

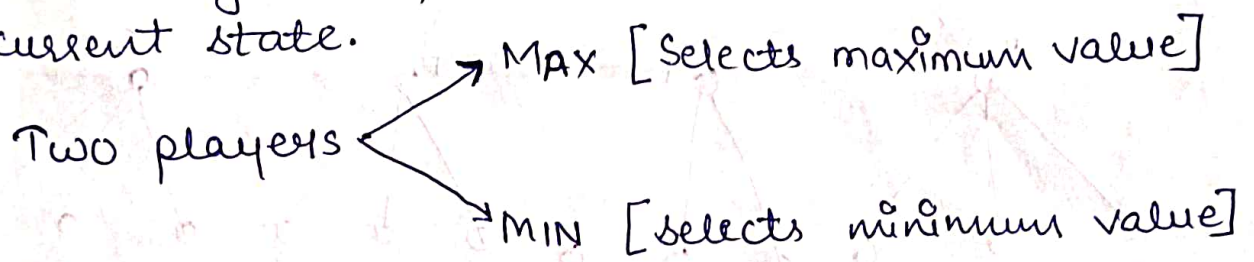
Unit - 4.

== Minimax procedure / theorem :-

It is a specialized search algo that returns optimal sequence of moves for a player in Zero sum game.

Recursive / Backtracking algo which is used in decision making & game theory.

- It uses recursion to search through game tree. Algoⁿ computes minimax decision for current state.



Properties :-

- ① Definitely found solⁿ.
- ② Optimal
- ③ Time Complexity = $O(b^m)$
 - maximum depth.
 - branching factor of game tree.
- ④ Space Complexity = $O(b^m)$

Always take initial value of the players

for $\left. \begin{array}{l} \text{Max} = -\infty \\ \text{min} = \infty \end{array} \right\}$ for worst case.

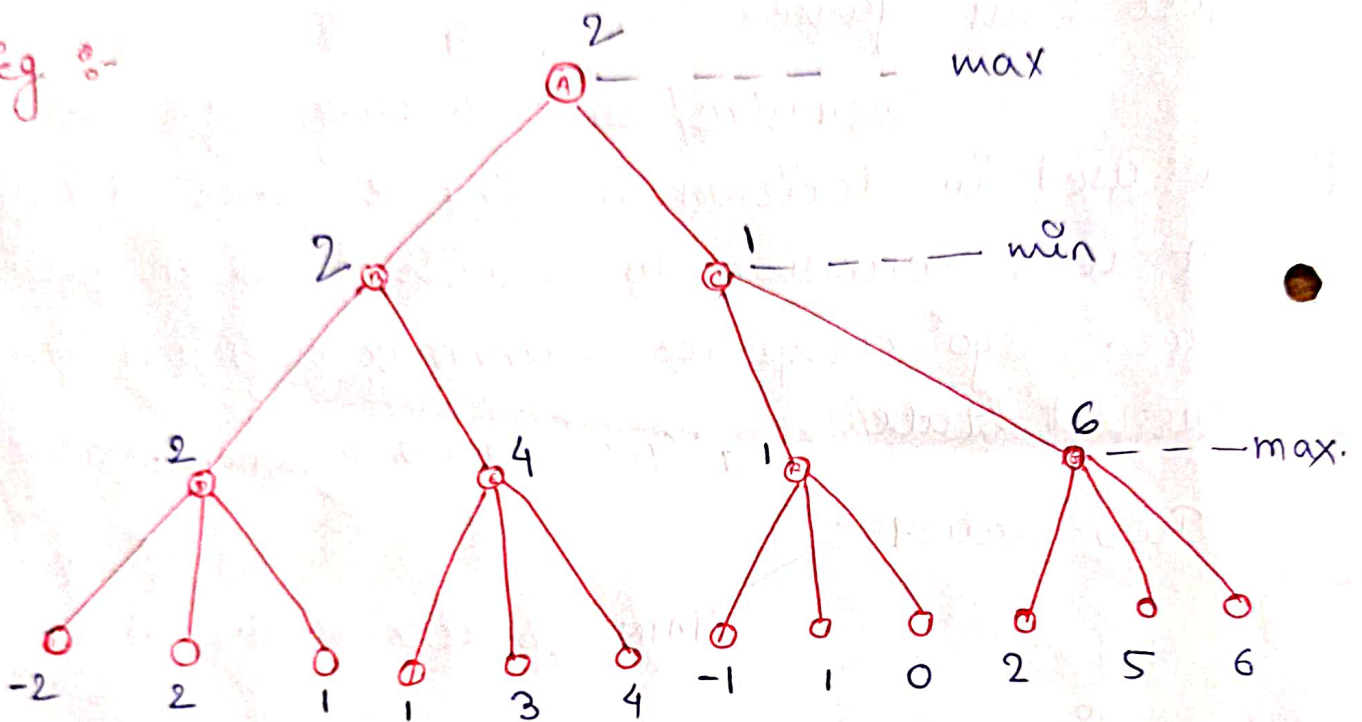
Limitations :-

* Slow for complex games such as chess.

↳ 35 choices/moves. & $d = 100$ (for both players).

↳ $(35)^{100} \rightarrow \text{big}$

Eg :-



initial values $\begin{bmatrix} \text{max} = -\infty \\ \text{min} = \infty \end{bmatrix}$

for maximum :-

① $(-2, \infty)$
↳ $(-2, 2)$
↳ $(2, 1)$
↳ ②

② $(1, -\infty)$
↳ $(1, 3)$
↳ $(3, 4)$
↳ ④

$$\textcircled{3} \quad (-1, -\infty) \xrightarrow{\quad} (-1, 1) \xrightarrow{\quad} (1, 0) \xrightarrow{\quad} \textcircled{1}$$

$$\textcircled{4} \quad (2, -\infty) \xrightarrow{\quad} (2, 5) \xrightarrow{\quad} (5, 6) \xrightarrow{\quad} \textcircled{6}$$

for minimum :-

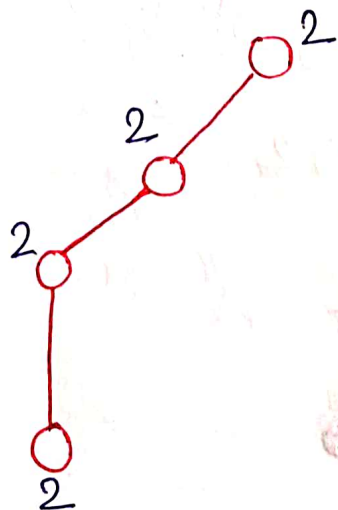
$$\bullet \quad \textcircled{1} \quad (2, \infty) \xrightarrow{\quad} (2, 4) \xrightarrow{\quad} \textcircled{2}$$

$$\textcircled{2} \quad (1, \infty) \xrightarrow{\quad} (1, 6) \xrightarrow{\quad} \textcircled{1}$$

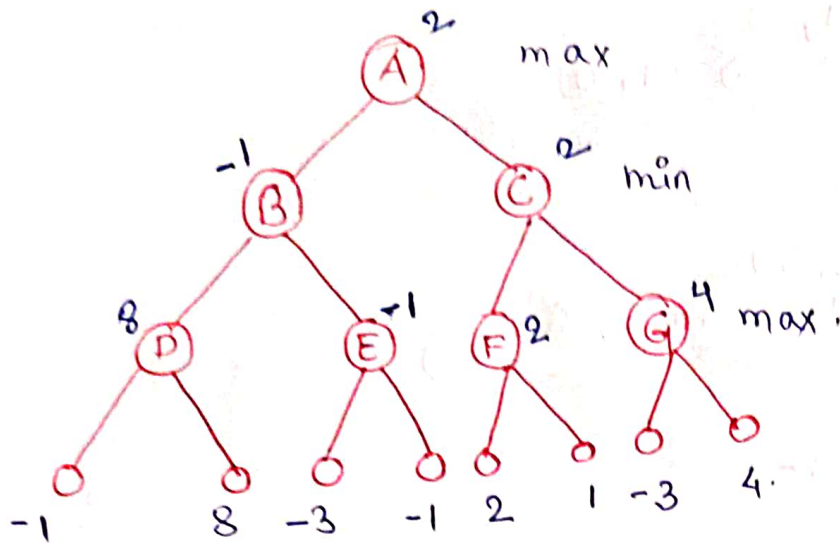
for maximum :-

$$\bullet \quad \textcircled{1} \quad (2, -\infty) \xrightarrow{\quad} (2, 1) \xrightarrow{\quad} \textcircled{2}$$

So the maximum benefit ~~for~~ path for the game.



Que :- Show the maximum profit for the game :-



So maximum value :-

for $D = 8$, $E = -1$, $F = 2$, $G = 4$.

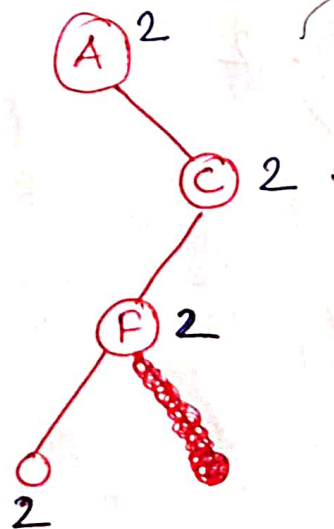
for minimum value :-

for $B = -1$, $C = 2$.

for maximum value

for $A = 2$.

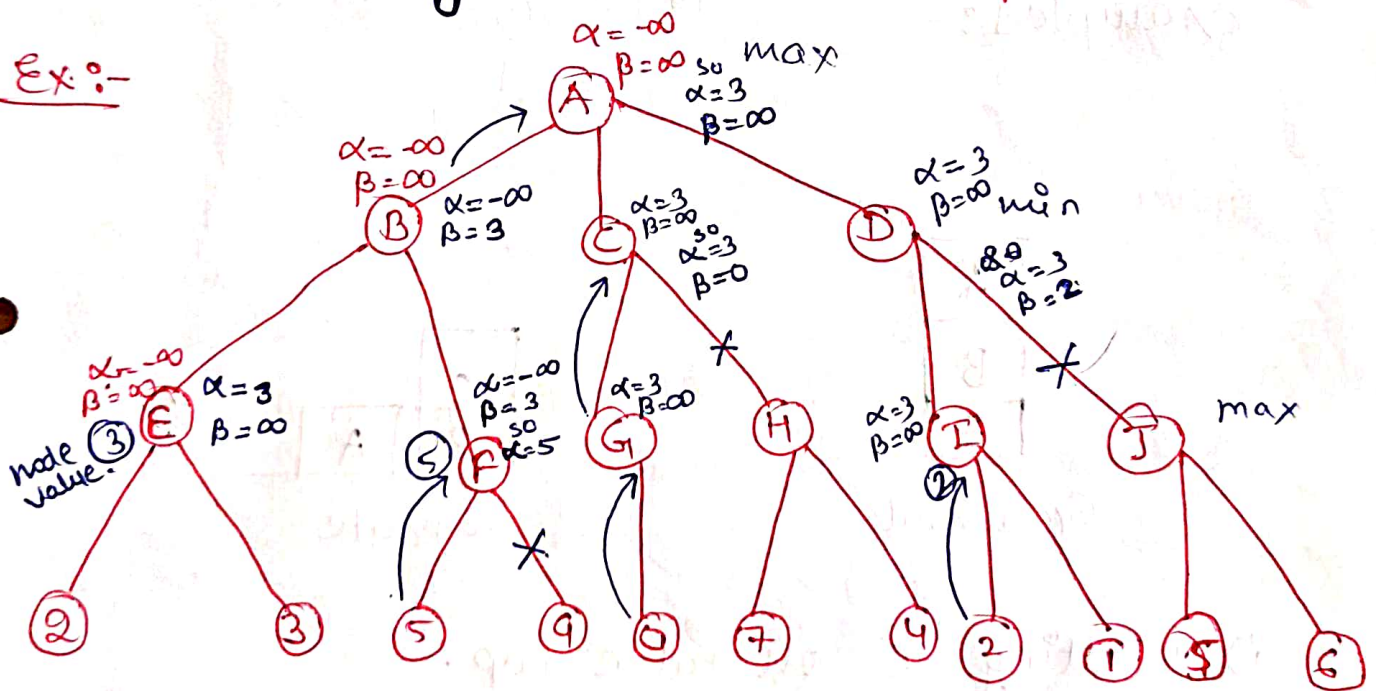
The maximum path will be :-



Alpha = Maximizer (lower bound)
 Beta = Minimizer (upper bound)
 It is a modified version of minimax algo. It is an optimization technique for minimax algorithm.

* There is a technique by which without checking each node of game tree we can compute correct minimax decision & this technique is called pruning. This involves a threshold parameter Alpha & beta for future expansion. So it is called Alpha beta pruning.

Ex:-



So the maximum value is

$\alpha = 3$.

No. of Alpha cutoff = 1

No. of Beta cutoff = 2

Block world Problem :-

There are n numbers of blocks resting on table with specified sequence.

↳ Goal is to arrange in desired sequence.

↳ Available moves $\begin{cases} \rightarrow \text{Put a block on table} \\ \rightarrow \text{Put a block on another block top.} \end{cases}$

↳ State is represented using a sequence of blocks in current position.

Example 1 :-



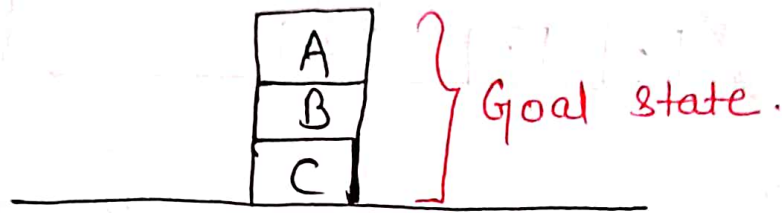
i) Putting 'C' on table top.



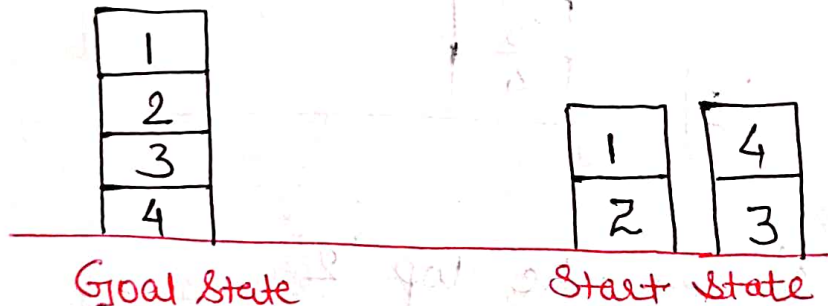
ii) Put 'B' on top of 'C'



iii) Put 'A' on top of 'B'

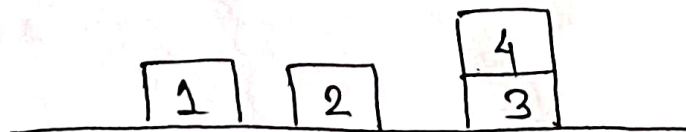


Example 2 :-

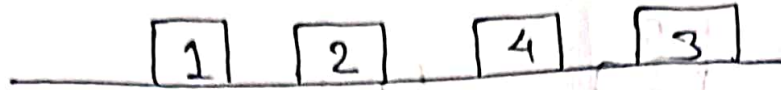


* Always try to place each block of the start state to the table.

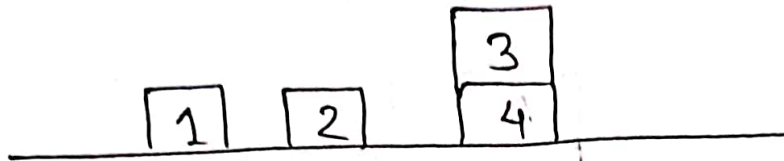
i) Putting '1' on the table



ii) Put '4' on the table.

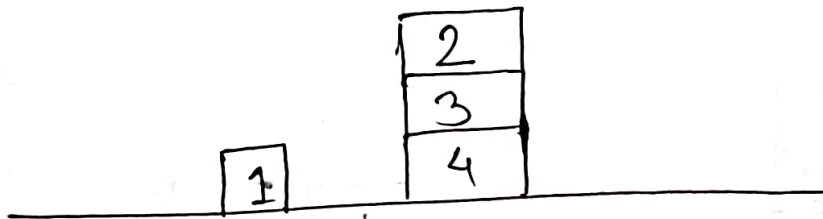


iii) Now put '3' on top '4'

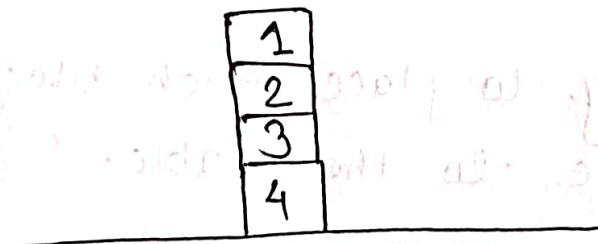


Intermediate
table.

iv) Now put '2' on the top '3'



v) Put '1' on the top '2'



Goal State.

Natural language processing :-

we use English language to communication between an intelligent system & Natural language processing.

Components of NLP :- Basically there are two component of

NLP system :-

① Natural language understanding :-

Basically the mapping to given i/p in NL into useful representation.

Analysing the different aspects of the language.

② Natural language generation :-

we have to produce meaningful phrases & sentence. That is in the form of NL from internal representation.

Text planning

Sentence planning

Text realization.

Application of NLP :-

- ① Speech recognition
- ② Sentimental Analysis
- ③ Machine translation
- ④ Chat bots



fig :- NLP.