AIML ALM-01

L. Abijeet
2320030193

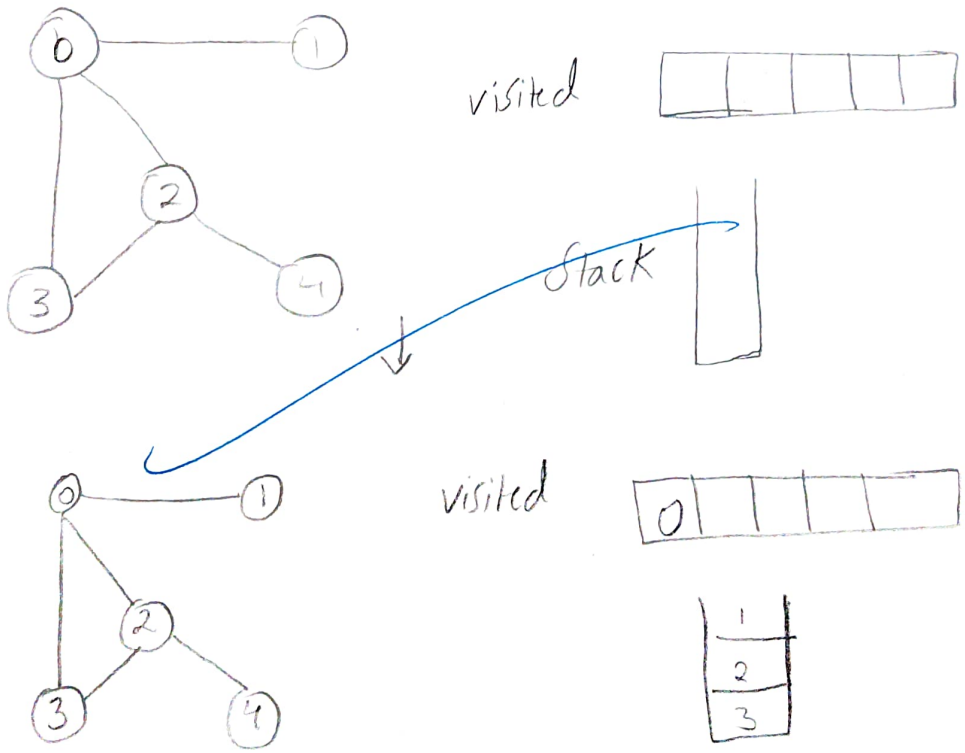
Uninformed search -

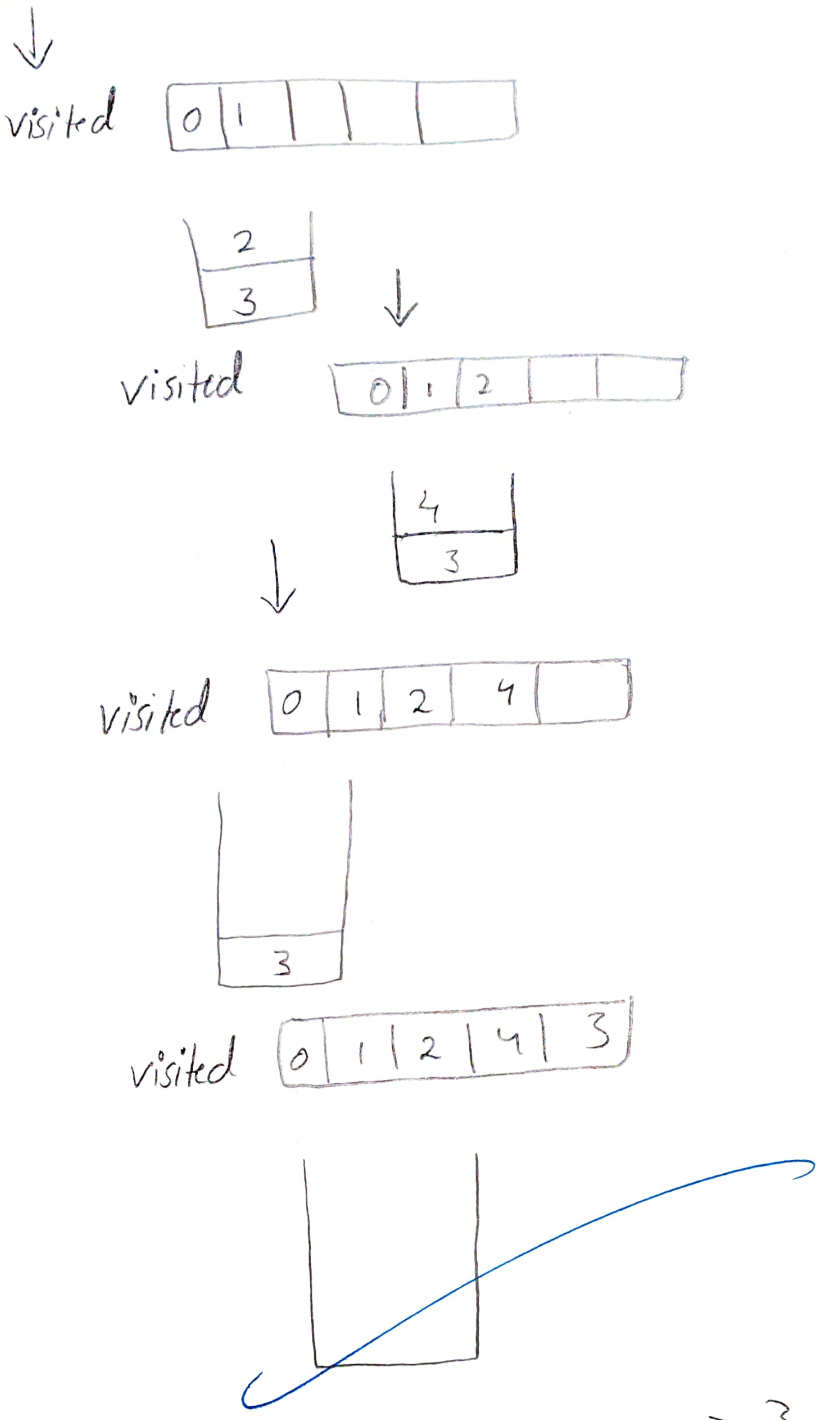
DFS (depth first search) :

Depth first search is an algorithm used for travelling 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 explicitly or through the system call stack in a recursive implementation

Ex

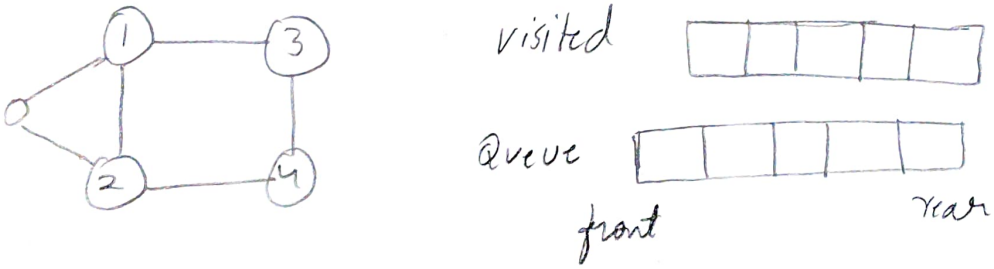


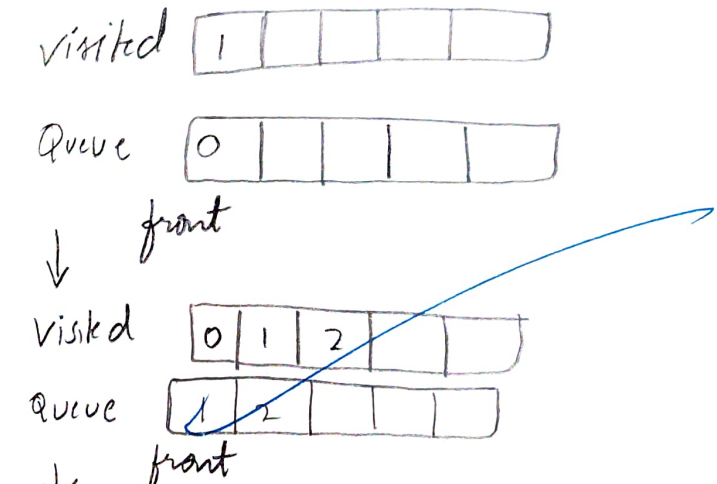

visited
0 1
2
3
visited
0 1 2
4
3
visited
0 1 2 4
3
visited
0 1 2 4 3
Path = 0 → 1 → 2 → 4 → 3

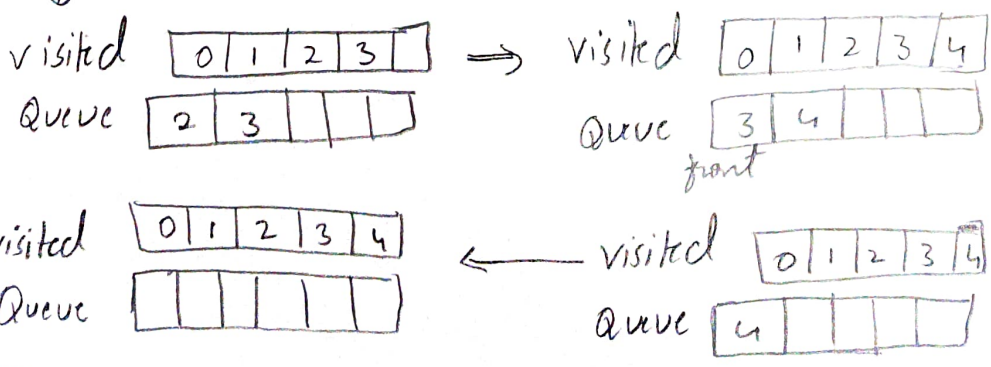
BFS (Breadth first search) -

Breadth first search is an algorithm used for travelling or searching tree or graph data structures. Unlike depth first search, BFS explores the neighbour nodes at the present depth prior to moving on to nodes at depth level

BFS is often used to find the shortest path in an unweighted graph, as it explores all nodes at the current depth level before moving deeper







IDFS (iterative deepning first search)

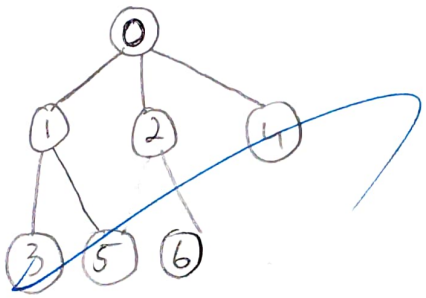
Iterative deepening depth first search is an algorithm that combines both DFS & BFS concept of exploring the search space level by level. IDFS is particularly usefull in scenarios where the search space is large & the depth of solution is unknown.

Ex:

Level-0

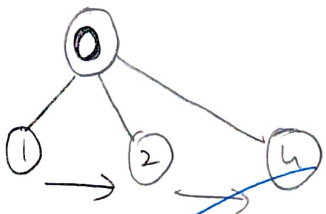
level-1

level-2



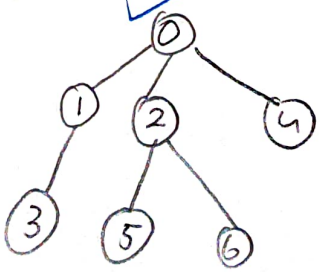
iteration - 0 - (0) 0

iteration-1



path

iteration 2



path

0 → 1 → 3 → 5 → 2 → 6

Informed Search

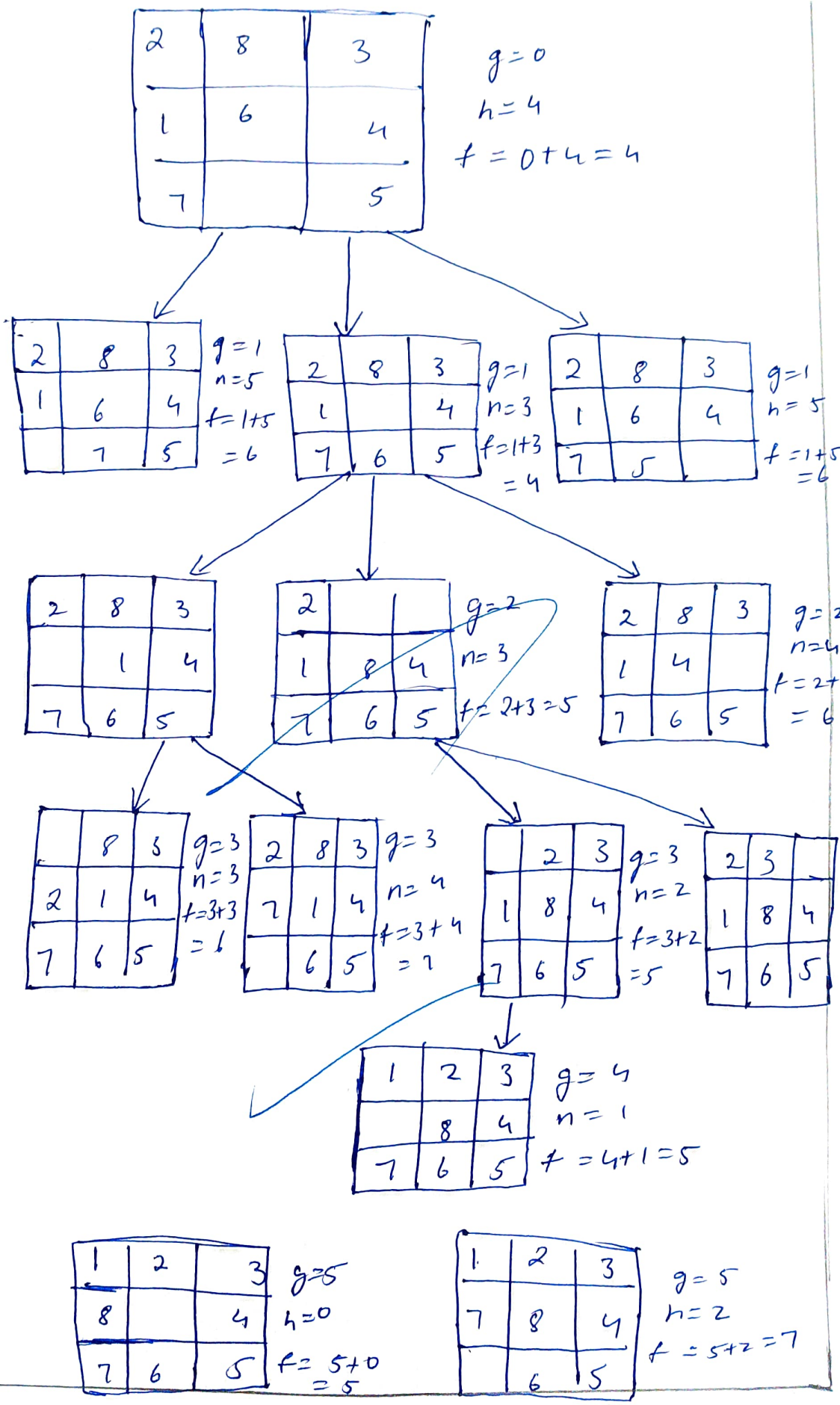
A* search algorithm is a popular & efficient algorithm used for finding the shortest path between nodes in a graph. It is widely used in various application, such as path finding in games, robotics & AI. A* is both complete & optimal meaning it will always find the shortest path if one exists & it does so efficiently by combining aspects of both depth-first-search (DFS) and Breadth first search (BFS)

A* uses a priority queue to explore node in a way that minimizes the total estimated cost from the start node to the goal node. The algorithm prioritize 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): The estimate of the cost from node n to the goal node. this estimate is typically a function of the straight-line distance or other domain-specific heuristics

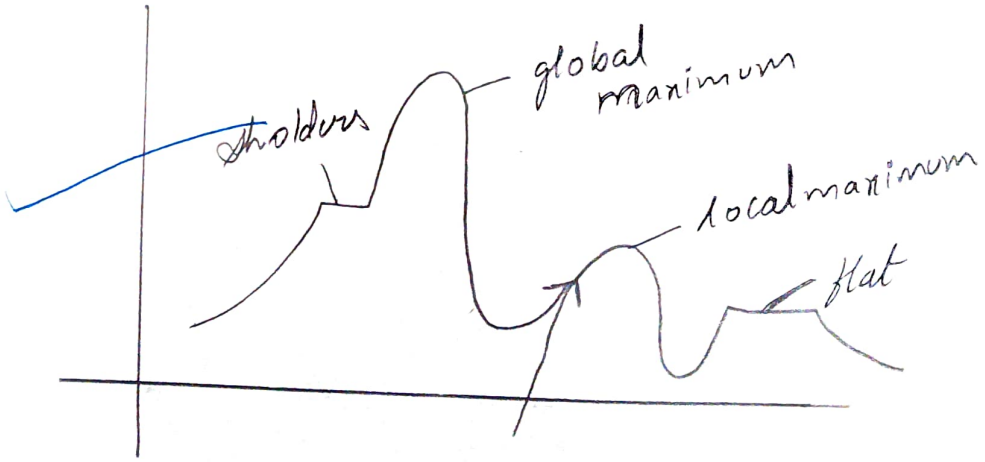

g=0
h=4
f=0+4=4
g=1
h=5
f=1+5=6
g=1
h=3
f=1+3=4
g=1
h=5
f=1+5=6
g=2
h=3
f=2+3=5
g=2
h=4
f=2+4=6
g=3
h=3
f=3+3=6
g=3
h=4
f=3+4=7
g=3
h=2
f=3+2=5
g=4
h=1
f=4+1=5
g=5
h=0
f=5+0=5
g=5
h=2
f=5+2=7

Hill climbing Algorithm

Hill climbing is a simple optimization algorithm used in AI to find the best possible solution for a given problem. It belong to the family of local search algorithm & is often used 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 are based on heuristic function that evaluates

to make these small changes until it reduces a local maximum, meaning that no further improvement can be made with the current set of moves.



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

Global maximum - It is a state which is better than its better than it. This states is better because here the value of the objective function is higher value.

plateau / flat local maximum : It is a flat region of states space where neighboring states have the same values

Ridges : It is a region that is highers than its neighbors but it dalf have a slope. It is special kind of local maximum.

current state - The region of the state spalt. diagram where we are currently present during the search.

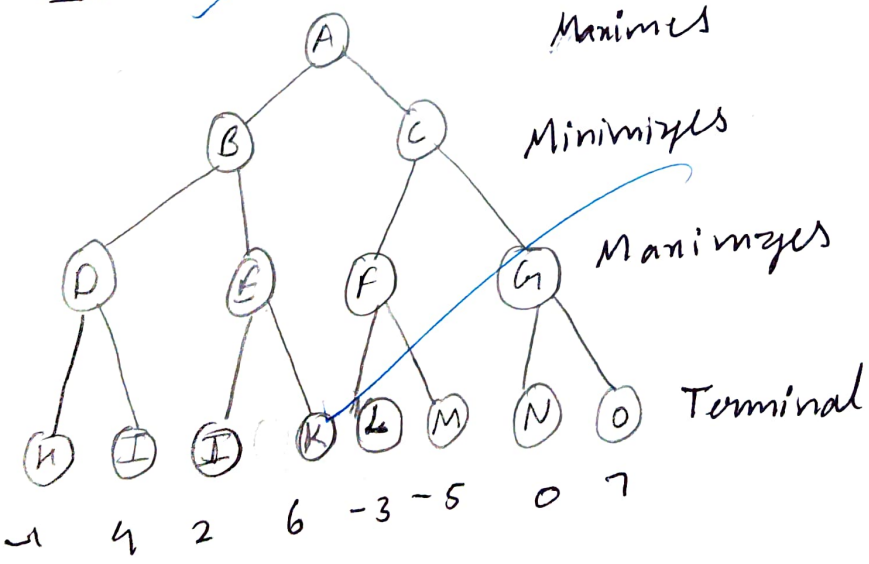
Shoulders - It is a plteau that has a uphill edge.

MinMax Algorithm

* MinMax algorithm is a recursive or back tracking algorithm which is used in decision making and game theory. It provides an optional move for the player assuming that the opponent is also playing optimally.
* MinMax algorithm use recursion to search through the game tree.
* MinMax algorithm mostly used for game playing in AI. Such as chess, checkers, tic-tac-toe go and various two-players game. This algorithm computes the minmax decision for the current state.

Ex

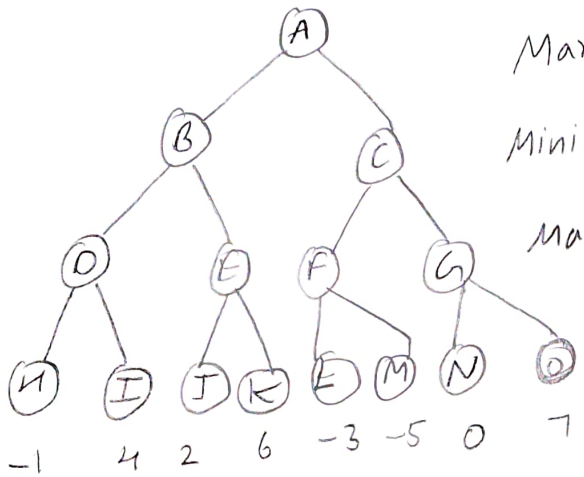
Step-1



Step-2 for the maximizes find the maximizes values from the child nodes & fix it

$= \max(J,)$

$G = \max(N, O) = \max(0, 7) = 7$



Maximizes

Minimizes

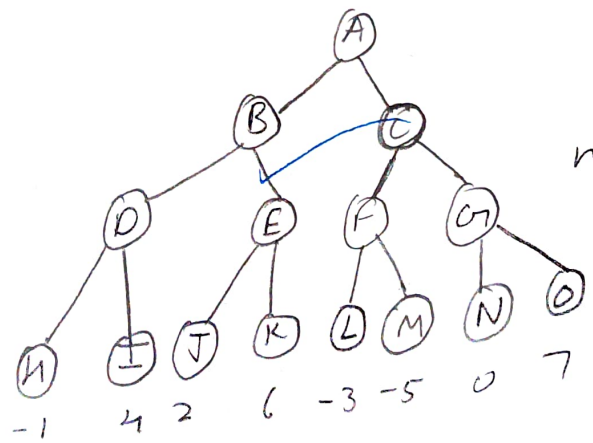
Maximizes

Terminal Node

Step-3 . for finding the Minimizes take the minimizes values of the child nodes

$B = \text{MIN}(D, E) = \text{MIN}(4, 6) = 4$

$C = \text{MIN}(F, G) = \text{MIN}(-3, 7) = 3$



Maximizes

minimizes

maximizes

Terminal node

Step 4 - for finding the maximizes take the maximum values from the child node

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