Adversarial Search and Game Playing.

* Explain Constraint-Satisfaction Problem with example.

=> Constraint-Satisfaction is an arithmetic problem which is show in letters.

It involves the decoding of digit represented by a character.

In this Problem, Some Arithmetic equation where digits are distincly represented by some characters.

Assign a decimal digit to each of the letters in such a way that the answer to the problem is correct.

=> Rules:

1 Each letters represented only one and a unique digit throughout the Problem.

2 When the digits replace letters, the resultant arithmetical operation must be correct.

Ex.
```
  S E N D
+ M O R E
---------
M O N E Y
```

=> Step 1 : In this problem, M must be 1.

```
  S E N D
+ 1 O R E
---------
1 O N E Y
```

=> Step 2 : Now value of S will be 8 or 9.
If S = 8 then Carry = 1
If S = 9 then Carry = 0

Take S = 9.

```
  9 E N D
+ 1 O R E
---------
1 O N E Y
```

Here, value of O is must be 0.

```
  9 E N D
+ 01 O R E
---------
1 0 N E Y
```

=> Step : 3 : We have to find value of E and N.

For No Carry,

$$N + R = 10 + (N - 1)$$
$$N + R = 9 + N$$
$$R = 9$$

But 9 is already taken,

So, With carry,

$$N + R + 1 = 10 + (N - 1)$$
$$N + R + 1 = 10 + N - 1$$
$$R = 8$$

```
   9  E  N  D
+  1  0  8  E
-----------
1  0  N  E  Y
```

They are 7, 6, 5, 4, 3 and 2 are left.

Let's take E=5 with carry one
So, value of N will be 6

```
      1     1
   9  5  6  D
+  1  0  8  5
-------------
1  0  6  5  Y
```

Here, We have to take carry at 6+8+1.

=> Step 4 : They are 7, 4 and 3, 2 are left.

We have to select those digit which is generate carry.

Here, Only 7 can generate carry. Then Value of Y is become 2.

$$\begin{array}{cccc} & 1 & 1 & \\ \therefore & 9 & 5 & 6 & 7 \\ + & 1 & 0 & 8 & 5 \\ \hline 1 & 0 & 6 & 5 & 2 \end{array}$$

This digit are perform arithmetic Operation very well.

* Explain MIN-MAX Algorithm with example.

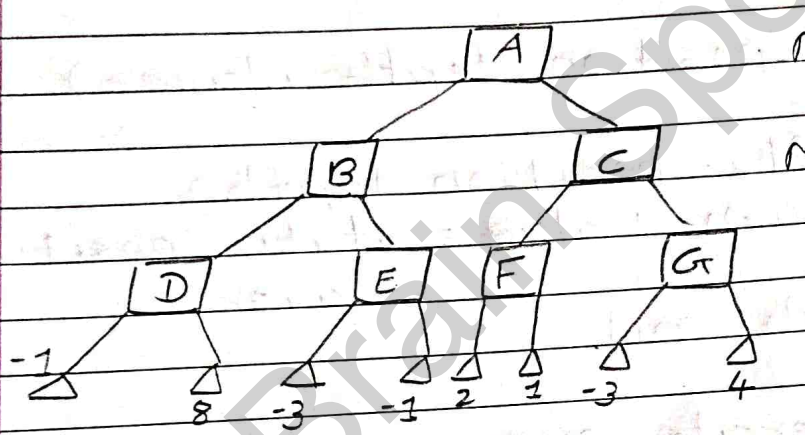
=> MIN-MAX Algorithm is use Depth First Search for the Searching Process.

This algorithm is allows to do backtracking but does not use BFS For Searching.

MIN-MAX Algorithm is used Best Move method for the searching process.

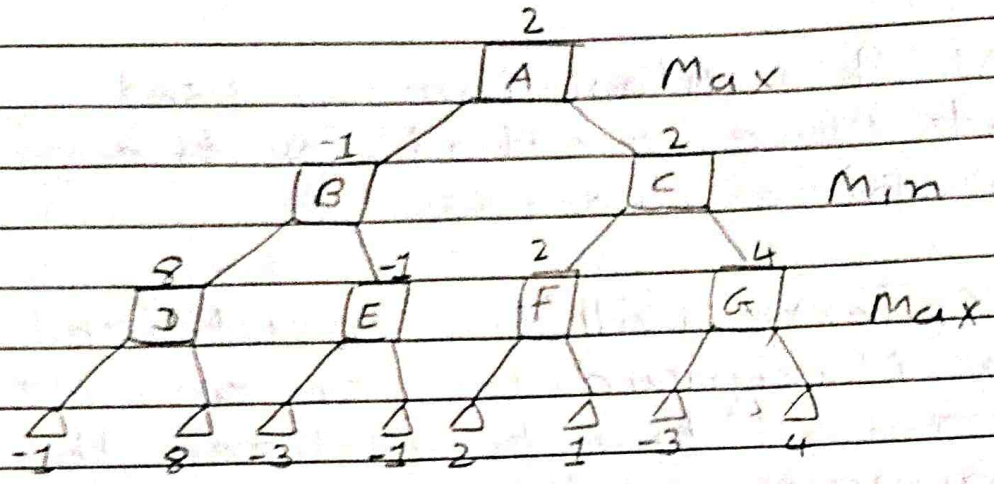
MAX Term will try to utilized the Maximum utility and MIN Term will try to utilized the minimum utility.

Ex.



For Max -> We have to select Maximum value.
Min -> We have to select Minimum value.

In Min-Max Algorithm, we have starting searching at the deapth of the graph.



=> Algorithm:

MTDMAX (position, depth, player)

1 IF Last Ply (position, depth)
then return value = static (position, player)
path = nil.

2 else, Generate one more ply of the tree by calling the function Move Gen (position, player) and set SUCCESORS to the list it returns.

3 If SUCCESSOR is empty,
then no moves to be made.
Return the same structure that would have been returned if Last_Ply had returned true.

4 If SUCCESORS is empty,
then examine each element in turn and keep track of the best one.

5 After examining all the nodes,
Return Value = Best-Score
Path = Best-Path

* Explain Alpha-Beta Algorithm with example.

=> Alpha-Beta Algorithm is a higher version of a Min-Max Algorithm.

In Min-Max Algorithm, we have to visited the all the node to Find the solution, this is take more time to solve the problem.

In Alpha-Beta Algorithm, we does not have to visit all the node to find the solution.

Alpha-Beta Algorithm is used to reduce the number of states to solve the problem.

In this algorithm, Those node is not visited, This node is called Cut-off.

Alpha-beta Search proceeds in a depth first Search rather than searching the entire space.

The alpha value is associated with Max nodes and Beta value is associated with Min nodes.

Ex.