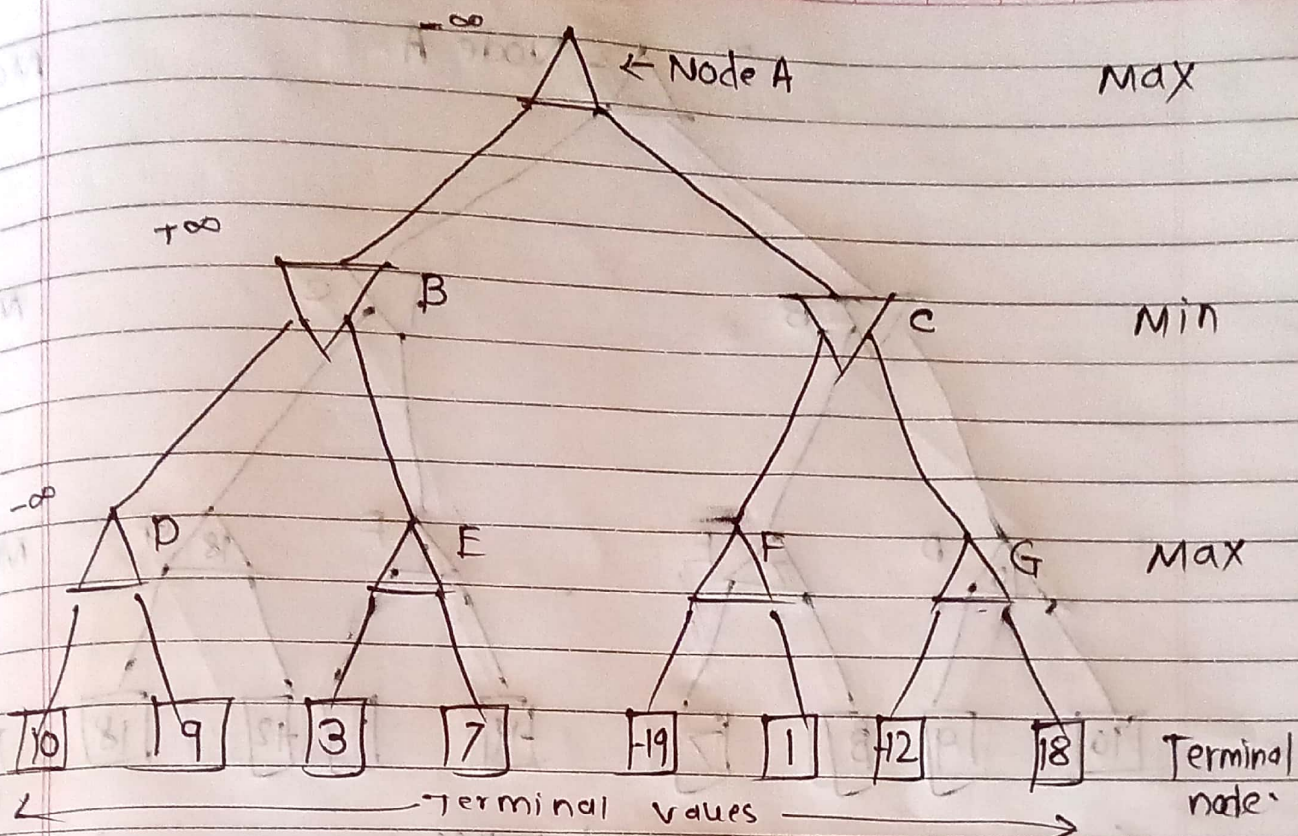
A Mini-Max algorithm
→ Minimax algorithm is a recursive or backtracking algorithm which is used in decision making and game theory. It provides an optional move for the player assuming that opponent is also playing.

- Mini-Max algorithm uses recursion to Search through the game-tree

- In this algorithm two players Play the game one is Call Max and other is Called MIN

- Min-Max algorithm is Mostly used for gameplay in AI. Such as Chess, checkers, tic-tac-toe. This Algorithm Computers the minimax decision for the current state.

Step1: In the first step, the algorithm generates the entire game tree and apply the utility function to get the utility values for the terminal states.
In Below tree diagram lets take A is the initial state of the tree.
Suppose Maximizer takes first turn which has worst-case initial value = -infinity, and Minimizer will take next turn which has worst-case initial value = +∞

$-\infty$

← Node A   Max

$+\infty$ B   C   Min

$-\infty$ D   E   F   G   Max

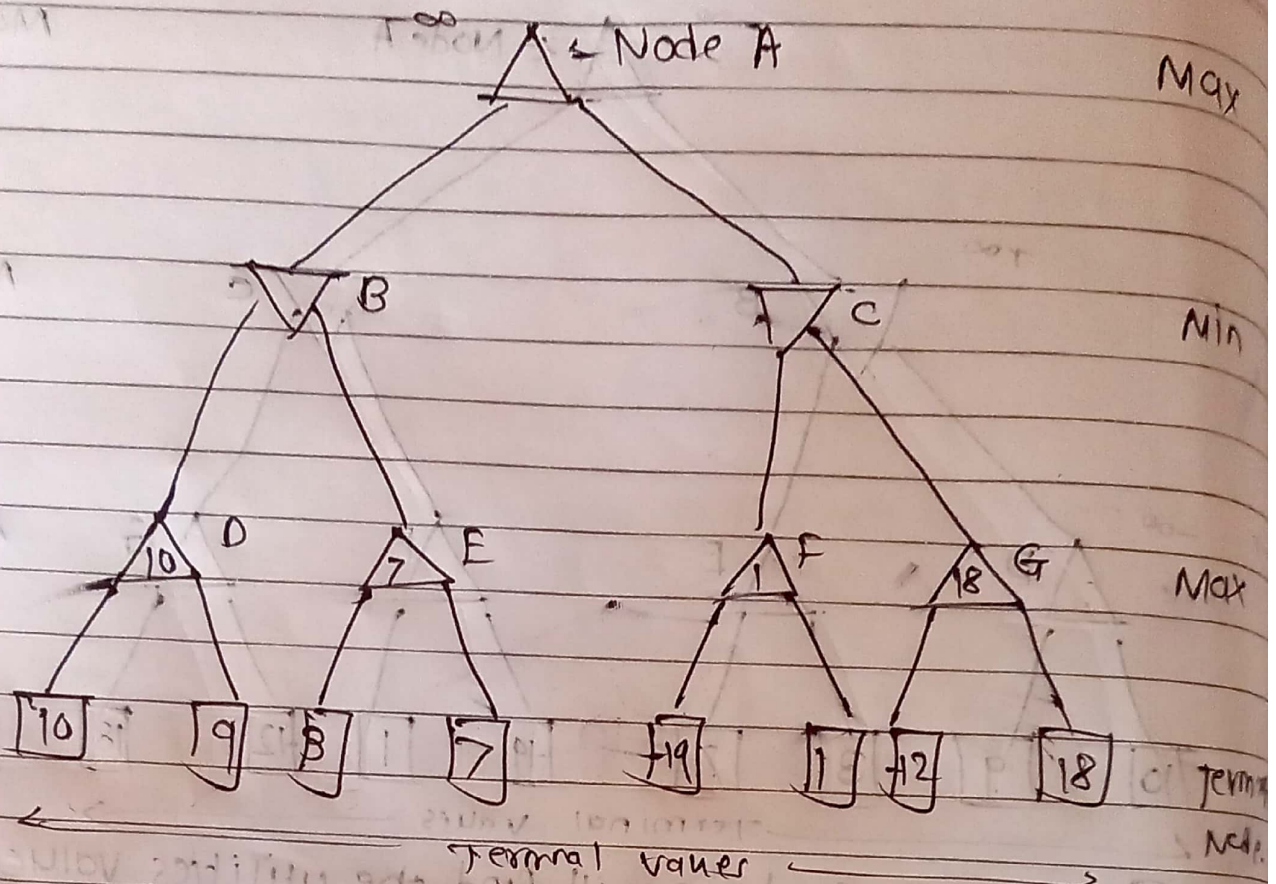| 10 | 9 | 3 | 7 | -19 | 1 | 12 | 18 | Terminal node: |

Terminal values

Step 2:- Now, first we will find the utilities value for the Maximizer. it initial value is $-\infty$ so we will Compare each value in Terminal state with initial value of Maximizer and determines the higher nodes values. It will find the Maximum among the all.

- for node D   max $(10, -\infty) = $ max $(10, 9) = 10$

- for node E   max $(3, -\infty) = $ Max $(3, 7) = 7$
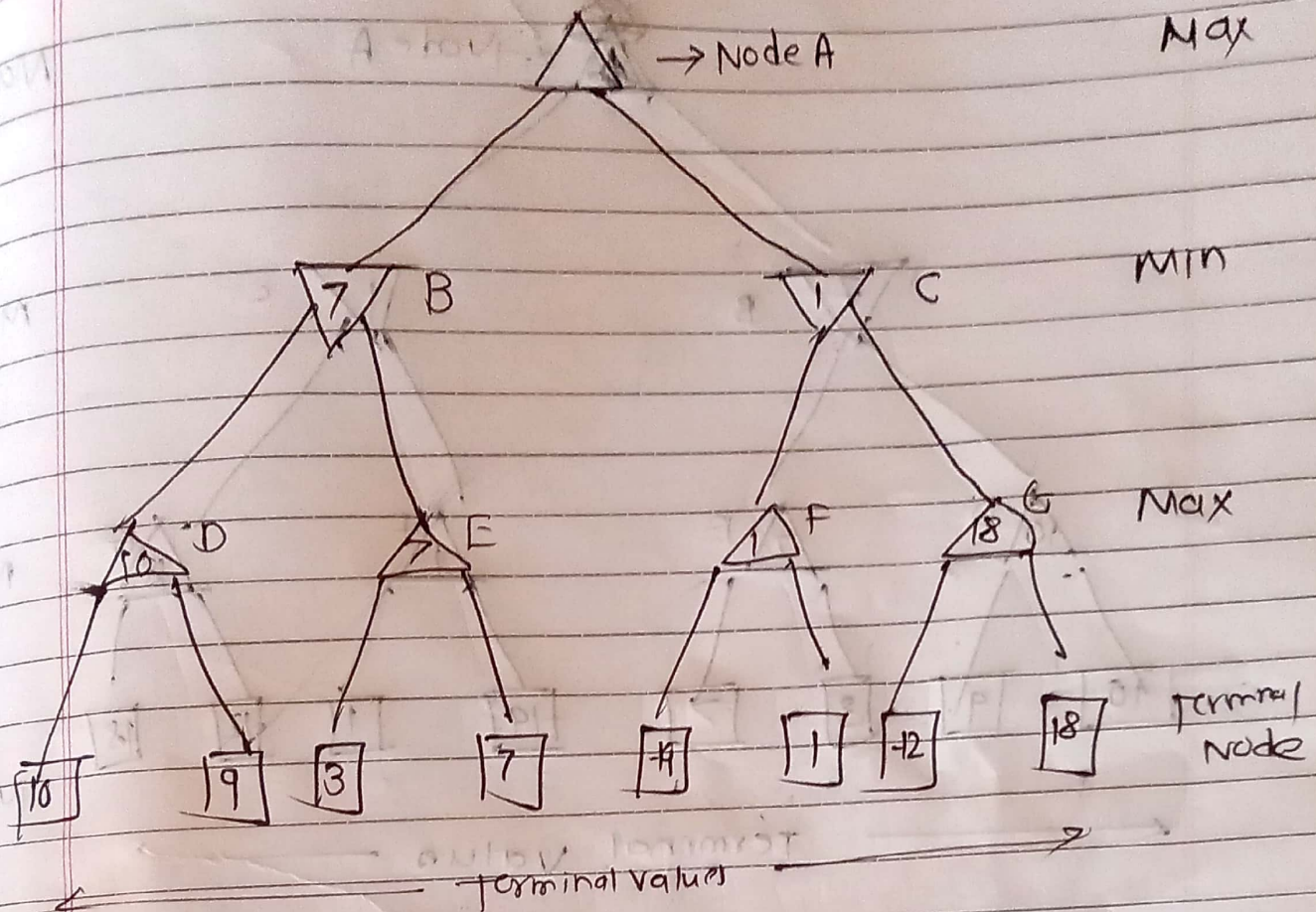
- for node F   max $(-19, -\infty) = $ Max $(-19, 1) = 1$

- for nod G   Max $(-12, -\infty) = $ max $(-12, 18) = 18$

∞ → Node A   Max

B   C   Min

10 D   7 E   1 F   18 G   Max

[10]   [9]   [8]   [7]   [19]   [1] [12]   [18]   Termi
                                                    Nel.

Terminal values →

Step 3:- In the next step, its a turn for minimizer
so it will compare all nodes value
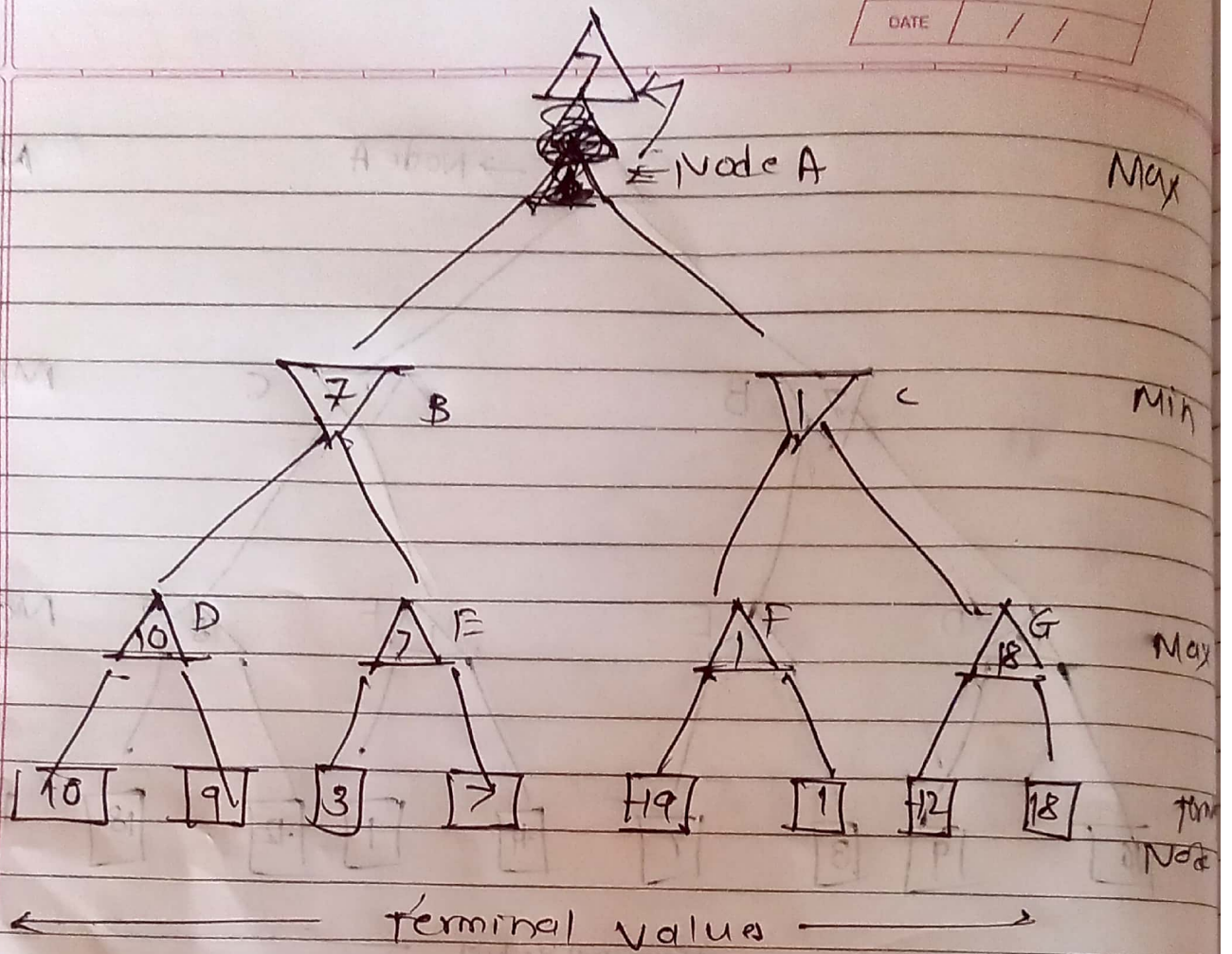with +∞, and will find the 3rd layer
nod values.

— for Node B = min(10, 7) = 7
— for node C = min(1, 18) = 1

→ Node A   Max

7 B   Min

1 C

D   E   F   8 G   Max

10   9   3   7   -4   1   -12   18   Terminal Node

Terminal Values

Step 4: — Now its a turn for Maximizer and it will again choose the Maximum of all nodes Values and find the Maximum Value for the root node. In this game tree, there are only 4 layers, hence we reach immediately to the root node, but in real games, there will be more than 4 layers.

— for Node A   max (7, 1) = 7

Node A   Max

7   B   C   Min

D   E   F   G   Max
10   7   1   18

| 10 | 9 | 3 | 7 | 19 | 1 | 12 | 18 |

← terminal value →

That was the complete work flow of the
Mimimax algoritm with two player game