2320030484
Sec-4
Branch - CSE

AI-ML AIM -01

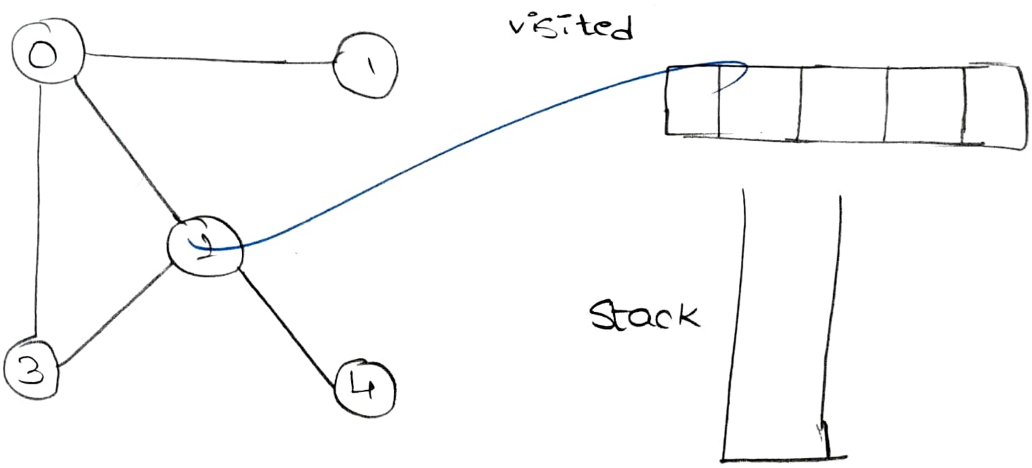
Uniformed Search:

DFS (depth first search):

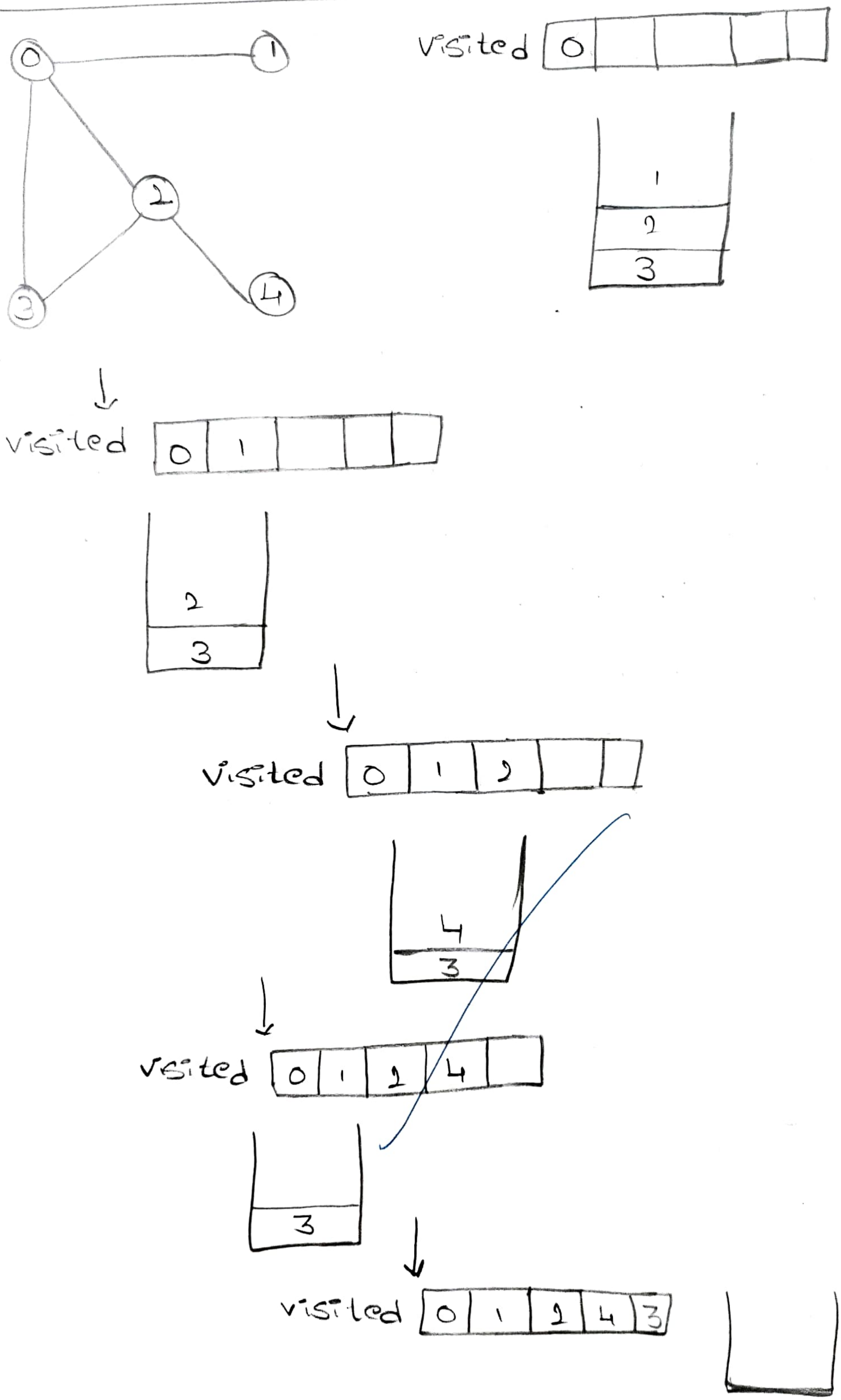
Depth first search is an algorithm used for traveling or searching tree or graph data structures. This algorithm starts at root (or any arbitary node in the case of a graph) & explores as far as Possible along each branch before back tracking.

DFS is often implemented using a stack, either explicity or through the System's call stack in a recursive implementation.

Ex:



↓



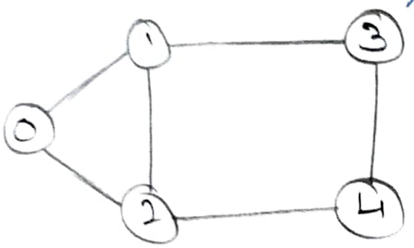
0
1
2
3
4
visited 0
1
2
3
visited 0 1
2
3
visited 0 1 2
4
3
visited 0 1 2 4
3
visited 0 1 2 4 3

Path = 0 → 1 → 2 → 4 → 3

BFS (Breadth first search):

Breadth first Search is an algorithm used for travelling or Searching tree or graph data structures. Unlike depth first search, BFS explores the neighboor nodes at the Present depth Prior to moving on to nodes at the depth level.

BFS is often used to find the Shorts Path in an unweighted graph, as it explores all nodes at the depth level before moving deeper.



visited

Queue

F R

↓

visited

1				

Queue

0				

F

↓

visited

0	1	2		

Queue

1	2			

F

↓

visited

0	1	2	3	

Queue

2	3			

⟶

visited

0	1	2	3	4

Queue

3	4			

F

↓

Visited	0	1	2	3	4
Queue					

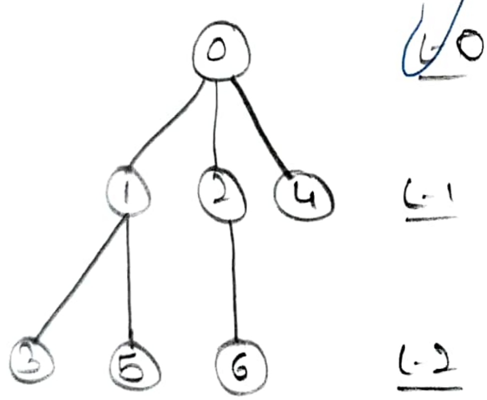
←

Visited	0	1	2	3	4
Queue	4				

IDFS (Itterative deepning first search):

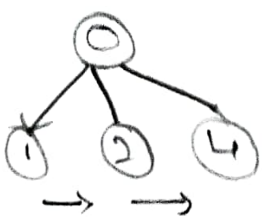
Itterative deepening depth first search is an algorithm that combines both DFS & BFS concept of exploring the search space level by level. IDFS is particularly useful in scenario where the search space is large & the depth of the solution is unknown.

Ex:



Iteration - 0 : 0 0

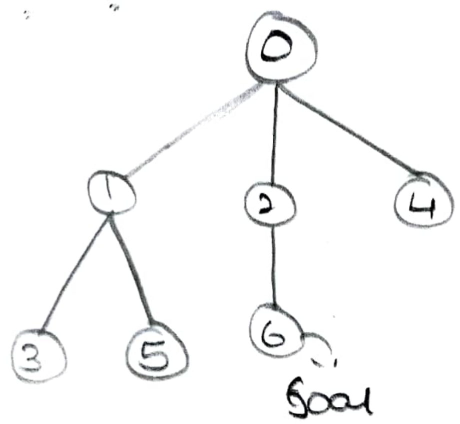
Iteration - 1 :



Path

↳ 0 → 1 → 2 → 4

Iteration - 2:



Path:

0 -> 1 -> 3 -> 5 -> 2 -> 6

Informed Search:

A* search algorithm:

A* Search algorithm is a Popular & efficient algorithm used for finding the shortest path between nodes in a graph. It is widely used in various applications, such as path finding in games, robotics & AI. A* is both complete & optional meaning it will always find the shortest path if one exist & it does so effectively by combining aspects of both depth first search (DFS) and breadth first search (BFS).

A* uses a priority queue to explore nodes in a way that minimizes that total estimated cost from the start node to the goal node

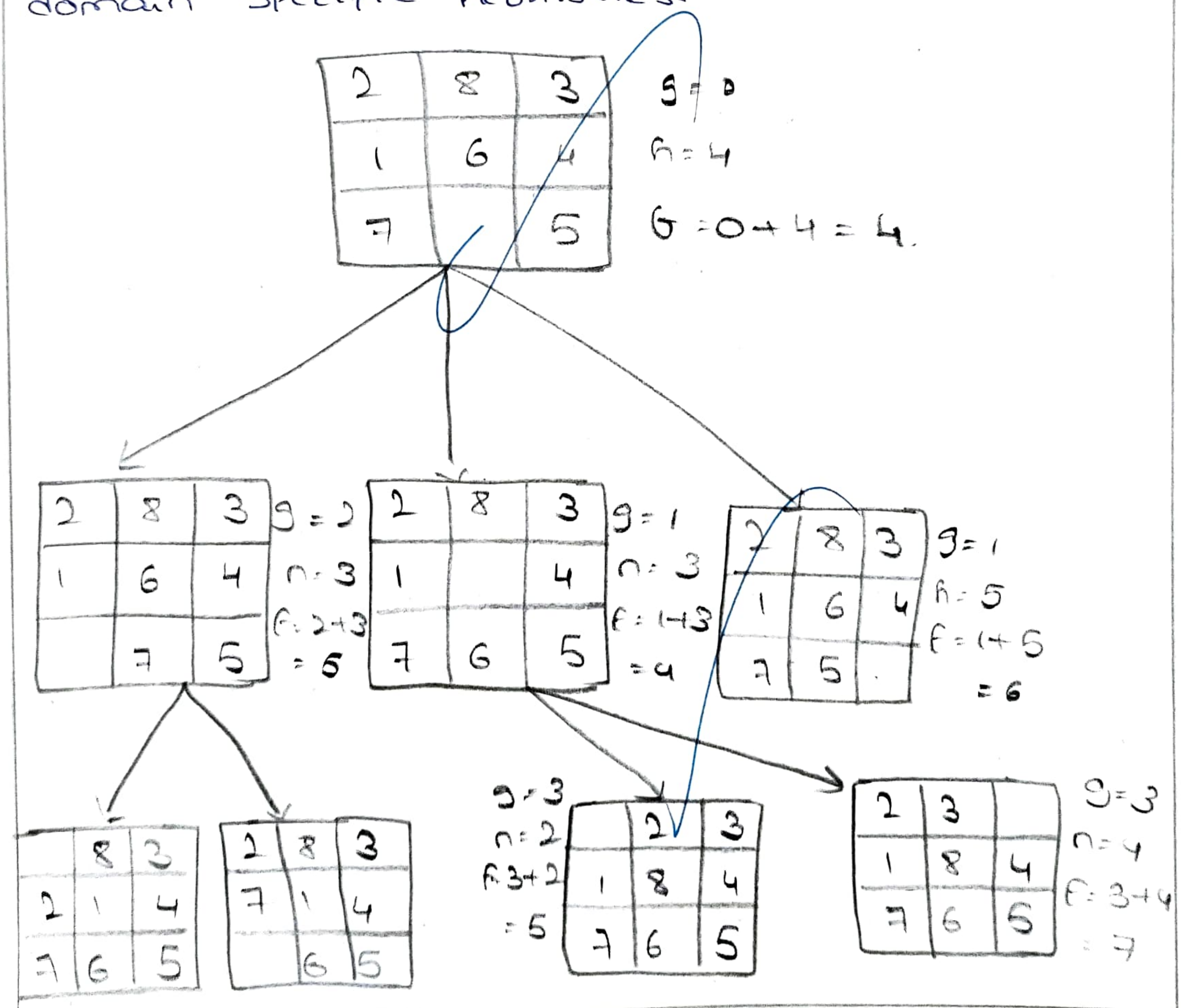
The algorithm Priorities nodes Based on a Cost function 'f(n)'.

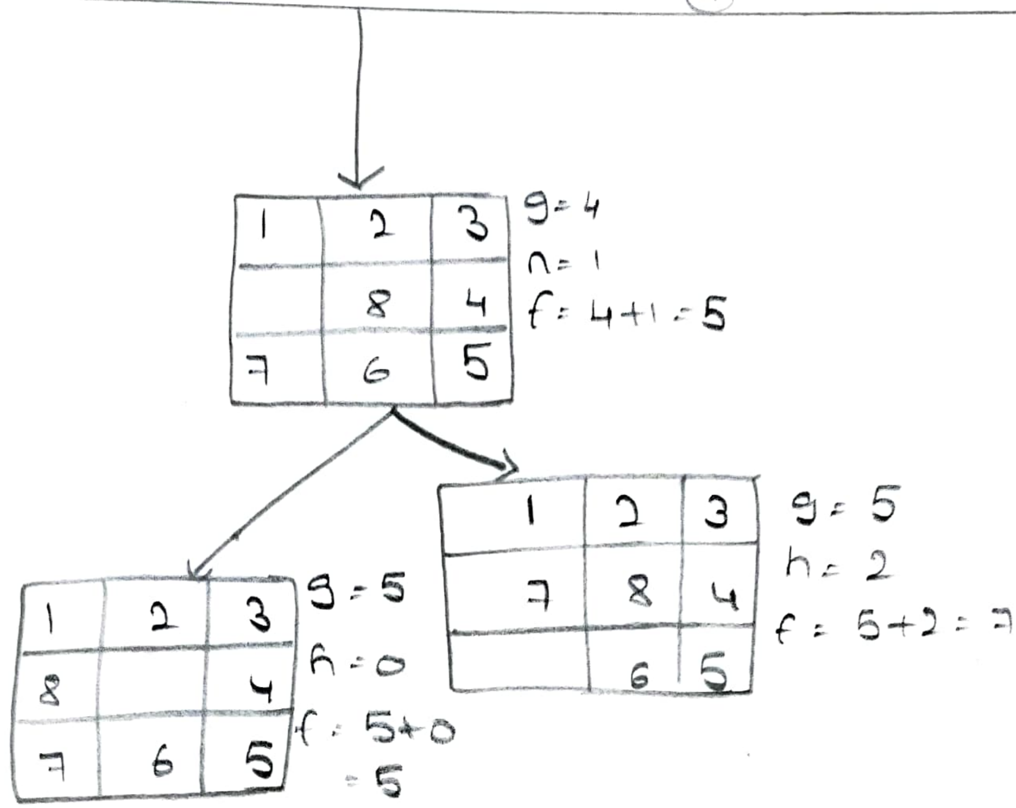
Cost function:

$$f(n) = g(n) + h(n)$$

* g(n): The actual cost from the start node to node n.

* h(n): estimates of the cost from node n to the goal node this esitmates is typically a function of the straight-line distance or other domain-specific heuristics.



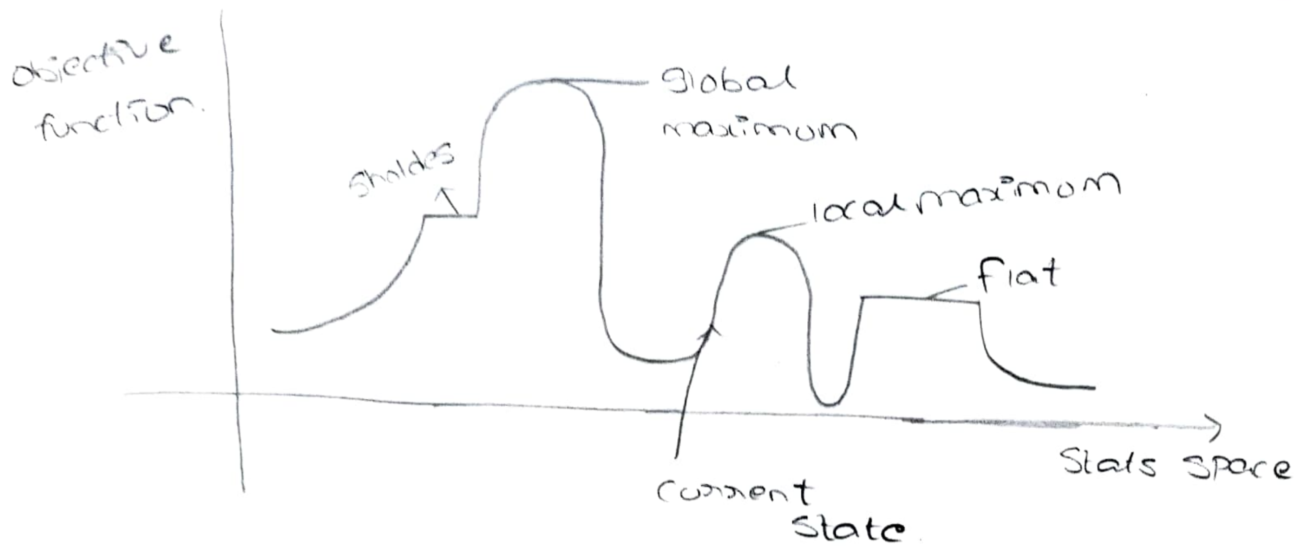


final
Stats.

Hill Climbing Algorithm:

Hill Climbing is a simple optimization algorithm used in AI to find the best Possible solution for a given problem. It belongs to the family of local search algorithms & is often and in optimization problems where the goal is to find the best solution from a set of Possible solutions.

* In hill climbing, the algorithm starts with an initial solution & then iteratively makes small changes it in order to improve the solution. These changes are based on a heuristic function.



local maximum: It is a State which is better than its neighbouring stats however there exists a state which is better than it. This stats is better here the value of the objective function is higher than its neighbour.

Global maximum: It is a stats which is better than its neighbour state however there exists a state which is better than it. This stats is better because here the value of the objective function is highest value.

Pleateau / flat local maximum: It is a flat region of stats space where neighbour stats have the same value.

Ridge: It is a region that is higher than its neighbours but it half have a slope. It is pecial kind of local maximum.

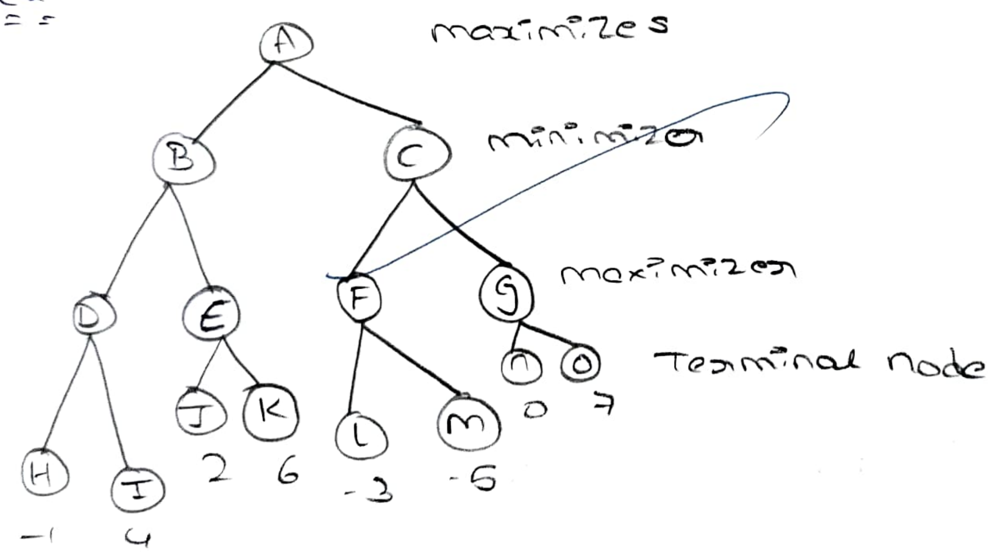
Min max Algorithm:

* Min max Algorithm is a recursive or backtraking algorithm which is used in decision making and game. theory: It provides an optimal move for the player assuming that the oponent is also playing optimally

* Min Max algorithm uses recursion to search through the game tree.

* In this algorithm two players plays the game one is max & other is Min.

Ex:



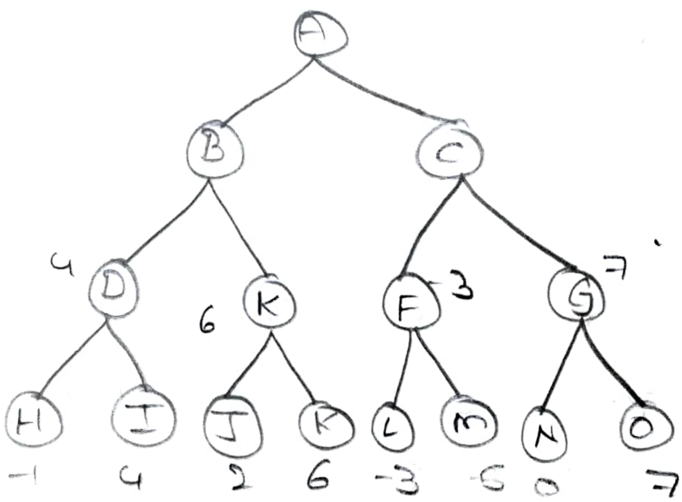
Step 1:

D = max (H, I) = max (-1, 4) = 4

E = max (J, K) = max (2, 6) = 6

F = max (L, M) = max (-3, -5) = -3

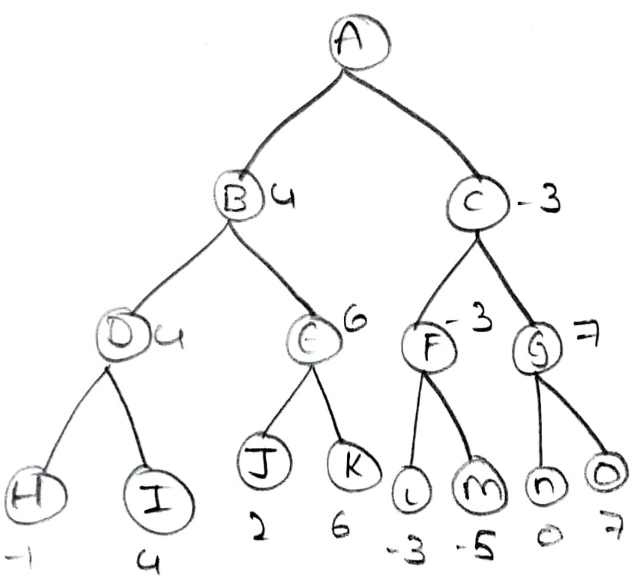
G = Max (N, O) = max (0, 7) = 7



Step-3:

B = MIN (D, E) = MIN (4, 6) = 4

C = MIN (F, G) = MIN (-3, 7) = -3



Step-4:

A = MAX (B, C) = MAX (4, -3) = 4