Unit·2

: Introduction to Search :-

-: Topic :-

Searching for Solutions, Uninformed Search strategies, Informed Search strategies, Local search algorithms and optimistic problems, Adversarial Search, Search for games, Alpha-Beta pruning.

Searching for solution :-

Search: Searching is a step by step procedure to solve a search-Problem in a given search space. A search problem can have three main factors:-

a) **Search space :-** Search space represents a set of possible solutions, which a system may have.

b) **Start state :** It is a state from where agent begins the search.

c) **Goal test :-** It is a function which observe the current state and returns whether the goal state is achieved or not.

Search tree :- A tree representation of search problem is called Search tree. The root of the Search tree is the root node which is corresponding to the initial state.

Properties of Search Algorithms:

Following are the four essential properties of Search algorithms to compare the efficiency of these algorithms:

Completeness :- A search algorithm is said to be complete if it gurantees to return a solution if at least any solutions exits for any random input.

Optimality :- If a solution found for an algorithm is guranteed to be the best solution (lowest path cost) among all other solutions, then such a solution for is said to be an optimal solution.

Time Complexity - Time complexity is a measure of time for an algorithm to complete its task.

Space Complexity:-

It is the maximum storage space required at any point during the search, as the complexity of the problem.

Types of Search algorithms:-

Based on the search problems we can classify the search algorithms into uninformed (Blind Search) search and informed search (Heuristic search) algorithms.

Search Algorithm

- Uninformed Search
  - Types of Example
    - Breath first Search
    - Depth first Search
- Informed Search
  - Best first search
  - A^* Algorithm
  - AO^* Algorithm
  - Hill Climbing
  - Constraint Satisfaction

*** Uninformed Search:-**

The uninformed search does not contain any domain knowledge such as closeness, the location of the goal. It operates in a brute-force way as it only includes information about how to traverse the tree and how to idenify leaf and goal nodes. It examines each node of the tree untill it achieves the goal node.

*** Informed Search:-**

Informed search algorithms use domain knowledge. Informed search strategies can find a solution more efficiently than an uninformed search strategy. Informed search is also called a Heuristic search.

A heuristic is a way which might not always be guaranteed for best solutions but guaranteed to find a good solution in reasonable time.

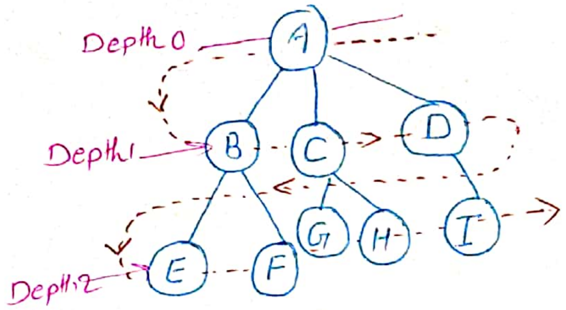
Breadth-First-Search:-

* It is used Uninformed Search Technique.

It starts from the root node, explores the neighboring nodes first and moves towards the next level neighbors. It generates one tree at a time untill the solution is found.

* It can be implemented using FIFO (Queue) datastructure.

Example:



Insert the first element in the Queue

← | A | ←

Open A in the Queue

← | B C D | ←

again Open B in the Queue

← | C D E F | ←

Next Again open C in the Queue

← | D E F G H | ←

Next Again open D in the Queue

← | E F G H I | ←

Now Again open E in the Queue

← | F G H I | ←

Now Again open F in the Queue

← | G H I | ←

* Breath first search always gives the knowlege of the shallowest node because it computes always Level wise search. Means it Computes the nearest node and DFS Performs depth level Searching Technique.

* Complete:- It will definitely give an answer because here we are going to complete all the levels one by one, so it will be called complete. This means it will answer.

* Optimal:- What is optimal?
Optimal means it will give us the shortest path. Suppose we keep the cost of all edges the same, then it will give us the shortest path. But even if we assign different cost it will also give shortest path.

* Time Complexity

BFS Time complexity is represented by

$O(b^d)$

where b is denoted branch factor and d is denoted depth

How many children will a particular node, we call it branch factor.
Moving from root node to n node is called depth.

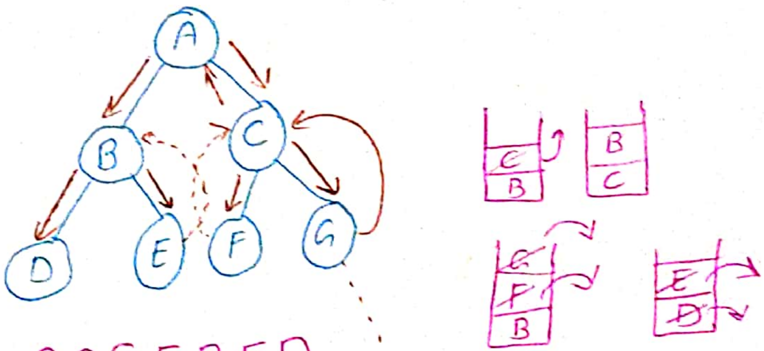
Ex Suppose that Search G

Where b = 3 $3^2 = 9$
d = 2

Total Consumes 9 Search to find the goal node of G.

* Depth Search tree :-
  * It is also used uninformed Search Technique.
  * It can be implemented using Stack (LIFO) data structure.

Example:



ACGFBED

* Deepest Node:- It will move in a single direction, so it will search like a deepest node.
* Incomplete:- Incomplete means that it is possible to not give us this result. Because if we are cycling in our graph or searching infinite in the same direction, then it will not give results.
* Non-optimal:- Non optimal means it will not provide the shortest path. Because they give different solutions, It means that the cost of getting a solution is higher and the cost of the next solution is less than that. Therefore, they are non-optimal.

Informed Search strategies:-

Local Search Algorithms and optimistic Problems:-

Local Search algorithms are used, in such types of algorithm, a single instead of multiple Paths and generally move only to neighbors of the state. Local search algorithms are not Systematic, they have two important advantages.

* Memory requirement is very Low.
* Find the optimal path in large or infinite state spaces.

Local Search Can be classified in to following categories:-

① Best First Search
② A* Algorithm
③ AO* Algorithm
④ Simple hill Climbing
⑤ Steepest ascent hill Climbing
⑥ Constraint Satisfaction

* Best First Search:-

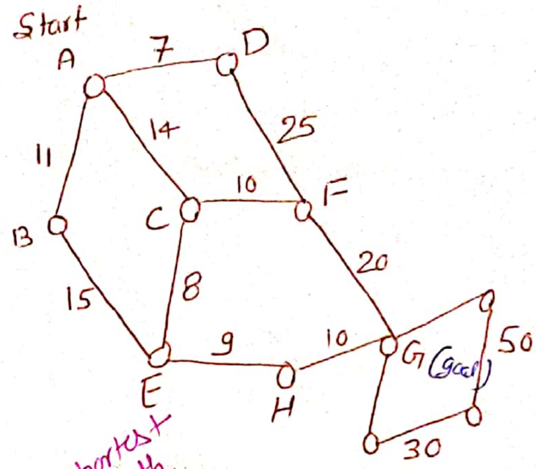
Best first search is the combination of Depth first Search and Breadth first Search. It is used to Informed Seach technique. The best first Search allows us to switch paths thus gaining the benefit of both approaches. At each step the most promising node is chosen.

Algorithm:-

→ Let 'Open' be a priority queue containing initial state.

Loop
 if Open is empty return failure
 Node ← Remove - First (Open)
  if Node is a Goal
  then return the path from initial to Node
 else generate all successors of Node and Put the newly generated Node into Open according to their f values.
 ↓
 Heuristic
 (Means it defined Less cost path or shortest distance path)
END Loop.

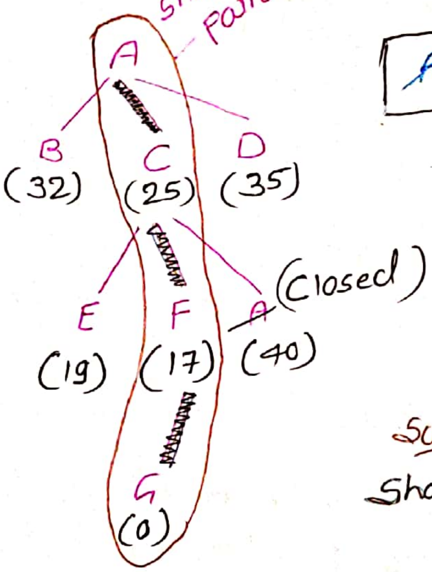
Example:



(Heuristic value)
Straight line distance

$A \to G = 40$
$B \to G = 32$
$C \to G = 25$
$D \to G = 35$
$E \to G = 19$
$F \to G = 17$
$H \to G = 10$
$G \to G = 0$

Sol^n.

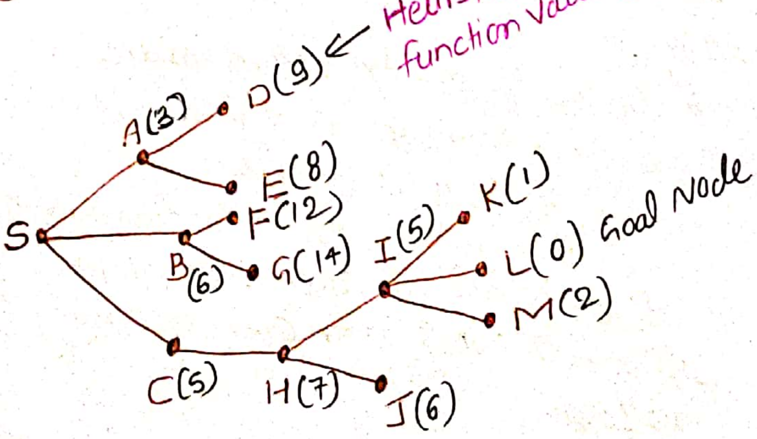


~~A~~ ~~C~~, B, D, ~~F~~, E, G

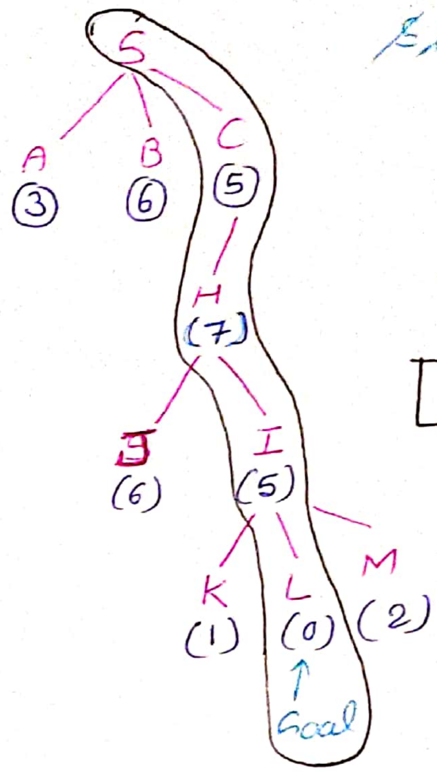
Sol^n
Shortest path — $A \to C \to F \to G$

* The best search method always works on greedy method. greedy is related to heuristic value which means the lower my heuristic value, I will take it with my local maximum. Means, I will explore the same path and will not explore the rest. The best search method has low time complexity, and it always gives good solutions and not optimal solutions.

another Example:



~~S~~, ~~A~~, ~~C~~, B, ~~H~~, I, J, K, L M

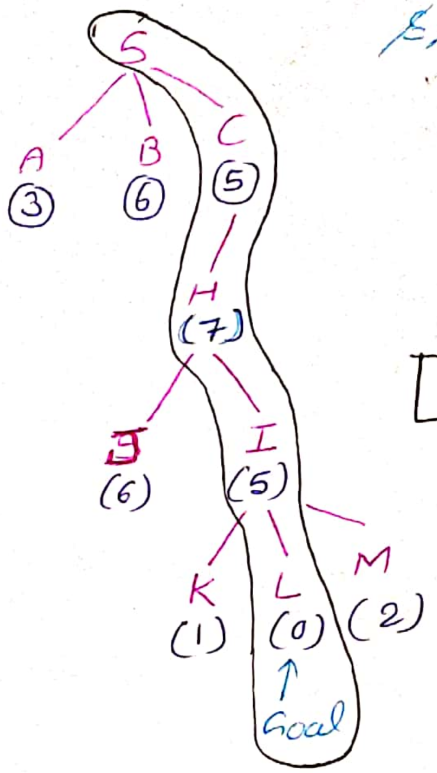


Shortest path is

$\boxed{S \to C \to H \to I \to L}$

A* Algorithm :-

~~S~~, ~~A~~, ~~C~~, B, ~~H~~, I, J, K, L M



Shortest path is

$$\boxed{S \to C \to H \to I \to L}$$

A* Algorithm :-

It is used informed Search Technique.

A* is Admisiable Algorithm that means they are directly gives a guorontee that we will give optimal solution anywhere.

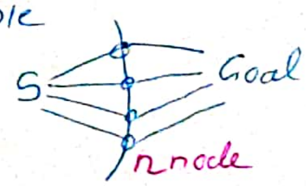
A* can be used whether we are intersted in finding Minimum cost over all path 'or' simply any path as quickly as possible.

$$\boxed{f(n) = g(n) + h(n)}$$

$g(n)$ → Actual cost from start node to n node

$h(n)$ → Estimation cost from n to Goal node

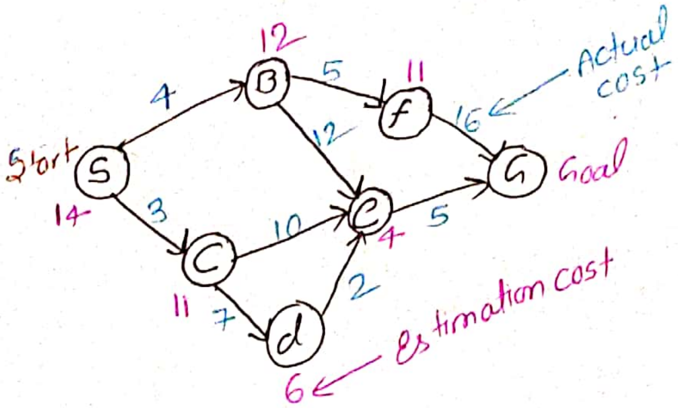
Example



It is best known form of best first Search It avoids Expanding paths that are already Expensive, but Expands most promising paths first.

Example:-

Example:-



Soln

S → S

S → g(n) + h(n)

0 + 14 = 14

S → B   S → C → choose Minimum path

S – 4+12 = 16   3+11 = 14

SC → d   SC → e

3+7+6 = 16   3+10+4 = 17

~~SCd~~ S to B & SC to d are same minimum path cost provided

SB → e   SB → f

4+12+4 = 20   4+5+11 = 20

Scd → e

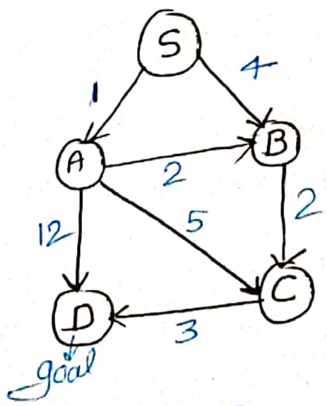
3+7+2+4 = 16

scde → G

3+7+2+5+0 = 17

So The best path is ScdeG → optimal solution.

Example:



heuristic function value

S	7
A	6
B	2
C	1
D	0

S → A   S → B

1+6 = 7   4+2 = 6

S → S

S = g(n) + h(n)

0 + ~~7~~ = ~~7~~ 7

SB → C

4+2+1 = 7

S → A & SB → C are same cost path

$SA \rightarrow B$ $\quad$ $SA \rightarrow C$ $\quad$ $SA \rightarrow D$

$1+2+2=5$ $\quad$ $1+5+1=7$ $\quad$ $1+12+0=13$

$SAB \rightarrow C$

$1+2+2+1=6$

$SABC \rightarrow D$

$1+2+2+3+0=8$

$SB \rightarrow C$ & $SA \rightarrow C$ both are same path cost

$SB \rightarrow C$ $\quad$ $SA \rightarrow C$

$\downarrow$

$SBC \rightarrow D$ $\quad$ $SAC \rightarrow D$

$4+2+3+0=9$ $\quad$ $1+5+3+0=9$

So the best path is SABCD → optimal solution.

AO* Algorithm (informed, Heuristic Method)

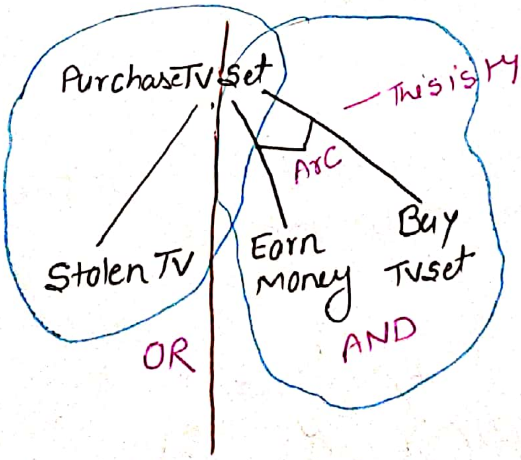
→ This is known as AND-OR-Graph (Algorithm)

→ They are based on problem Decomposition Approach. Means the complex problem is divided into a set of subproblems, where Each sub-problem can be solved seperately.

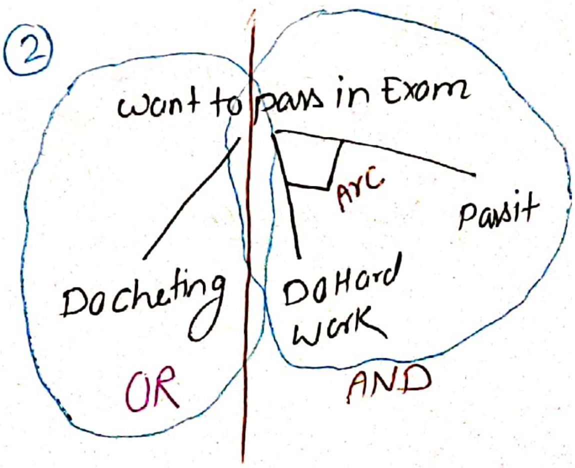
→ AO* does not replace all the solution paths once it got a solution.

Example:-

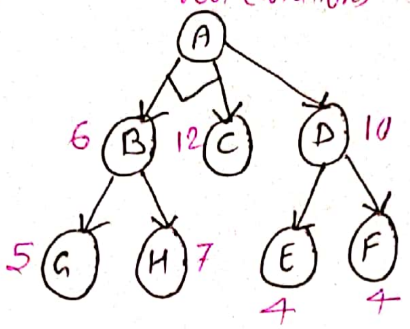
①



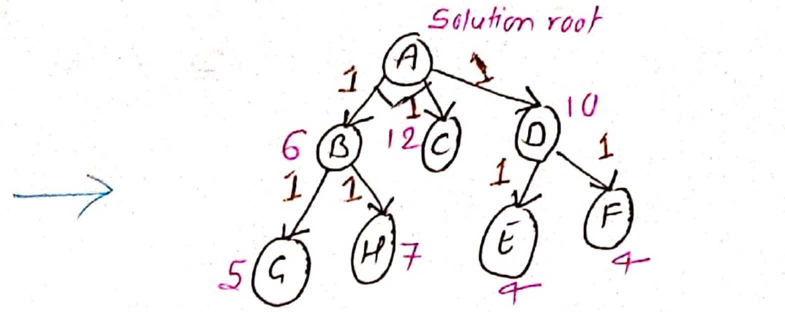
②



Example

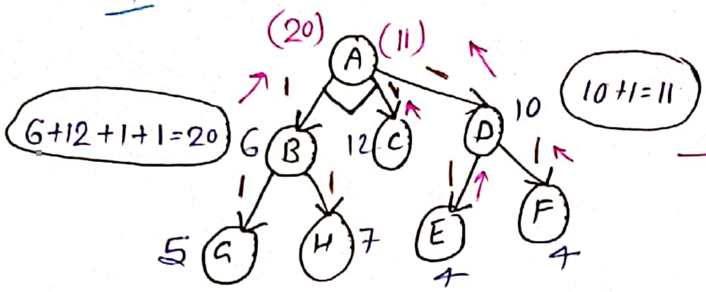


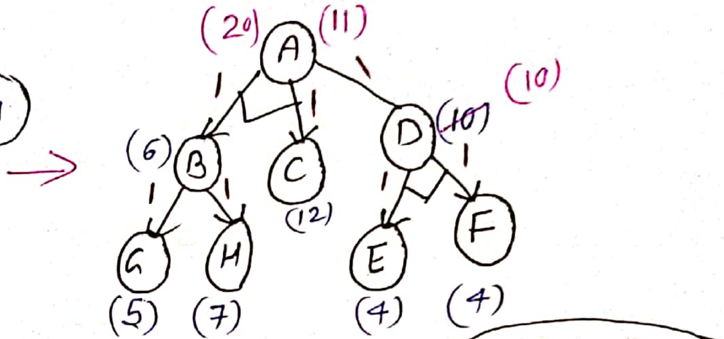
(Problem)



where all edges cost (Actual) is = 1

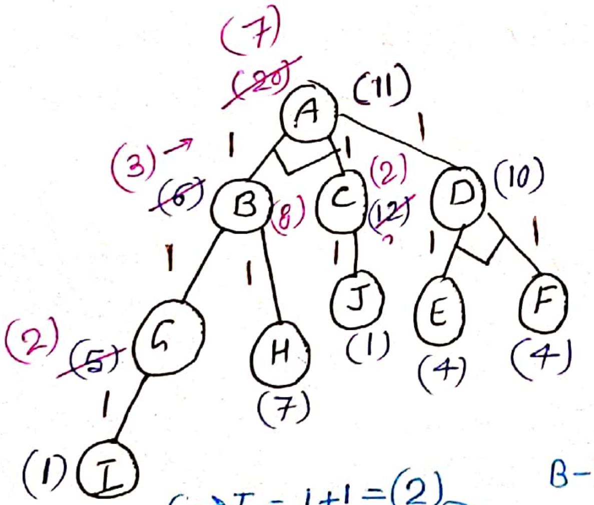
Sol^n





4+4+1+1=10

(get the solution one side so next side not Explore)

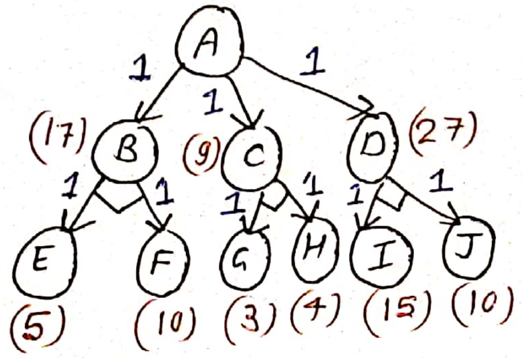


$G \to I = 1+1 = (2)$ $B-H \to 7+1=8$ $A \to BC \to 3+2+1+1=7$

$B \to G = 2+1 = 3$ (Choose) $C \to J = 1+1=2$

(Explore the next side again)

Next question.



Where all edges cost (Actual) = 1

Scanned with CamScanner

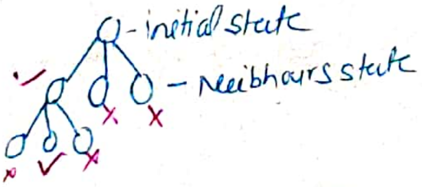
Hill Climbing:-

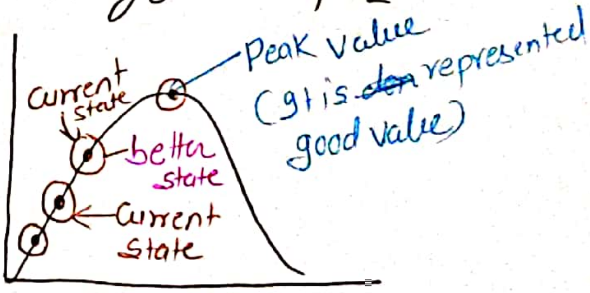
* It is also known as Local search Algorithm. means It has knowlege of its local domain, it does not have knowlege of the entire global domain or global problem.
* It uses greedy Approach, it will run till the time the best move is found. If the best move stops, then it stops at the same time.
* It does not use the backtracking Approach.

Algorithm:-

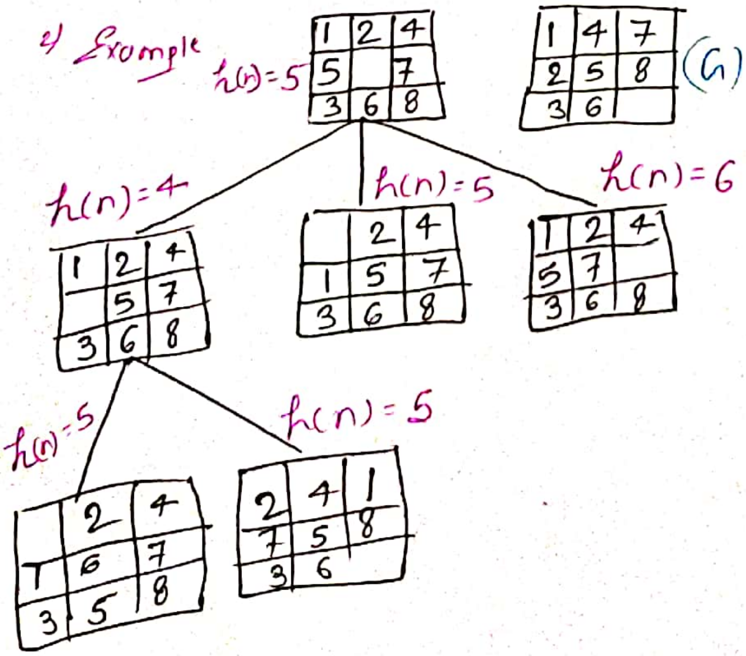
1. Evaluate the initial state if it is goal state then return and quit.
2. Loop untill a solution is found or there are no new operators Left.
3. Select and apply new operator
4. Evaluate new state:-
   1. If it is goal state, then quit.
   2. If it is better then current state then make it new current state.
   3. If it is not better then current then go to stop. 2.

1) Example



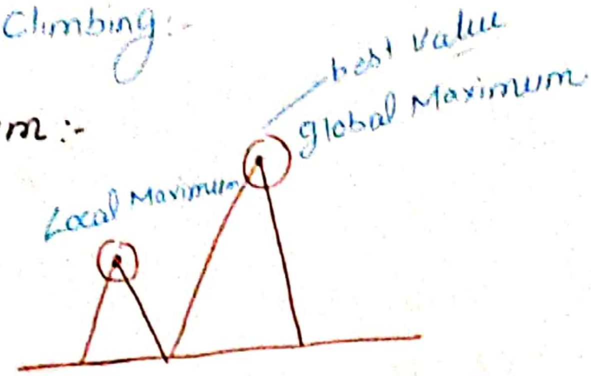


2) Example



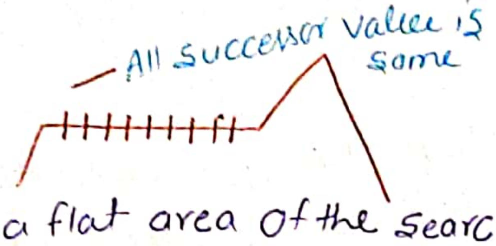
Problem in Hill Climbing :-

① Local Maximum :-



A Local Maximum is a state that is better than all its neighbours but is not better than some other states further away.

② Plateau :

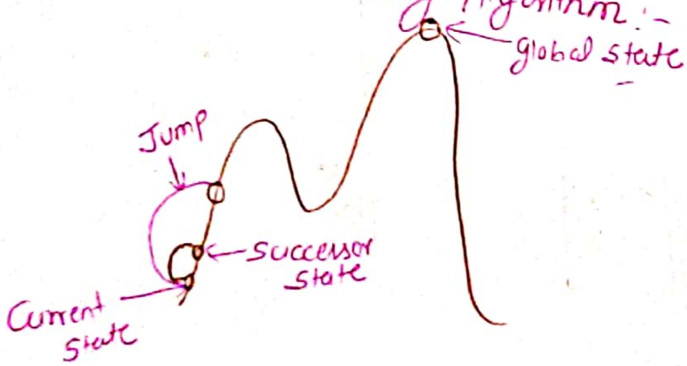


A plateau is a flat area of the search space in which a whole set of neighbouring states have the same value.

③ Ridges :-

If the Landscape contains ridges, local improvements may follow a zigzag path up the ridge, slowing down the search.

Steepest Ascent Hill Climbing Algorithm :- (Self study topic)



Some problem :

Monkey and Banana Problem :-

Monkey is on floor, at door, A block is on floor, at window. Banana is hanging from roof or ceiling at the middle of the room. problem is " How the monky can get the banana".

Solving the Monkey Banana Problem behalf of proper planning.

Procedures of step by step:-

1. After several unsuccessful attempts to reach the bananas.
2. the monkey walks to the box.
3. Pushes it under the bananas.
4. Climbs on the box.
5. Picks the bananas and eats them.

8-Puzzle Problem without Heuristic:-

→ Blind Search (uninformed)

→ BFS

→ $O(b^d)$

→ 4 moves (UP, down, Left, right)

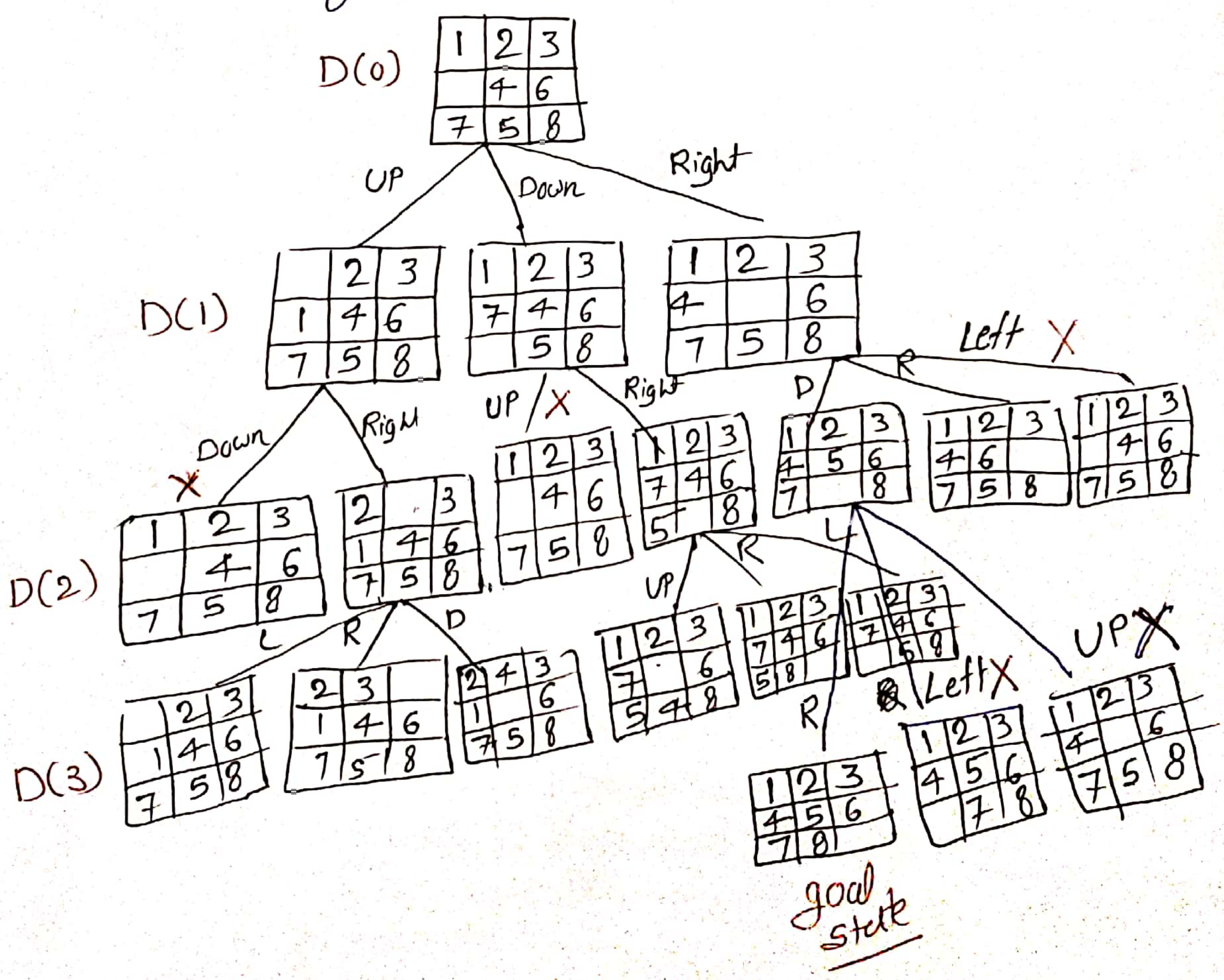
1	2	3
	4	6
7	5	8

Start state

1	2	3
4	5	6
7	8	

Goal state

Solve the this problem using agent. Agent Explore the all state space.



8 Puzzles problems with Heuristic :-

- Informed Search:-
- 4-Moves (UP, down, Left, right) with Heuristic function

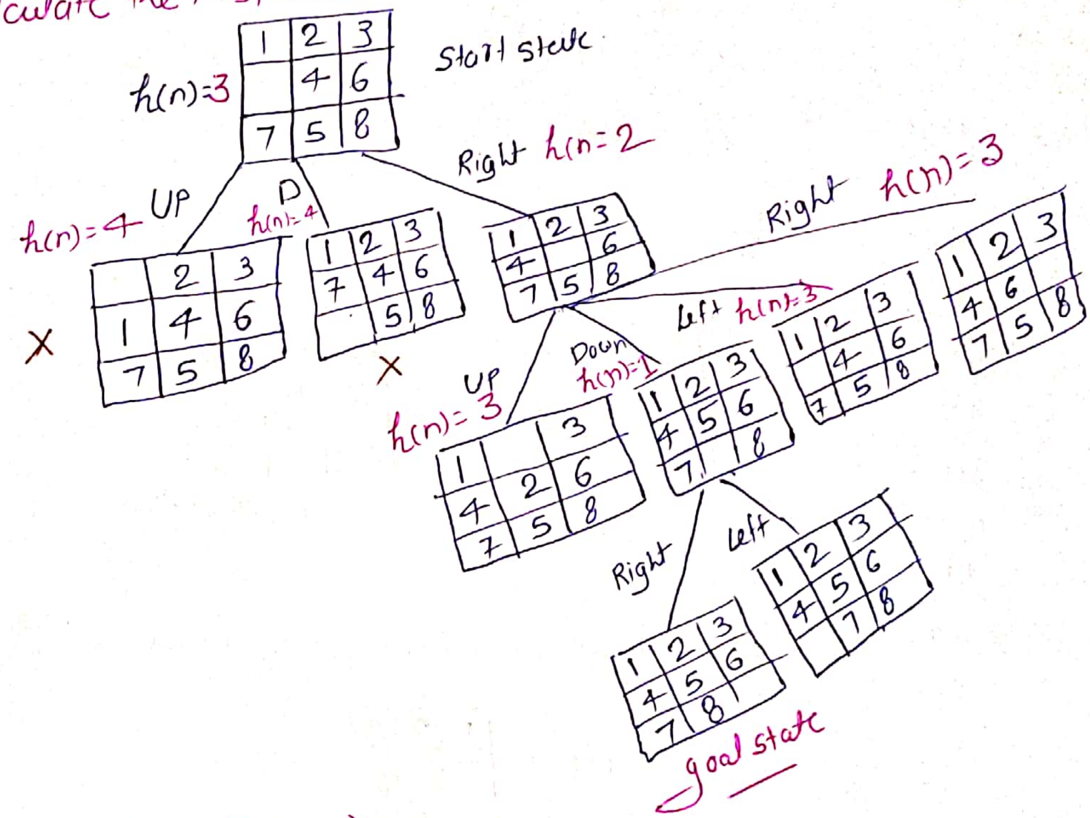
Start State

1	2	3
	4	6
7	5	8

Goal State

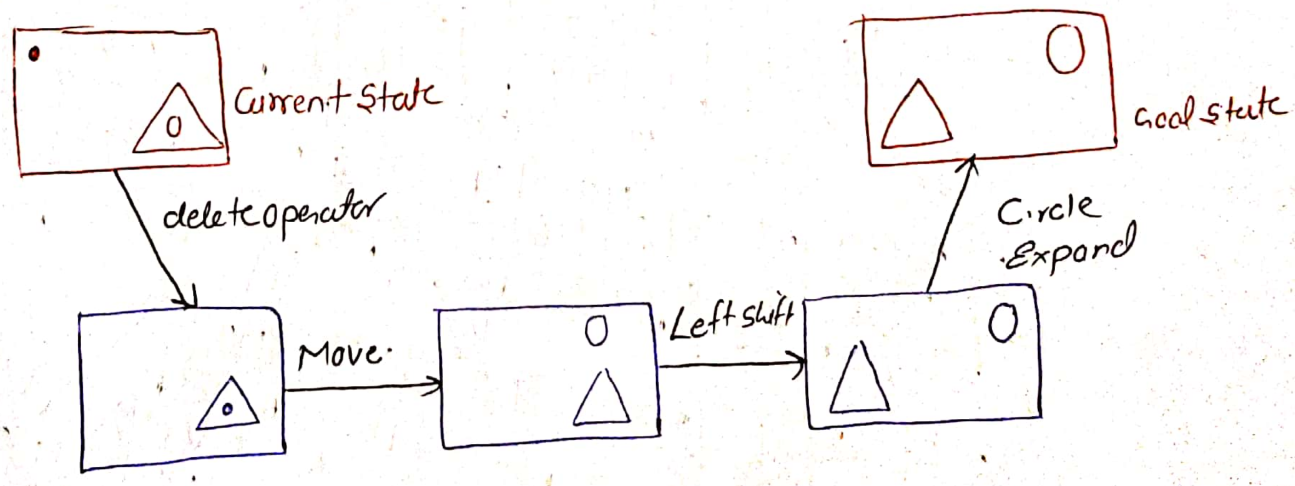
1	2	3
4	5	6
7	8	

$h(n)$ = Calculate the Misplace tiles position.



Means Ends Analysis (MEA)

Example

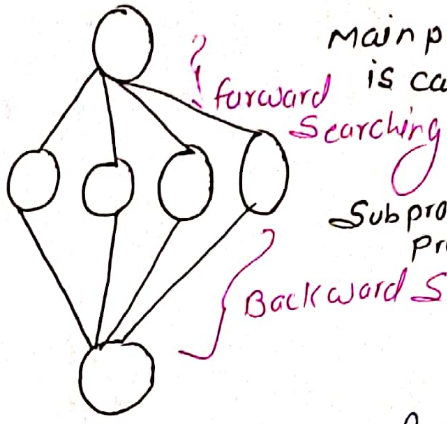


Operation:

1. First of all Remove the Dot symbol
2. Removing the Circle from the triangle and bringing it over the triangle.
3. Bringing the triangle to the Left
4. then after Expand the Circle.

Mean ends analysis:-

1. Means ends analysis is a problem solving technique.
2. Allows both backward and forward searching.



Main problem is solving into sub problem is called Forward searching

Subproblem is compute into a single problem is called Backward Searching.

3. Computing the current state to a goal state by compute their difference (Findout the different types of difference)
4. To reduce the difference between current and goal state we will use operators.

Algorithm:-

To perform means-ends analysis,

1) Untill the goal is reached or no more procedures are available,
   - Describe the current state, the goal state, and the diffence between the two.
   - Use the difference between the current state and goal state, possibly with the description of the current state or goal state, to select a promising procedure.
   - Use the promising procedure and update the current state.

2) If the goal is reached, announce success; otherwise, announce failure.

Problem:- Apply means-ends analysis to solve the monkeys and bananas problem.

Constraint Satisfaction:-

→ CSP Consists of three components V, D, C.

→ V is set of variables $\{V_1, V_2 \cdots V_n\}$, finite variable

→ D is set of Domains (finite) $\{D_1, D_2, D_3 \cdots D_n\}$ one for Each Variable

→ C is set of constrainsts that specify allowable Combination of values. $\{C_1, C_2, C_3\}$

$$\boxed{C_i = (\text{scope}, \text{rel})}$$

→ Where scope is set of variables that participate in constraint.

→ rel is relation that defines the values that variable can take.

Example: consider two variable V_1, V_2 and Domain A, B so V_1 Consider the value of A Domain and V_2 consider the value of B Domain But Constrainsts specify the rule V_1 & V_2 value are not same.

$$\begin{array}{cc} V_1 & V_2 \\ A & B \end{array}$$

Representation of Constraints

$$C_1 = (\underbrace{(V_1, V_2)}_{\text{scope}}, \underbrace{(V_1 \neq V_2)}_{\text{relation}})$$

another forms.

$$C_1 = ((V_1, V_2), (A, B))$$

But

$$\begin{array}{cc} V_1 & V_2 \\ A & B \\ (1,2) & (2,4) \end{array}$$

$$C_1 = ((V_1 V_2), (1,2)(1,4)(2,4))$$

Example:-

1. N-Queen Problem
2. A Map coloring problem
3. Boolean satisfiability problem
4. Cryptarithmetic problem

Booleon Satisfiability Problem (SAT)

Boolean formula:

Variables x_1, x_2, x_3 { "true"/1, "false"/0 }

"not" $\bar{x}_1\ \bar{x}_2$ (x_1 = true $\Rightarrow$ $\bar{x}_1$ = false)

"and" $(x_1 \wedge x_2)$ $0 \wedge 0 = 0$, $1 \wedge 0 = 0$, $0 \wedge 1 = 0$, $1 \wedge 1 = 1$

'or' $(x_1 \vee x_2)$ $0 \vee 0 = 0$, $1 \vee 0 = 1$, $0 \vee 1 = 1$, $1 \vee 1 = 1$

An instance of the problem is a Booleon Expression written using only 'AND', 'OR', NOT, variables and parentheses.

Example $((x_1 \vee \bar{x}_2) \vee x_3) \wedge (\bar{x}_1 \vee x_2)$ $\Rightarrow$ where x_1 = true(1), x_2 = true(1) and x_3 = true(1)

Sol^n

$((1 \vee 0) \vee x_3) \wedge (\bar{x}_1 \vee x_2)$

$(1 \vee 1) \wedge (\bar{x}_1 \vee x_2)$

$1 \wedge (0 \vee 1)$

$1 \wedge (1)$

$1 \wedge 1 = 1$ (true)

So answer is false or true '0','1'

In mathematics a formula of propositional logic is said to be satisfiable if truth-values can be assigned to its variable in uses that makes the formula true.

SAT is of central importance in various areas of computer science, including theoretical computer science, algorithms, artificial intelligence, hardware design and verification.

Cryptarithmetic Problem:

Cryptarithmetic problem is the science and art of creating and solving cryptarithms. This is the example of constraint satisfaction. Crypt-arithm is a type of mathematical puzzle in which the digits are replaced by alphabets or other symbols.

The following Crypt-arithmetic conventions are to be considered for solving the problem in hand.

(1) Each letter or symbol represents only one digit throughout the problem.

(2) When letters are replaced by their digits, the resultand arithmetical operations must be correct.

(3) The numerical base, unless specifically stated is 10,

(4) Numbers must not begin with a zero.

Example:-

(1)

$$\begin{array}{r} T\ O \\ +\ G\ O \\ \hline O\ U\ T \end{array} \qquad \begin{array}{r} \boxed{2}\ \boxed{1} \\ \boxed{8}\ \boxed{1} \\ \hline \boxed{1}\ \boxed{0}\ \boxed{2} \end{array}$$

Letter	digit
T	2
O	1
G	8
U	0

Solve the Left Most = 1

$2+G=U$

$2+8=10 \rightarrow U(0)$

(2)

$$\begin{array}{r} S\ E\ N\ D \\ M\ O\ R\ E \\ \hline M\ O\ N\ E\ Y \end{array} \qquad \begin{array}{r} C_4\ \ C_3\ \ C_2\ \ C_1 \\ \boxed{9}\ \boxed{5}\ \boxed{6}\ \boxed{7} \\ \boxed{1}\ \boxed{0}\ \boxed{8}\ \boxed{5} \\ \hline \boxed{1}\ \boxed{0}\ \boxed{6}\ \boxed{5}\ \boxed{2} \end{array}$$

Letter	digit
S	9
E	5
N	6
D	7
M	1
O	0
R	8
Y	2

Solve the Left Most = 1

(1) $S+M='O'$

$S+1='O'$

$9+1=10$ 'O'

↓

5

(2) $E+\overset{\text{zero}}{0}=N$

$E=N$ (unique) } if $C_2=0$ where E & N value are same

$C_2=1$ Let C_2 Carry equal to 1

Let $E=5$

$E+0+C_2=N$

$E+1=N$

$5+1=N$

$N=6$

(3) $N+R=E$

$\underbrace{6+R=5}$ Let $R=9$

$6+9=15$ where 9 given to S so it is not possible.

$6+8+1=15$

↳ R

(4) $D+E=Y$

$D+5=Y$

$D>5$ as 5 to bigger

$7+5=Y$

$12=Y$ — 2, 1 Carry

(3) B A S E (with ③ ② above S and E)

$$
\begin{array}{r}
BASE \\
+\ BALL \\
\hline
GAMES
\end{array}
\qquad
\begin{array}{r}
\boxed{7}\ \boxed{4}\ \boxed{8}\ \boxed{3} \\
\boxed{7}\ \boxed{4}\ \boxed{5}\ \boxed{5} \\
\hline
\boxed{1}\ \boxed{4}\ \boxed{9}\ \boxed{3}\ \boxed{8}
\end{array}
$$

B – 7
A – 4
S – 8
E – 3
L – 5
G – 1
M – 9

(1) $G=1$

(2) $E+L=S$ — ⓐ (no carry)

$E+L=S+10$ — ⓑ (with carry)

$E=S-L+10$

$S+L=E$

$\cancel{S}+L=\cancel{S}-L+10$

$2L=10$

$L=5$

③ $S-E=L$

or $S-E=L$

or $S-E=5$

Put the value of S & E

$(S,E)=(5,0)$ X

$(6,1)$ X

$(7,2)$ X

$(8,3)$

$(9,4)$

Put the value of S & E (5,0)

$5-0=5$

└ This value also assign L (X)

Next put the value of S & E (6,1)

$6-1=5$

└ 1 is also assign (G) X

So Now Let put the value of (S,E) (7,2) and B=6

$B+B=A$

$6+6=12$ — A, 1 — G

So $S-E=5$

$7-2=5$

└ so 2 are not assign E because 2 are also assign A

So Now let again put the value of (S,E) = (8,3) and B=7

$B+B=A$

$7+7=14$ — A, 1 — G

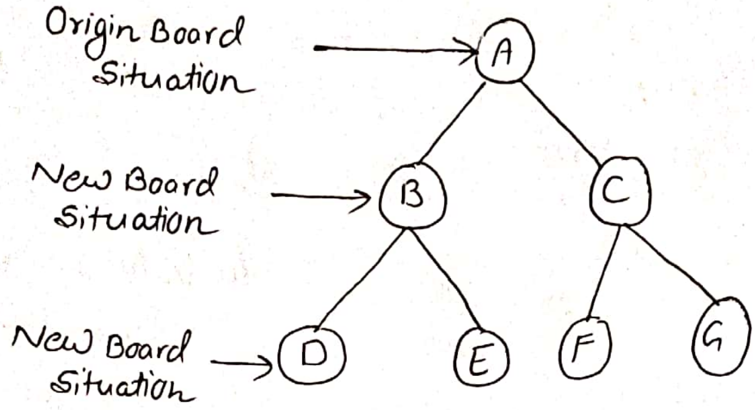
So Now put S=8 & E=3 & B=7 & A=4

④ So Now Let go $B+B=A$ + carry

Put the value of B — 5 (X), 6, 7, 8, 9

Adversarial Search:-

Searches in which two or more players with contrary goals are trying to Explore the same solution space in search of the solution are called Adversarial searches.



The node is a game tree represent board configuration, and the branches indicate how moves can connect them.

games are used in Adversarial search:-

The basic Methods available for Game playing are:-

1. Minimax strategy
2. Minimax strategy with alpha-beta cutoffs or Pruning.

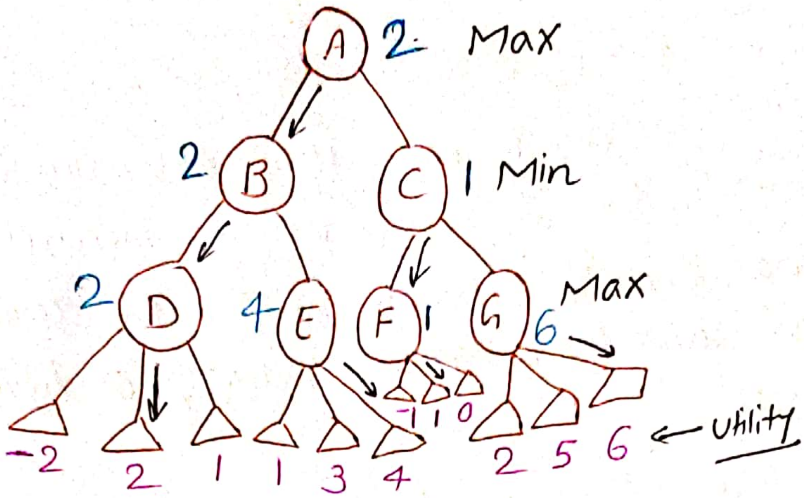
Minimax strategy:-

Minimax is a simple strategy for two-person game playing. They are using Backtracking Algorithm & best move strategy. Where one player is called a maximizer and other is called Minimizer.

1) Where Max will try to maximize its utility (Best move)
2) & Min will try to minimize utility (worst case)

Means both the players, maximizer and minimizer fight it out to see to that the opponent gets the minimum benifit while they get the maximum benefit.

It is applied in two player games like chess, checkers, tic-tac-toe, etc.



Demerits: Each node are Search and compared to all Each Other and this problem is managed to Alpha - Beta cutoffs.

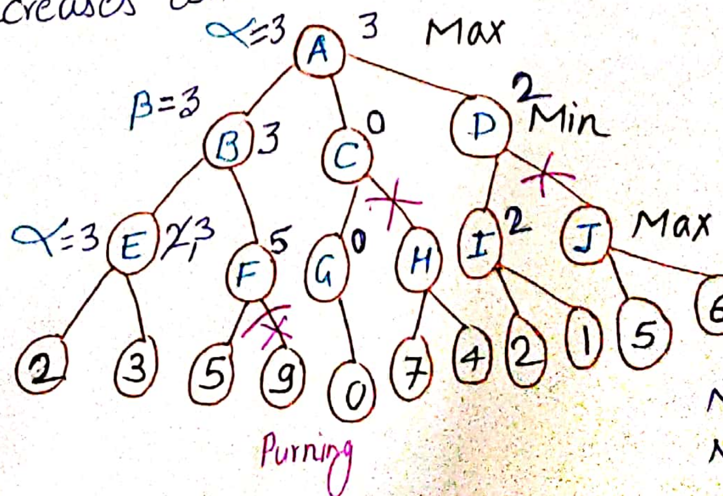
Alpha - Beta Pruning:-

Alpha - beta pruning is a method that reduces the number of nodes Explored in Minimax strategy. It reduces the time required for the search. The Exact implementation of alpha-beta keeps track of the best move for Each side as it moves through out the tree.

Where α & β are two values α is assign Max node & β is assign Min node.

For the min nodes $(+\infty)$ the score computed starts with +infinity and decreases with time.

For Max nodes, scores computed starts with -infinity $(-\infty)$ and increases with time.



No of α cut off = 1

No of β cut off = 2

Search For Games:-

There several reasons for game-playing in AI:-

1. The rules of the game are limited. Hence Extensive amounts of domain-specific Knowlege are celdom needed.
2. Many human Experts exist to assist in the developing of the programs. Hence the problem of paucity of human Experts does not arise.
3. Games provide a structured task wherein success or failure can be measured with least Effort.
4. For the Human Expert, it is easy to Explain the rationoble for a move unlike other domains.
5. The logical reasoning ability of the human in a normal condition and under stress is clearly Exhibited in game-playing. Moreover, game-playing permits one to simulate real-life situations.