Name: Abhishek Srivastava.
Reg No: 19BCE10071
Subject: Artificial Intelligence.
Slot: C11
Date: 2/09/2021
Class No: 0040

1) Uninformed Search

This strategy have no additional information about states beyond that provided in the problem definition.

All they can do is generate successors and distinguish a goal state from a non-goal state.

All search strategies are distinguished over by the order in which nodes are expanded.

Example: Breadth-first Search
Depth-first Search

Lets discuss in detail about Breadth first search.

Breadth-first Search.

It is a simple strategy in which root node is expanded first, then all the successors of the root node are expanded next, then their successors, and so on.

All nodes are expanded at a given depth in the search tree before any nodes at the next level are expanded.

Algorithm:

0) function Breath-first-search (problem) results a solution, or failure.

1) node ← a node with STATE = problem.INITIAL STATE, PATH-COST = 0.

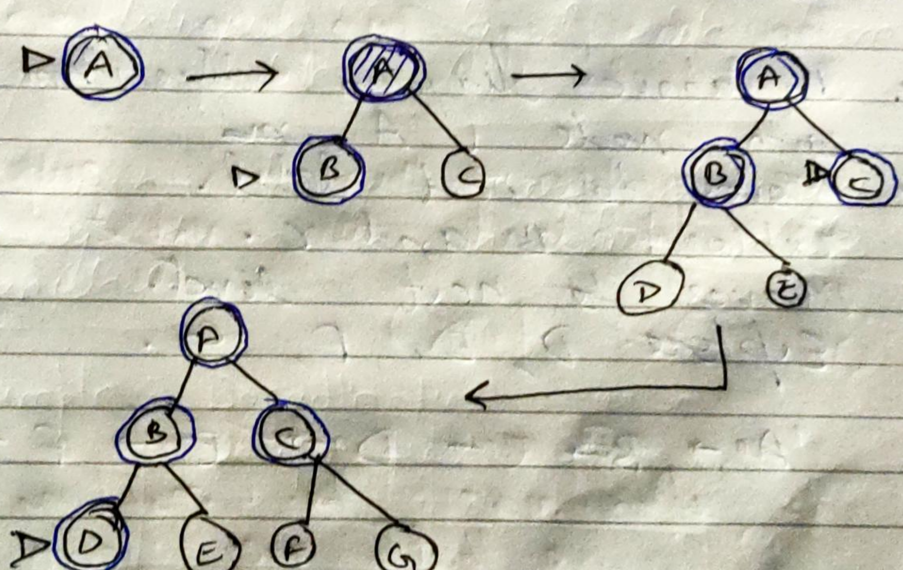
2) if problem.GOAL-TEST(node.STATE) then return SOLUTION(node).

3) frontier ← a fifo queue with node as only element

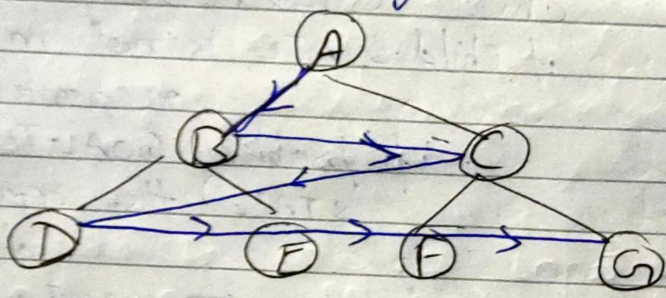
4) explored ← an empty set.

5) loop do
if EMPTY?(frontier) then return failure
node ← POP(frontier)
add node.STATE to explored.
for each action in PROBLEM.action(node.STATE) do
child ← CHILD-NODE(problem.node.action)
if child.STATE is not in explored or frontier then
if problem.GOAL-TEST(child.STATE) then return SOLUTION(child)
frontier ← insert(child.frontier)

<u>Explanation</u>



The algorithm visits and mark all the key nodes in the Graph in an accurate breadthwise fashion. This algorithm selects a single node and then visits all the adjacent node of the selected node.



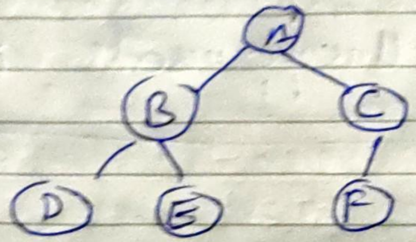
(Blue line denotes BFS traversal)
A → B → C → D → E → F → G

- In above diagram, algorithm starts with node A, ~~and~~
- Moves to next level.
- Explores B and C.
- Moves to next level.
- Explore D, E, F, G.

A → B → C → D → E → F → G.

Example

Let Graph be :



BFS traversal will give :-

A → B → C → D → E → F

(i.e, level wise).

Applications:-

1) Un-weighted Graphs.
2) P2P Network
3) Web Crawlers.
4) Navigation System
5) Network Broadcasting.

2) Machine Learning Areas

Supervised　Unsupervised　Reinforcement.

Supervised Learning.

This process involves input variable, which we call (X), and an output variable, which we call (Y). We use an algorithm to learn the mapping function from the input to the output.

In simple mathematics, output (Y) is dependent on input (X).

$$Y = f(X)$$

Here, ~~out~~ our end goal is to try to approximate the mapping function (f), so that we can predict the output variable (Y) when we are given new input variable (

Key Points

- Every input pattern that is used to train the network is associated with an output pattern, which is the target or the desired pattern.

- Learning process is based on comparision, between networks computed output and the correct output, generating "Error".

- The error generated is used to change network parameter that results in improved performance.

Real World Example:

It is a cat or a dog?
Image classification is a popular problem in Computer science.(computer vision) field. Here, the goal is to predict the class from which the image belongs to. In this problem we are interested to find class label, ie, it is cat or dog.

Unsupervised Learning.

Here, we use data points as reference to find meaningful structure and patterns in the observations.

Unsupervised learning is used to find meaningful patterns and grouping inherent in data, extracting generative features, and exploratory purpose.

Key points :-

- The expected or desired results are not presented to the network, ie, Target output is not presented to the network.

- The system learns of its own by discovering and ~~of~~ adapting to the structural features in the input pattern.

Real World Example.

Finding Customer Segment:

Clustering is an unsupervised technique. which is used to find natural groups on features space and interpret input data.

One common approach is to divide the data points in a way that each data point falls into a group that is similar to other data point in the same group based on a predefined similarities or distance metric in feature space.

Being able to determine different segment of Customer helps the marketing team.

Reinforcement Learning.

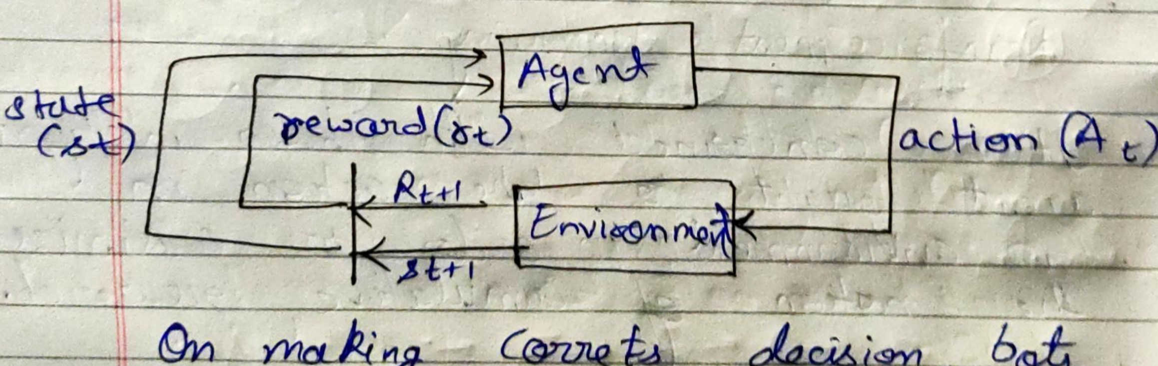
It is concerned with how intelligent agent ought to take action in an environment in order to maximise the notion of cumulative reward.

Key points.

- Indicated if the computed output is correct or incorrect.
- The information provided helps the network ~~is~~ in it's learning process.
- The reward is given for correct answer computed and a ~~particular~~ panelty for wrong answer.

Real life Example.

Trading.
With help of reinforcement learning, we train bots to learn online trading and act as traders.

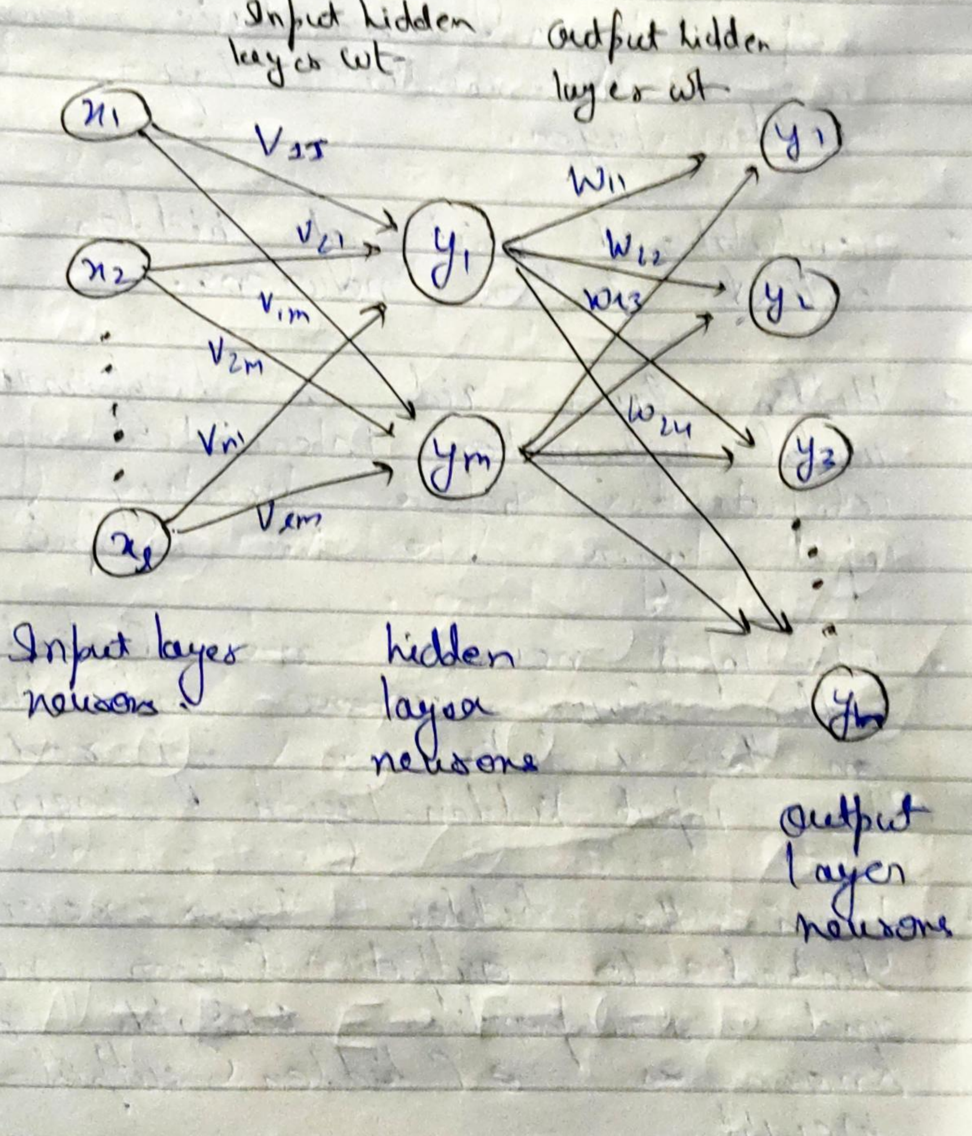


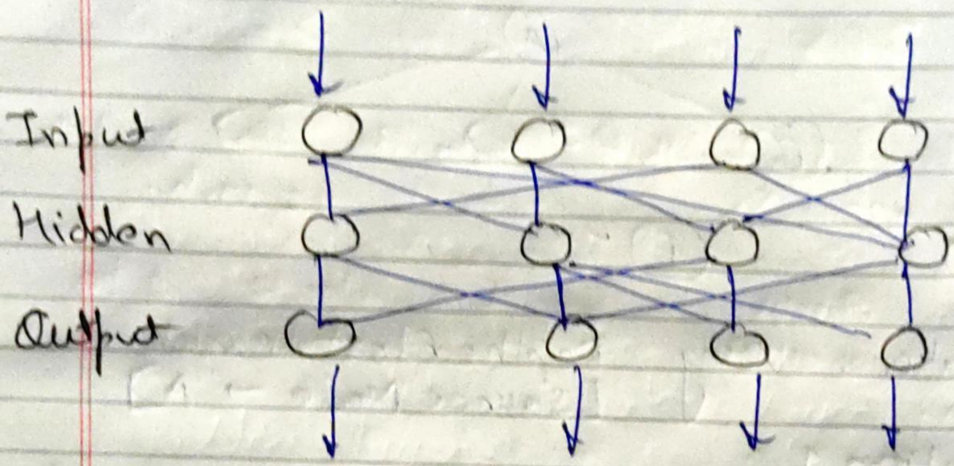
On making correts decision, bots are rewarded, else they have ~~are~~ to ~~penal~~ paid panelty.

3) Multilayer feedback Network.

- The name suggest, it consist of multilayers.
- The architecture of this class of network, besides having the input and the output layer, also have one or more intermediate layer, called hidden layer.
- Computational unit of hidden layer is called hidden neurons.
- Hidden layer aids in performing of useful ~~information~~ intermediatory computations before directing the input layer to the output layer
- The input neurons are linked to hidden layer neurons and the weights on these links are referred to as input-hidden-layer weights.
- Again the hidden layer neuron are linked to output layer neuron and corresponding weights are known as hidden output-layer weight.

- A multilayer forwarding network with I input neurons, m neurons in hidden layer and n output neurons is showed in configuration I-m-n.



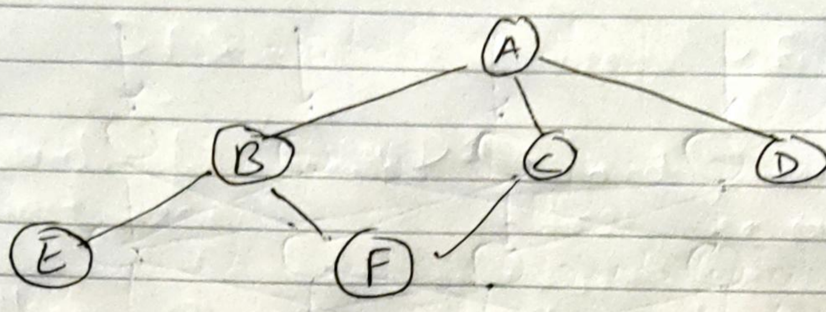


Information always moves in one direction; it never goes backward.

Mathematical function.

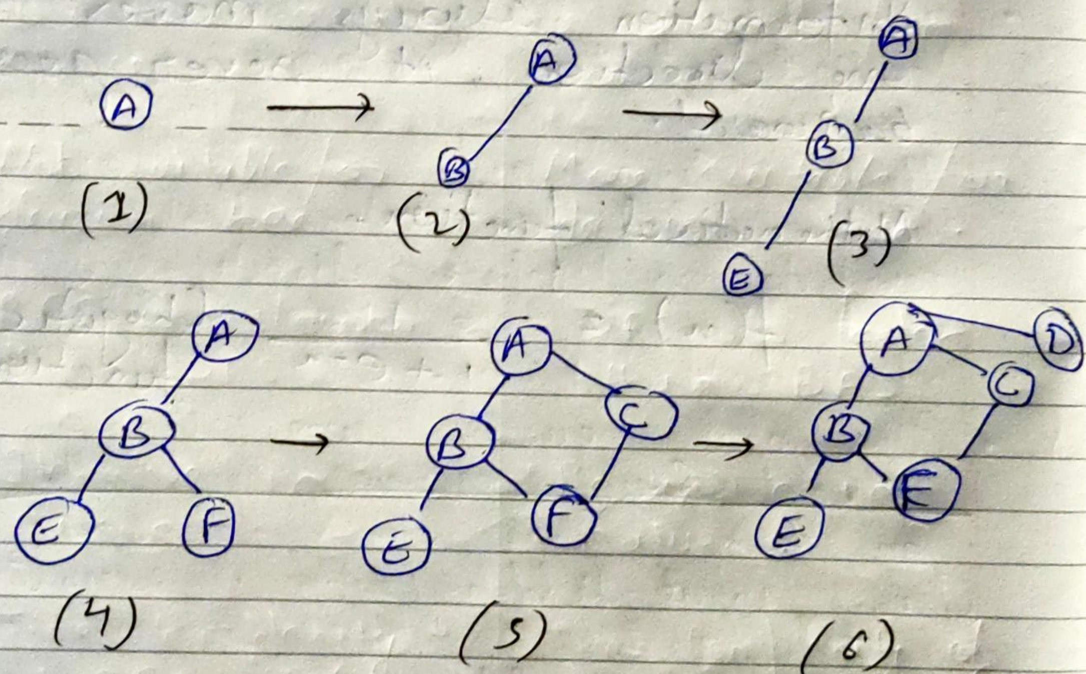
$$f(x) = \frac{1}{1+e^{-x}}$$ (Logistic function)

4)



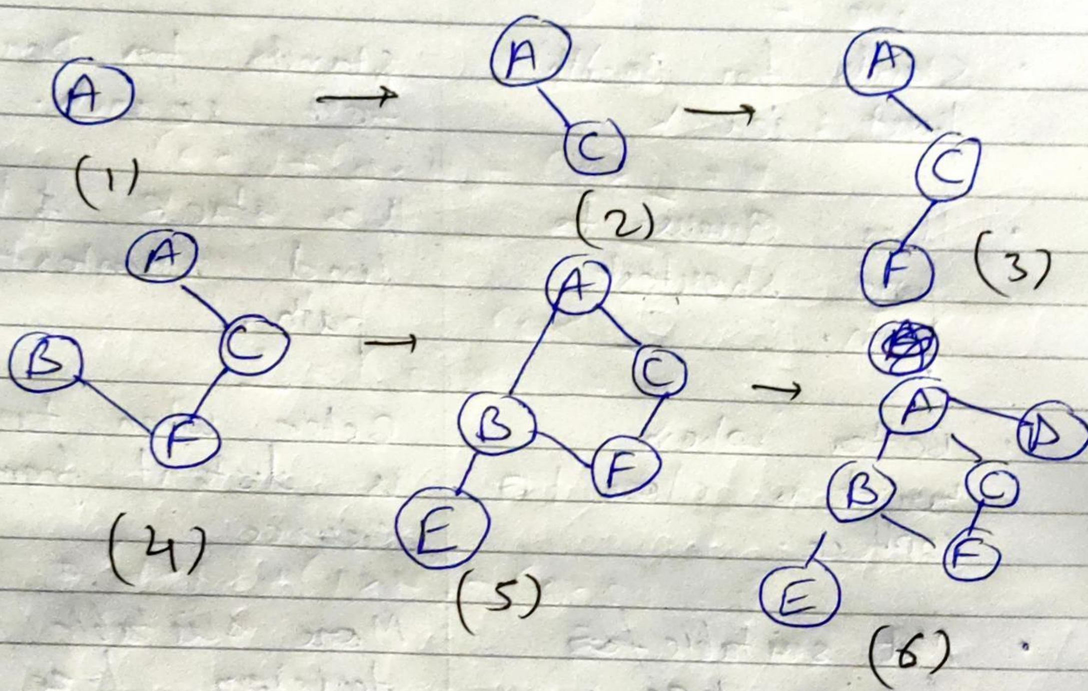
Solutions using Depth-First-Search [Source Node → A]

<u>Solution 1</u>



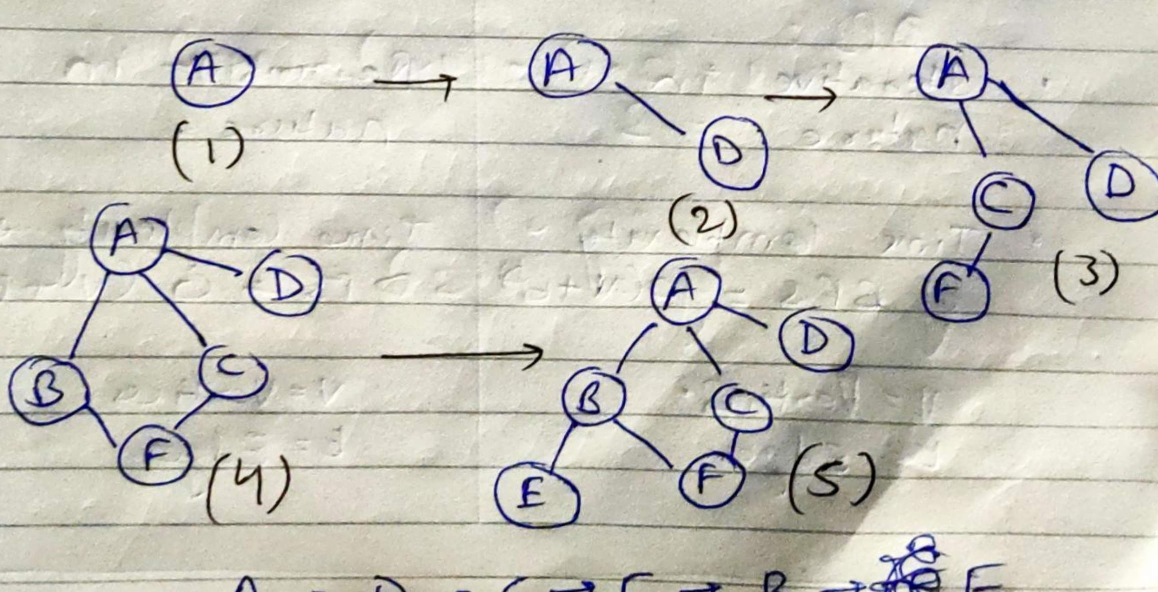
A → B → E → F → C → D

Solution 2



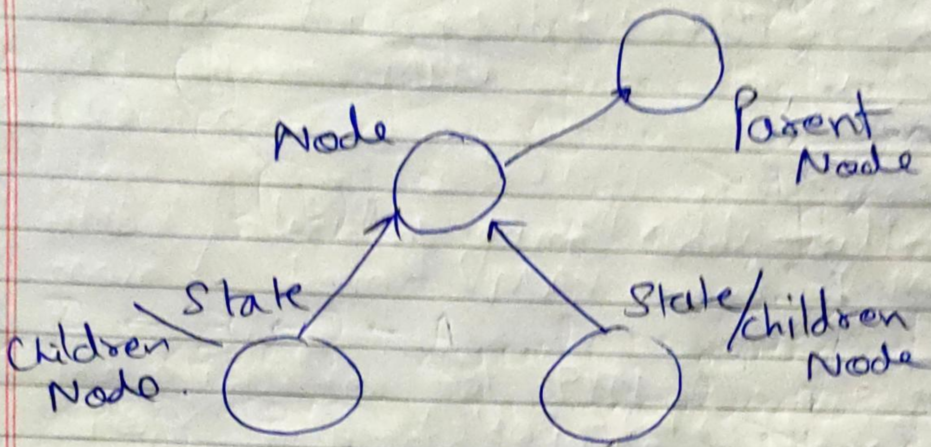
A → C → F → B → E → D

Solution 3



A → D → C → F → B → E

Nodes
~~Graph~~ data structure is most suitable data structure to store vertices of a graph in DFS algorithm

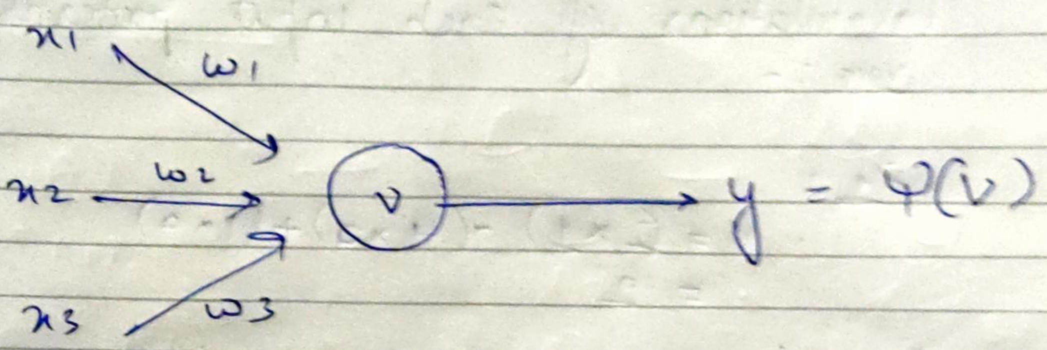


In Graphs, Node data structure is defined in such a way that it contains Link (pointers) to its parent and child.

Using this structure it is easy for us to maintain Graph structure and keep track of each Node.

BFS	DFS
Stands for Breadth first search	Stands for Depth first search.
Uses Queue to find shortest path.	Uses stack to find shortest path.
Better when target is close to source.	Better when target is distant from source
Not suitable for decision tree.	More suitable for decision tree.
BFS is slower than DFS	DFS is faster than BFS.
Iterative in nature	Recursive in nature.
Time complexity of BFS = $O(V+E)$ V = Vertices. E = Edges	Time complexity of DFS = $O(V+E)$ V = Vertices E = Edges.

5)



$w_1 = 2$
$w_2 = -4$
$w_3 = 1$

~~Activat~~ Step function:

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Table : ~~Pattern~~

Pattern	P_1	P_2	P_3	P_4
x_1	1	0	1	1
x_2	0	1	0	1
x_3	0	1	1	1

Calculations of each input patterns are :-

P_1 :

$$v = (2 \times 1) - (4 \times 0) + (1 \times 0)$$
$$= 2$$

$\because (2 > 0)$

$\therefore y = \varphi(2) = 1$

P_2 :

$$v = (2 \times 0) - (4 \times 1) + (1 \times 1)$$
$$= -3$$

$\because (-3 < 0)$

$\therefore y = \varphi(-3) = 0$

P_3 :

$$v = (2 \times 1) - (4 \times 1) + (1 \times 1)$$
$$= 3$$

$\because (3 > 0)$

$\therefore y = \varphi(3) = 1$

P4 :

$v = (2\times1) - (4\times1) + (1\times1)$
$= -1$

$\because (-1 < 0)$
$\therefore y = \varphi(-1) = 0.$