7. Local Beam Search

In this algorithm, it hold k no of states at any given time. At the start, these states are generated randomly. The successors of these k states are computed with the help of objective function. If any of these successors is the maximum value of the objective function, then the algorithm stops. otherswise the (intial k states and k no of successors of the states = 2k) states are placed in a pool. The pool is then sorted numerically. The highest k states are selected as new initial states. This process continues until a maximum value is reached

function BeamSearch (problem, k) return a solution stat
start with k randomly generated states
loop
  generate all successors of all k states
  if any of the states = solution, then return the state
  else select the k best successors
end

190031187

9. Minimax is a kind of back tracking algorithm that is used in decision making and game theory to find the optimal move for a player, assuming that your opponent plays optimally. It is widely used in two player games such as Tic-Tac-Toe, chess etc

In Minimax the two players are called maximizer and minimizer. The maximizer tries to get to highest score possible while the minimizer tries to do the opposite and get the lowest score possible

function MINIMAX-DECISION (state) return an action

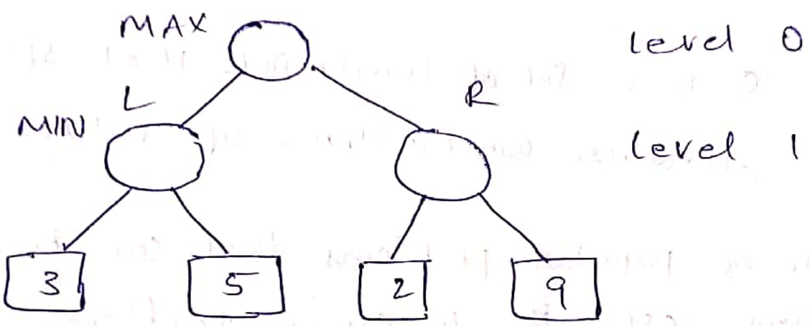
return arg $max_a$ ∈ Actions MIN-VALUE (RESULT(state,a))

function MAX-VALUE (state) returns a utility value
if TERMINAL-TEST (state) then return UTILITY(state)
$v \leftarrow -\infty$
for each a in ACTIONS (state) do
$v \leftarrow$ MAX (v, MIN-VALUE (RESULT (s,a)) )
return v

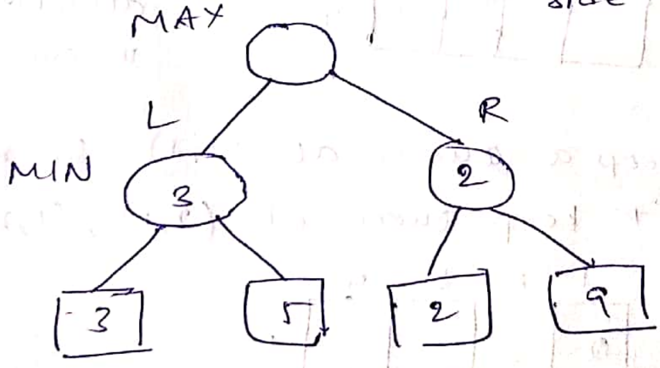
function MIN-VALUE( state) returns utility value
if TERMINA-TEST (state) thenreturn UTILITY (state)
$v \leftarrow \infty$
for each a in ACTIONS (state) do
$v \leftarrow$ MIN (v, MAX-VALUE (RESULT (s,a)) )
retun v

Example:

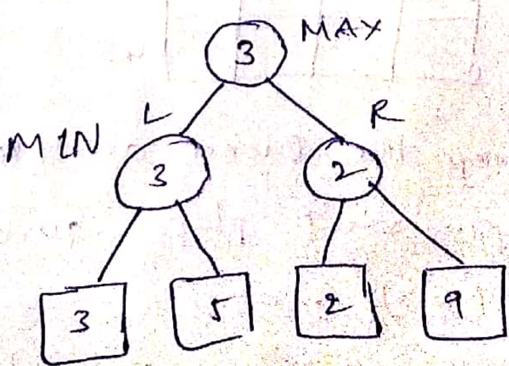
Consider a game which has 4 final states and paths to reach final state are from root to 4 leaves of a perfect binary tree as shown. Assume you are the maximizing player and you get the first chance to move i.e, you are the root and your opponent at next level.




out of 3 & 5 3 is min so in left side Level 1 3 is filled in similar way on right side: Out of 2 & 9 2 is Min




out of 3 & 2 3 is Max so in root 3 is filled




| | |
|---|---|
| Time | $O(b^d)$ |
| space | $O(db)$ |

190031187

11. A Constraint satisfaction problem is a problem that requires its solution within some limitations also known as constraints.
It consists of the following

X is a set of variables

D is a set of Domains (one for each variable)

C is a set of constraints that specify allowable combinations of values

One of popular problems that can be solved using CSP is 4-Queen problem

1.

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | Q | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |

Constraint

4 Queens must be placed in non attacking positions in every row

First keep a queen at (1,1) so now we can't keep queen at (2,1), (1,2), (2,2)

2.

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | Q | | | |
| 2 | | | Q | |
| 3 | | | | |
| 4 | | | | |

Now keep the Queen at (3,2)
Now you can't place queens at (3,3) (2,3), (4,3), ... etc

Hence

our $1^{st}$ move is wrong

Again start from $1^{st}$

Now keep Queen at (2,1)

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | X | Q | X | |
| 2 | X | X | X | |
| 3 | | | | |
| 4 | | | | |

Now keep another Queen at (4,2)

| | | | |
|---|---|---|---|
| | Q | X | X |
| | | X | Q |
| | | X | X |
| | | | |

Now keep third queen at (1,3)

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | | Q | | |
| 2 | | | | Q |
| 3 | Q | | | |
| 4 | | | | |

Now one more Queen at (3,4)

| | | | |
|---|---|---|---|
| | Q | | |
| | | | Q |
| Q | | | |
| | | Q | |

Hence solved

190031187

12. $\;\;\; {}^{C_2 C_1}$

$\;\;\;\;\; {}_{C_3}E\,A\,T$

${}^{C_4}\;\; 9\,T\,H\,A\,T$

---

A PPLE

$\{0,1,2,3,4,5,7,8,9\}$

$c \in (0,1)$

| | |
|---|---|
| E | 8 |
| A | 1 |
| T | 9 |
| H | 2 |
| P | 0 |
| L | 3 |

$T + T = E + 10 \times C_1$

$A + A = L + 10 \times C_2$

$E + H = P + 10 \times C_3$

$C_3 + T = P + 10 \times C_4$

$C_4 = A$

leading value cannot be 0

Hence $\boxed{A = 1}$

$$\begin{array}{r} {}^{1}\;\;\; 8\;1\;9 \\ {}^{1}\;9\;2\;1\;9 \\ \hline 1\;0\;0\;3\;8 \\ \hline \end{array}$$

if $C_1 = 1$

$C_1 + A + A = L$

$1 + 1 + 1 = L$

$\boxed{L = 3}$ $\Rightarrow \boxed{C_2 = 0}$

For $C_4$ to be 1 $C_3 + T = P$ must be $\geq 10$

$\boxed{T = 9}$ $C_3 = 1$

$\boxed{P = 0}$

$E + H = P + 10$

$E + H = 10$

$\Rightarrow E = 8,\; H = 2$