Artificial Intelligence (AI)

Assignment - 1

Name:- Ankit Kr. Srivastava
Roll No:- 00514823121
Class :- 6 ITE

Q.1 Explain Iterative Deepening Depth first Search with example.

Ans DFID (Depth-first iterative deepening) is a graph traversal algorithm that combines the advantage of both Depth-first Search (DFS) and Breadth-first search (BFS) algorithms. It is an uninformed search Algorithm, which means that it doesn't use any heuristic information to guide its search.

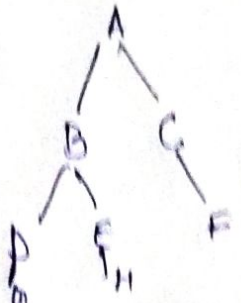
Algorithm:-

Input : START and GOAL states
Local variables : found;
Output : yes or no
Method :

- Initialized d = 1 /* depth of the search tree */
  found = False;
- while (FOUND = false)
  do {
  - perform a depth first search from start to depth d.
  - If goal state is obtained then found = true else discard the node generated in the search of depth d
  - d = d + 1

  }
- If found = True then return yes otherwise return no
- Stop.

For example :-



- Starting from the root node A, DFID performs DFS with a depth limit of 1, then with depth limit 2 and so on, until all nodes in the tree have been visited.

- In the first iteration, DFS starts with A with a depth limit of 1. It visits B and C, but doesn't visit their children because they are beyond the depth limit. The algorithm then backtracks to A.

- In the second iteration DFS starts again from A, but with a depth of limit 2. This time it visits B, C, D, E, F. But doesn't visit them because it out of the depth. Then Algorithm backtracks.

- In the final iteration DFS starts from A with a depth limit of 3. It visits all the node in the graph and the algorithm terminate.

Q.2 What is constraint satisfaction? Solve following cryptographic puzzle using constrained satisfaction procedure:

```
  FOUR          SEND
+ NICE        + MORE
------        ------
 FOUND         MONEY
```

Ans Constraint satisfaction refers to the process of finding a solution to a problem where certain constraints must be satisfied. It is the task of finding a combination of values for a set of variable that meets all the requirement of constraint problem.

Q.2.

(a) SEND + MORE = MONEY

$$\Rightarrow \begin{array}{cccccc} C_4 & C_3 & C_2 & C_1 & & \\ & S & E & N & D & \\ + & M & O & R & E & \\ \hline M & O & N & E & Y & \end{array}$$

$M=1$, $C_4=1$.

then, $S+1+C_3 > 9$

$S+C_3 > 8$.

$S=9$

then $C_3=1$

$O=0$.

$E+O+C_2=N$

$E+C_2=N \quad \because (E \neq N)$

therefore $C_2=1$ 0

assume $E=5$

then $N=6$.

$C_1+N+R=E$

$C_1+6+R=15$

let $C=1$, then

$7+R=15$

$R=8$.

$D+5>9$. remained numbers are (2, 3, 4, 7)

so, $D=7$.

$Y=2$.

so,
$$\begin{array}{r} 9567 \\ SEND \\ 1085 \\ +\ MORE \\ \hline MONEY \\ 10652 \end{array}$$

```
 FOUR
+NICE
-----
FOUND
```

$\Rightarrow$

```
  C4  C3  C2  C1
       F   O   U   R
       N   I   C   E
-------------------
   F   O   U   N   D
```

$F = C_4$

$F = 1.$

$C_3 + F + N = 0$

let $C_3 = 0$

then $1 + N = 0$

$N = 9$

so, $O = 8.$

$C_2 + O + I = U$

let $C_2 = 1,$

then

$1 + I = U$

assume $I = 6$

then $U = 7.$

$C_1 + 7 + C = N$

let $C_1 = 1$

then

$8 + C = N$

let $C = 4,$

then $N = 2.$

After that

$R + E > D.$

$R = 8, E = 5$

$D = 3.$

$0 - O$

$1 - F$

$2 - N$

$3 - D$

$4 - C$

$5 - E$

$6 - I$

$7 - U$

$8 - R$

$9 - H$

equation:-

```
  1078
  FOUR
  9645
+ NICE
-------
 FOUND
 10723
```

Q.3 What is Uniform cost Search? Explain with example.

Ans In Uniform cost Search method, cost function is designed that assign cumulative expense to the path from start node to the current node x by applying the sequence of operators.

While generating a search space, a least cost path obtained so far is expanded at each iterations till we reach to goal state.

for example:- in travelling salesman problem, g(x) may be the actual distance travelled from start to current node x. During search process, there are many incomplete paths contending for further consideration. The shortest one is always expanded one level further, these new paths along with old ones and shortest path is always choosen for extension, with the value of cost function 'g', the path first reaching to the goal is certain to be optimal; but it is not guranteed to find the solution quickly.

Algorithm:—

Input: start and goal state
Local variable: open, closed, Node, Succes, found;
output: yes or no
Method:

- Initially store the root node with g(root) = 0 in a open list;
  closed = ϕ ; found = false;
- while (open $\neq \phi$ and found = false) do
  { remove the top element from open list and call it Node;
  if Node is the goal Node, then found = true else

}

put Node in closed list;

- find succes of Node, if any, and compute their g' values and store them in open list;
- store all the nodes in the openlist based on their cost-function value;

}

} /* end while */

- If found = true then return Yes otherwise return No.
- Stop.

Q.4 What is water Jug problem? Explain with an production rules to find a possible solution for water Jug problem.

Problem statement:-

We have two jugs, a 5-gallon (5-g) and the other 3-gallon (3-g) with no measuring marker on them. There is endless supply of water through tap. our task is to get 4 gallon of water in the 5-g jug.

solution:-

State space for this problem can be described as the set of ordered pair of integers (x, y) such that x represents the number of gallons of water in 5-g jug and y for 3-g jug.

production rules for water jug problem.

Rule No.	Left of rule	Right of rule	Description
1	$(x,y \mid x<5)$	$(5,y)$	Fill 5-g jug
2	$(x,y \mid x>0)$	$(0,y)$	Empty 5-g jug
3	$(x,y \mid y<3)$	$(x,3)$	Fill 3-g jug
4	$(x,y \mid y>0)$	$(x,0)$	empty 3-g jug
5	$(x,y \mid x+y \leq 5 \wedge y>0)$	$(x+y,0)$	empty 3-g into 5-g jug
6	$(x,y \mid x+y \leq 3 \wedge x>0)$	$(0,x+y)$	Empty 5-g into 3-g jug
7	$(x,y \mid x+y \geq 5 \wedge y>0)$	$(5, y-(5-x))$	Pour water from 3-g jug into 5-g jug
8	$(x,y \mid x+y \geq 3 \wedge x>0)$	$(x-(3-y),3)$	Pour water 5-g jug until 3-g jug is full.

Solution:-

Rule applied	5-g jug	3-g jug	Step No
Start state	0	0	
1	5	0	1
8	2	3	2
4	2	0	3
6	0	2	4
1	5	2	5
8	4	3	6
Goal state	4	-	