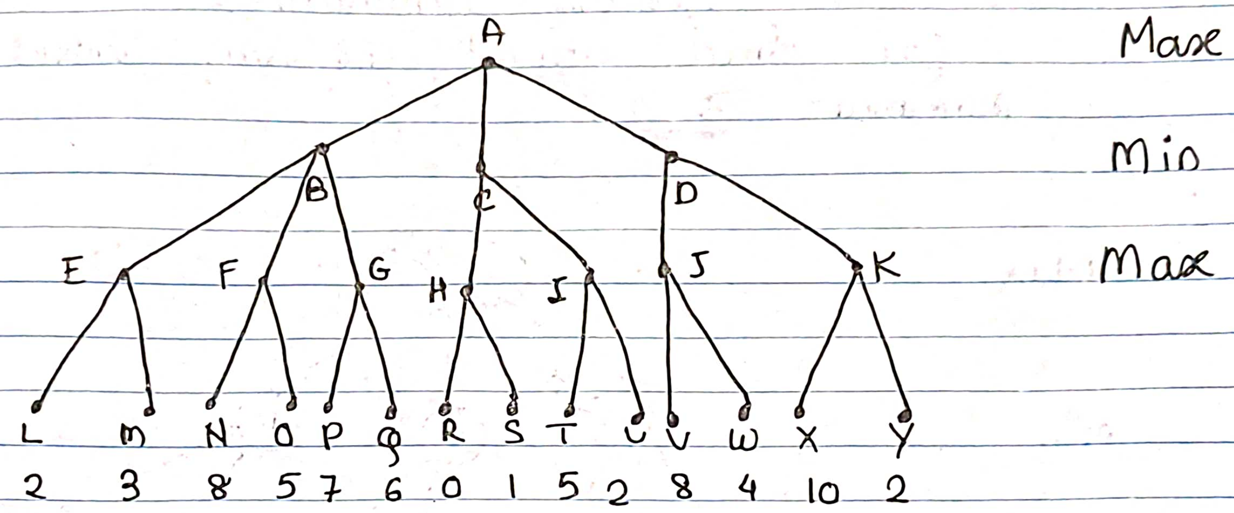
Name : Paurari Nagarkar
ID : W1650209

Homework No : 3

1] Consider following game tree in which utility Value are all from first Players point of view. Assume that first Player is maximizing Player

→

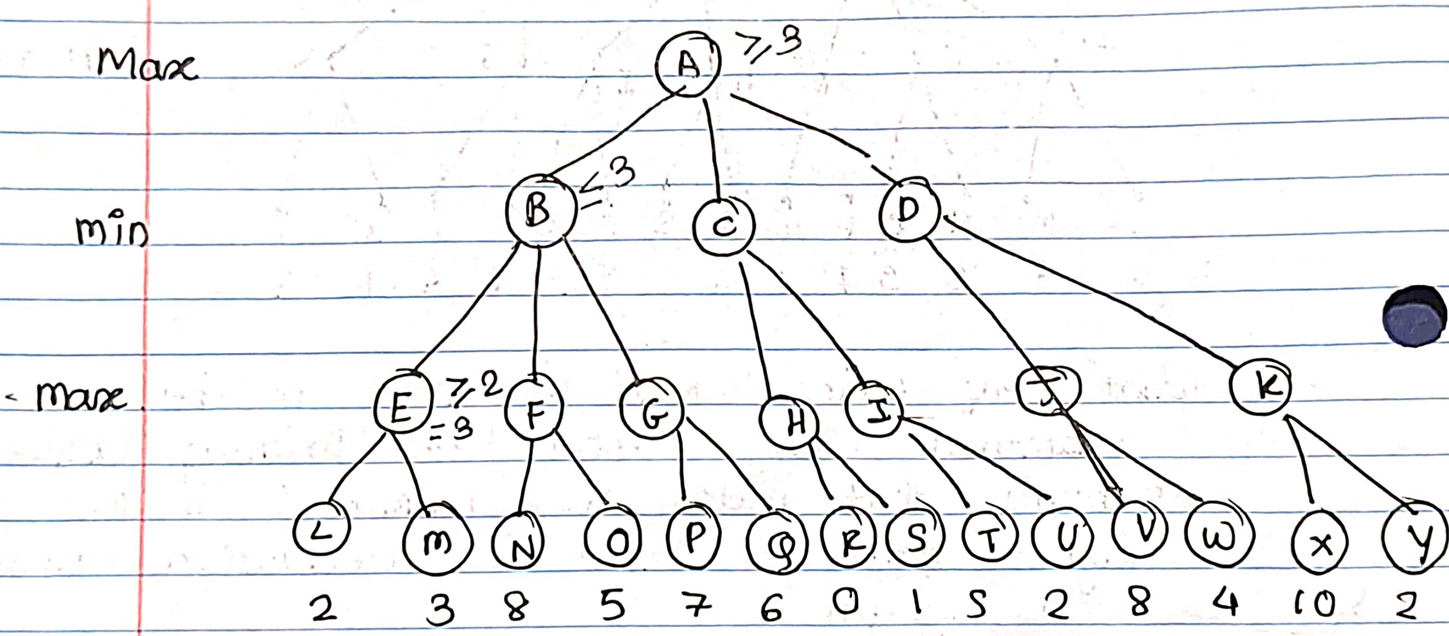


a] What leaf nodes would not need to be examined using alpha beta pruning algorithm assuming that nodes are examined in left to Right order ? Show the derivation Procedure on graph

→ A → Node (Root Node) → Collect Maximum as Value
B, C, D → Collect minimum Value from its leaf node.
E, F, G, H, I, J, K → Collect maximum Value from its leaf Nodes.

L M N O P Q R S T U V W X Y → leaf nodes.

To decide which leaves/Node to be emitted. lets start from lowest/leftest Node E, m for Node E collect maximum from both. Now maximum node for E but for B we are Selecting minimum ≤ 3 But again for third round we are considering maximum ≥ 3.



→ ① Now for E consider Initial value as it traces leaf L : 2 our objective is to maximize the value for node E. We will apply condition $E = \geq 2$. i.e E can began at 2 and go further But can not be less than 2.

→② tracing leaf m : 3 as it ≥ 2. E can be Initialized to 3.

③ Now as after getting value of E. The main objective is to find value of A.

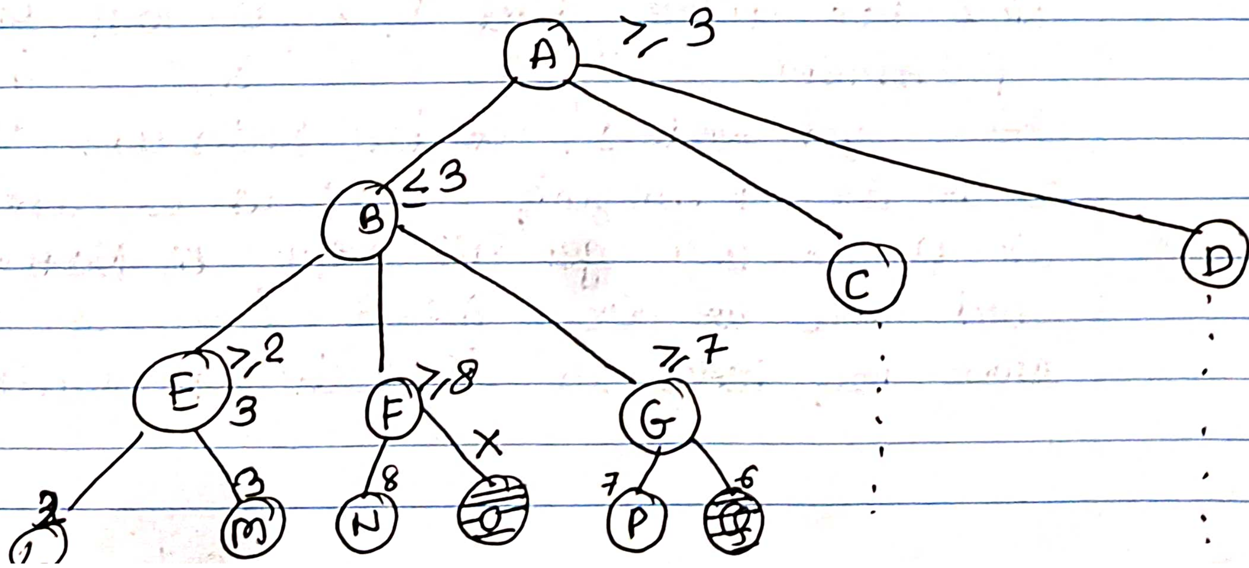
④ When Player Min Plays at Node B. It gets minimum value for its Node among the

group of nodes hence B will always have Value $B \leq 3$.
Hence without going from all the nodes we can Predict $B \leq 3$ as we initialize minimum Value of B amongst Nodes leafs.

(5) We can pre determine the condition for A as well. The value of A can be anything greater than 3. as we are maximizing the value of node. hence from Earlier Initial Info that we have. $A \geq 3$.

(6) Consider for Node F we consider its node N : 8
As we are maximizing it we can consider $F \geq 8$
But we know its Parent node $B \leq 3$ hence it does not matter what next node is as we are minimizing at node B.
hence we can Prune its Succedding node O.

(7) Similarly for Node G. we consider nodes P, Q.
Consider $G \geq 7$ as we are taking maximizing
But for Node B we use condition ≤ 3.
hence does not matter what other node is.
as it will take maximum value which is greater than 3. hence Pruning next node.

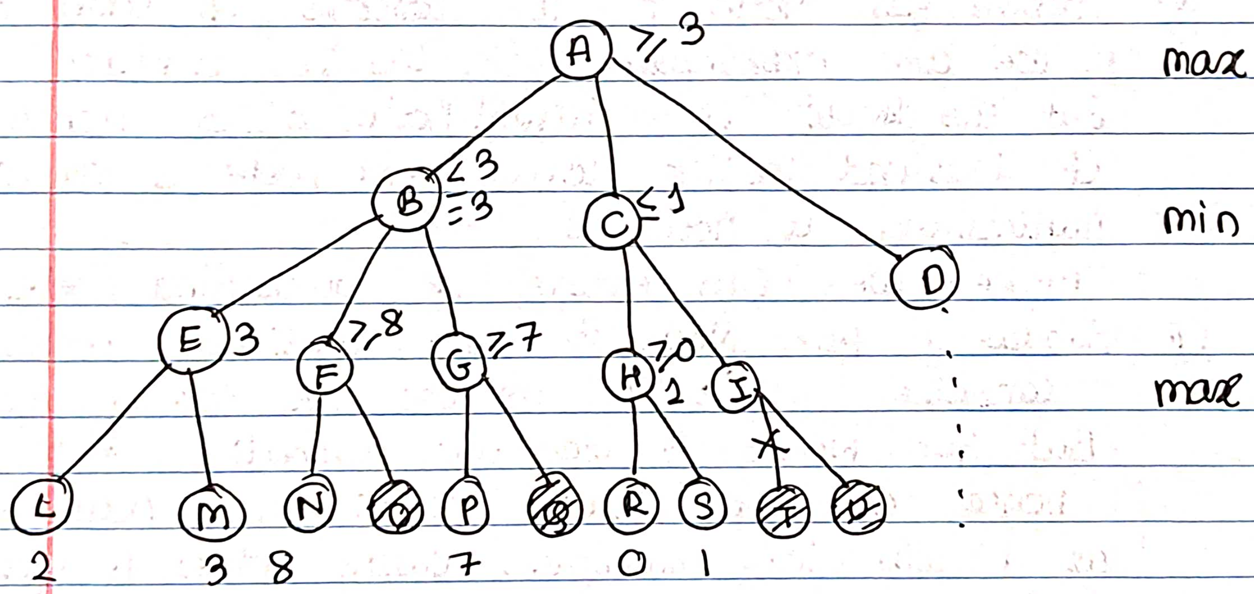


→ Similarly for Node H we have R, S as Initial nodes. As we are maximizing hence as R = 0 we set Condition that $H \geq 0$.

→ going to next node S = 1. As $H \geq 0$, value of H is set to 1.

→ At level 1 we are minimizing hence checking Condition $C \leq 1$.

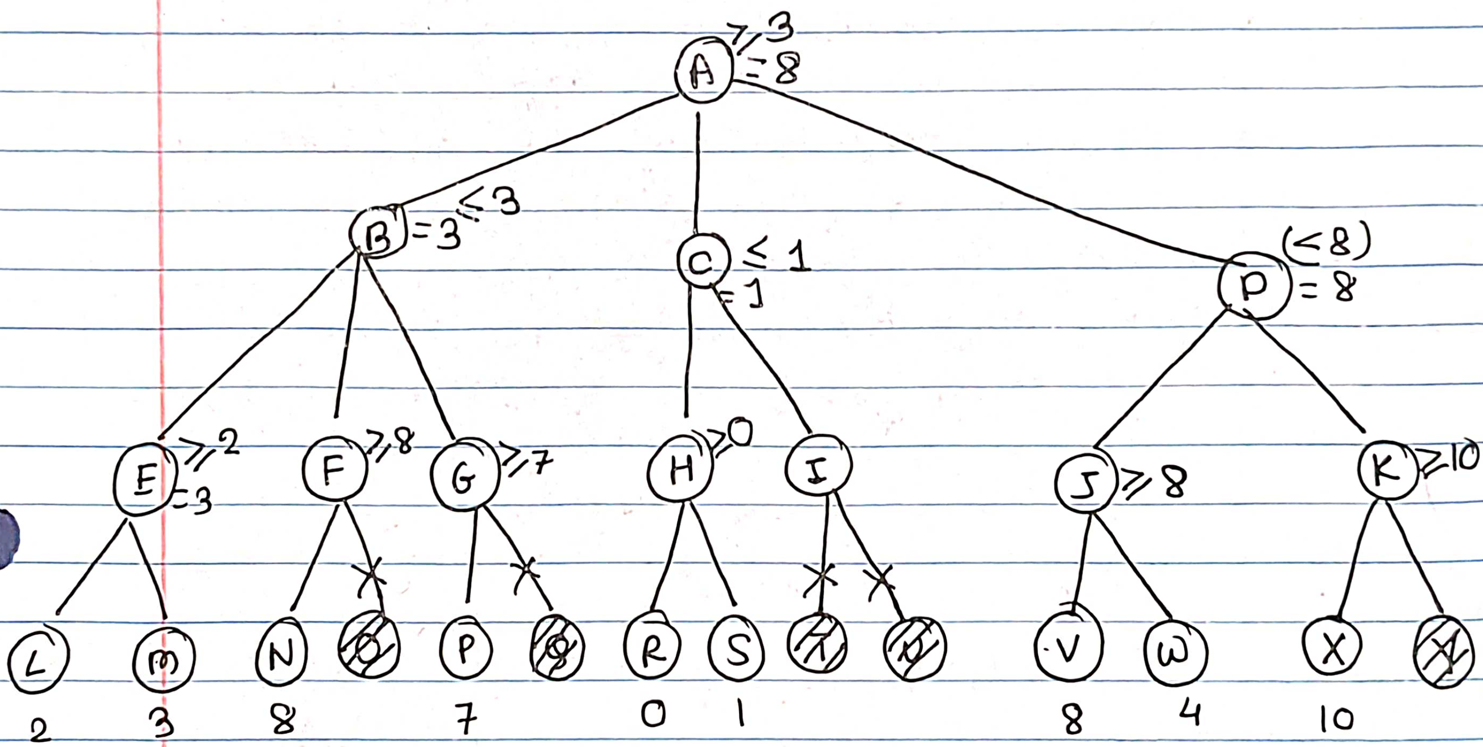
→ Now checking A whose Condition $A \geq 3$. But Value of Node C is ≤ 1, hence we won't need to check further as no Node will satisfy the Condition hence we can Prune remaining nodes.



Now for D its child J, K we will follow Same Procedure.

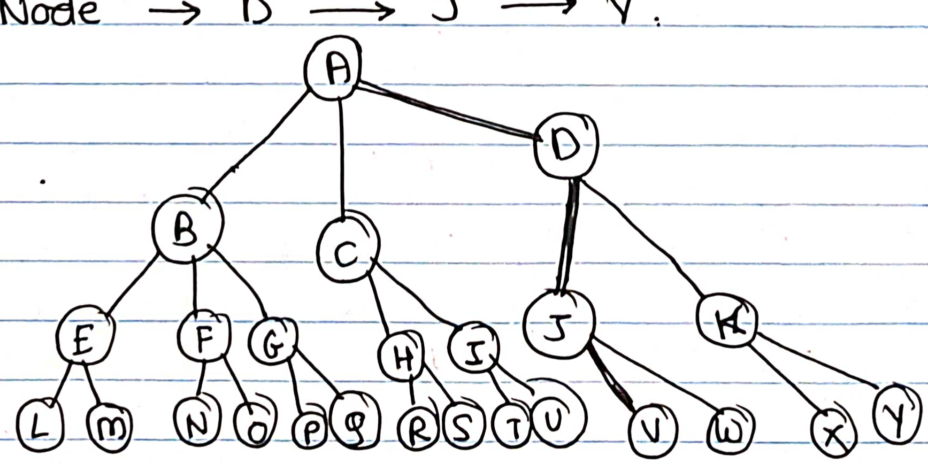
Check for Node J, its leaf Nodes are V, W, V: 8 as we are maximizing $J \geq 8$. as we are minimizing at D we will get all values of Nodes ≤ 8. and Now we know $A \geq 3$. hence we can check other leaf nodes as it Satisfies Condition.

Now w = 4 hence w can be excluded
Moving to node K. tracing its leaf node X.
with value 10. hence K ≥ 10. But we are
minimizing at D hence D ≠ 10. hence we can
skip its remaining node.



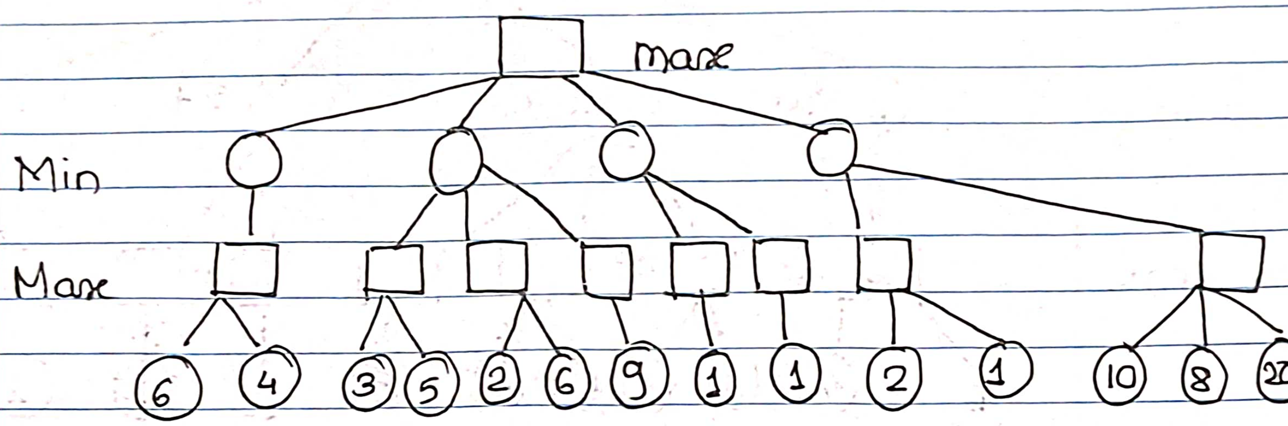
Hence By alpha Beta Pruning Algorithm we get
that leaf nodes O, Q, T, U, Y can be Pruned.

b] As A=8 is optimal Actual value of node hence
first Player should move towards D. to Right.
Root Node → D → J → V.



c. The final minmax value for Root Node is $\underline{\underline{8}}$
Indicating A → Root Node → 8

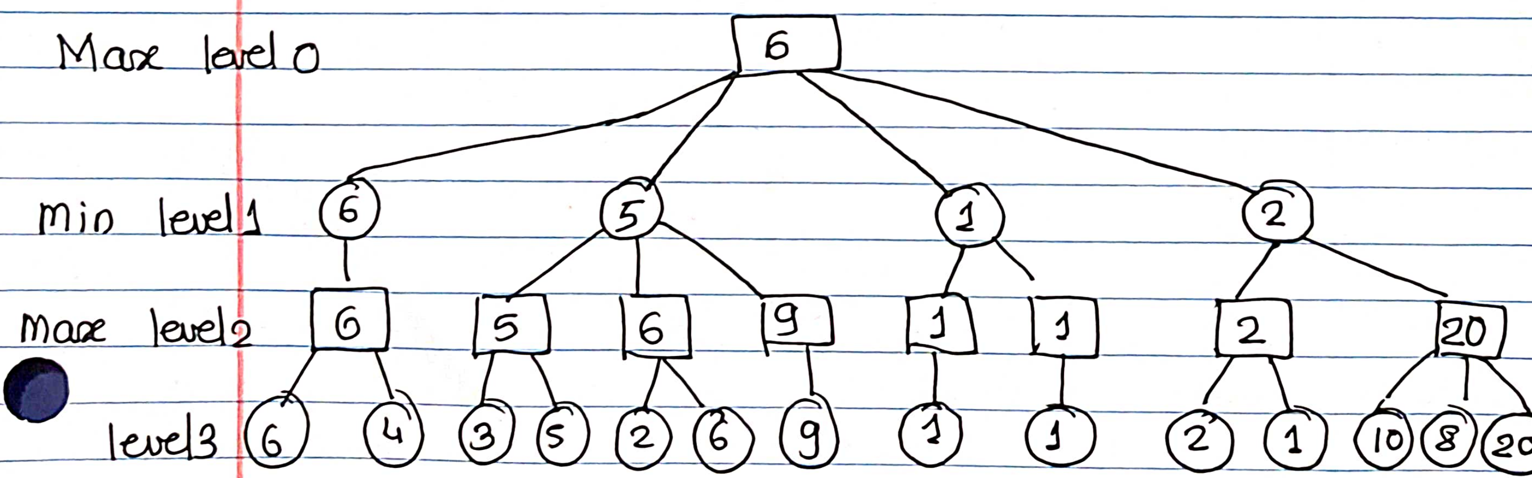
Problem 2. Consider the Search ~~house~~ tree.



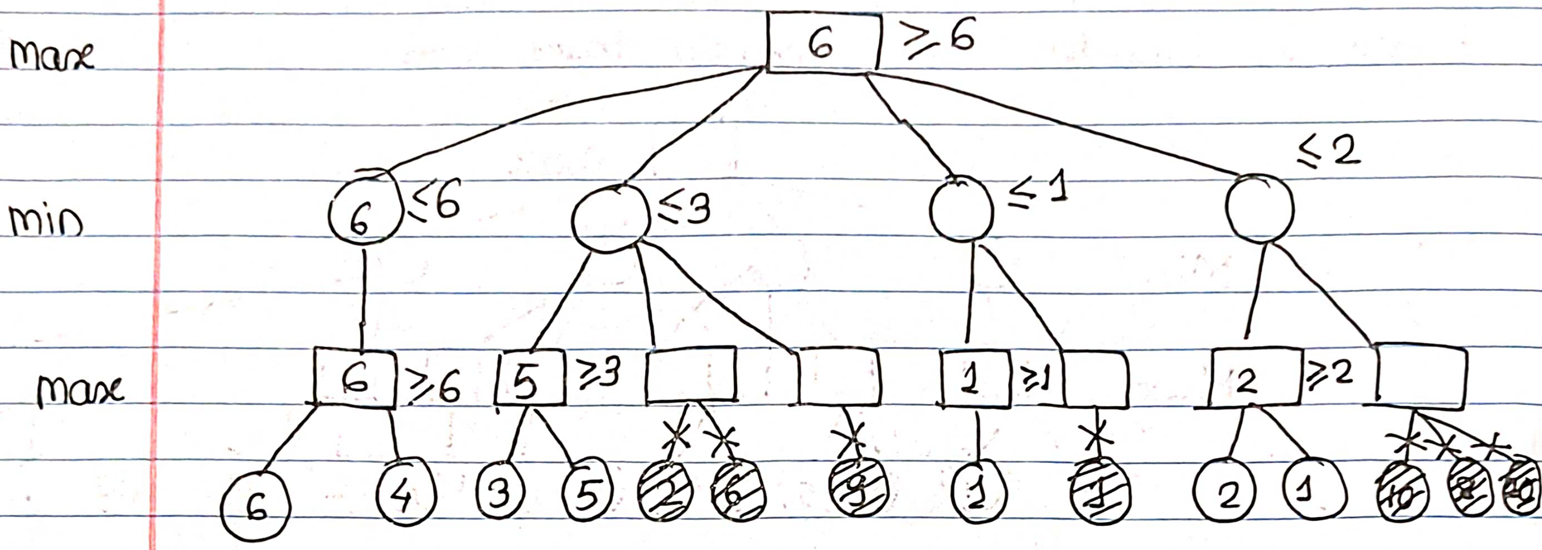
a] fill in Square and circles with backed up values resulting from Regular minimax Search.

→ □ → fill in maximum values among the leaf nodes at level 3 & level 1.

○ → fill in minimum values among the leaf nodes at level 2



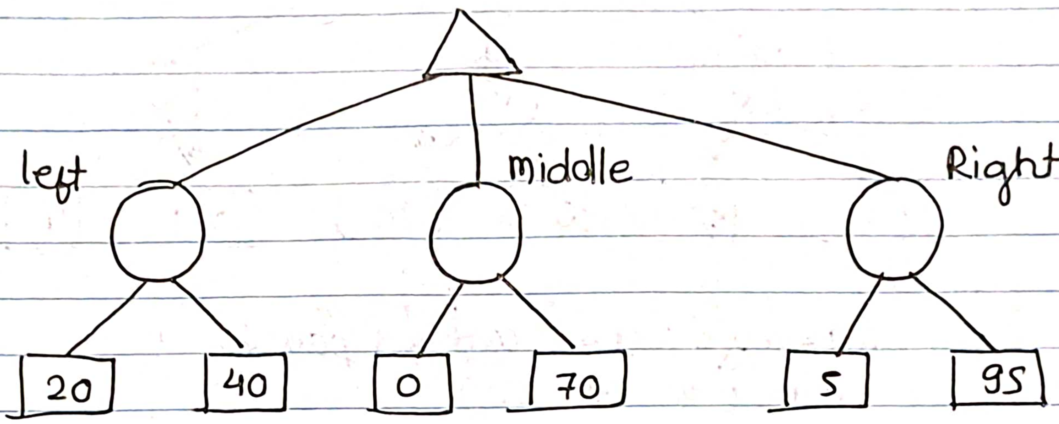
b] What leaf nodes would not be examined using alpha beta Pruning algorithm.



→ The cross marked leaf nodes are removed, they do not need to be evaluated. using alpha beta Pruning we have got 2, 6, 9, 1, 10, 8, 20 nodes which are not required to be evaluated.

c] The value of Root Node is '6'.
using minimax algorithm we get value of Root Node is 6.

Problem 3 Consider game tree, below where terminal values are the Payoff of game. fill in expectimax value of each node, assuming the Player 1 is maximizing expected Payoff and Player 2 Plays uniformly at Random.



a] What is Player 1's expected Payoff if she takes expectimax optimal action

→ Using the expectimax calculating the values for calculating we consider the chance node value will be Probability of both the node hence

for Chance Node = Value 1 x(0.5) Probability + Value 2*0.5

$$= \frac{\text{Value1} + \text{Value2}}{2}$$

① calculating for left move

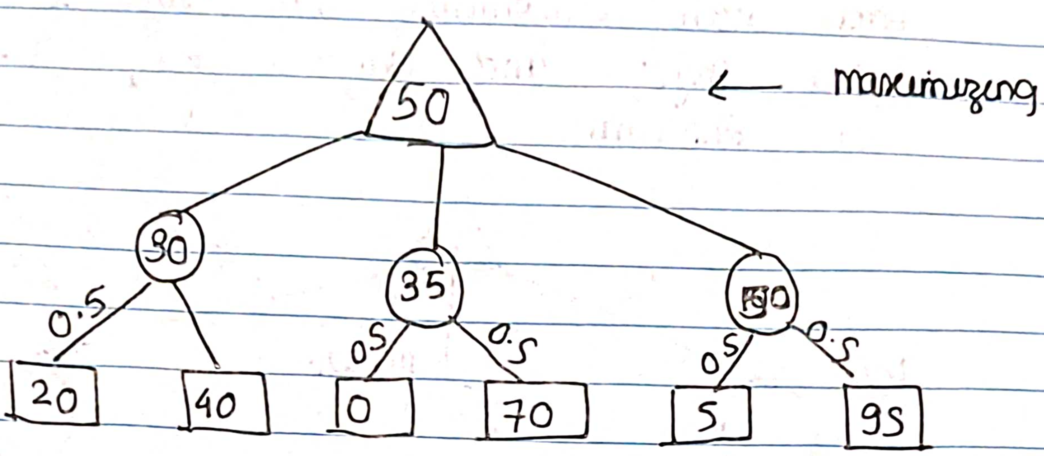
= 20 * 0.5 + 40 * 0.5

= 30

② middle node = 0 * 0.5 + 70 * 0.5

= 35

Right Node = $5 * 0.5 + 95 * 0.5$
= 50



To maximize the expected payoff
max (30, 35, 50)
= 50.

b. multiple customers are possible from player 1's expectimax play. what is worst possible payoff could see from that action ?

→ The worst possible payoff would be getting minimum value.
We get minimum value will be by selecting the middle. The worst in that case we will get is 0 or either 70.
But as player we can see its expectimax play
→ hence players 1's 1 expected payoff is the expectimax optimal action, as maximizing expected payoff max (30, 35, 50) = 50.
So By that worst possible payoff she could see from that action is taking Right Path: hence has equal chance of getting '5' & '95'
worst possible value is '5'

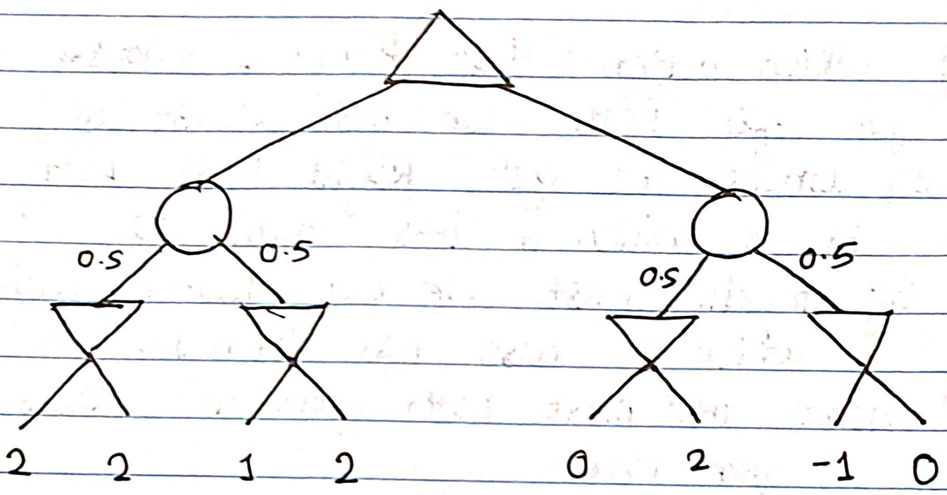
c] Even if average outcome is good, Player 1 doesn't like that very bad outcomes are possible. therefore, rather than purely maximizing expected Payoff using expectimax, Player 1 chooses to perform a modified Search. In particular she only whose worst case outcome is 10 or better.

C.1] Which action does Player 1 choose for this tree?
→ for left Path we get worst case more than 10
By others we get Right Most Path give worst As 5 which is less than 10.
By middle most we get worst outcome as 0 which is also less than 10.
Hence left most path gives us value more than 10 for worst.

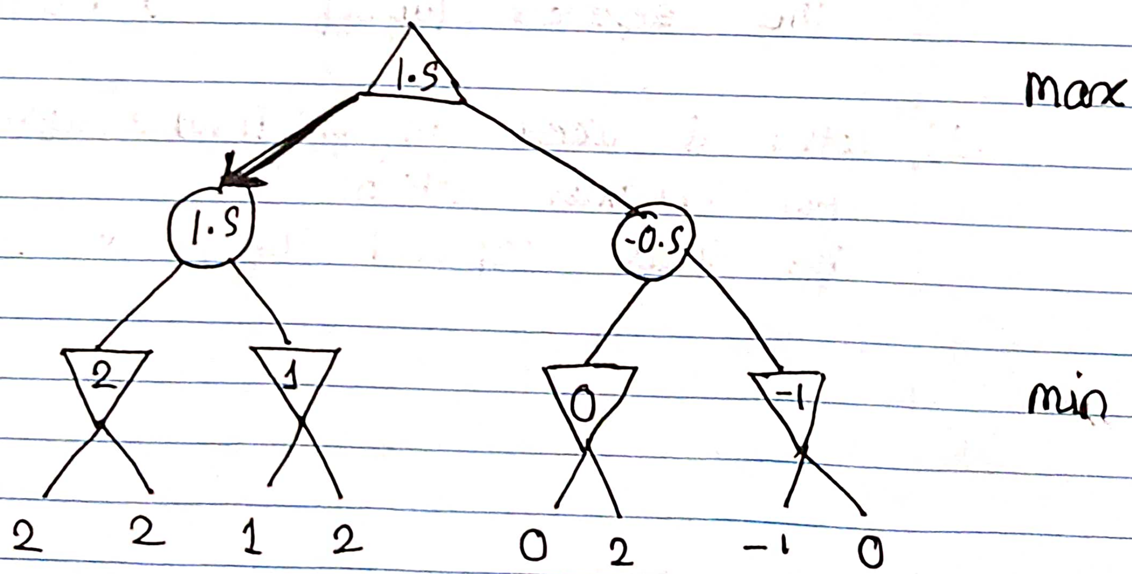
C.2] What is expected Payoff for that action?
→ for left most action / node
The expected payoff is $20 * 0.5 + 40 * 0.5 = 30$.

C.3] What is worst Payoff Possible for that action
for leftmost action
The worst expected Payoff is 20

Problem 4 This Question Considers Pruning in games with chances nodes. Shows the complete game tree for trivial game. Assumes that the leaf nodes are to be evaluated in left-to-Right order and that before a leaf node is evaluated, we know nothing about its value the range Possible value is $-\infty$ to $+\infty$



a] Copy figure, mark the value of all internal nodes and indicate the best move at root with arrow



a] All values are marked on internal node for best move at root node (maximizing the Payoff value) is left branch Indicated in tree above with arrow.

4.b] Given values of first Six leaves, do we need to evaluate the Seventh & Eight?
given Value of first Seven leaves do we need to evaluate the Eight leaf?

→ With first Six leafs we get the Idea of one half of the node above that and the other two leaf nodes can have any value Ranging from $-\infty$ to ∞, thus we need to evaluate further.

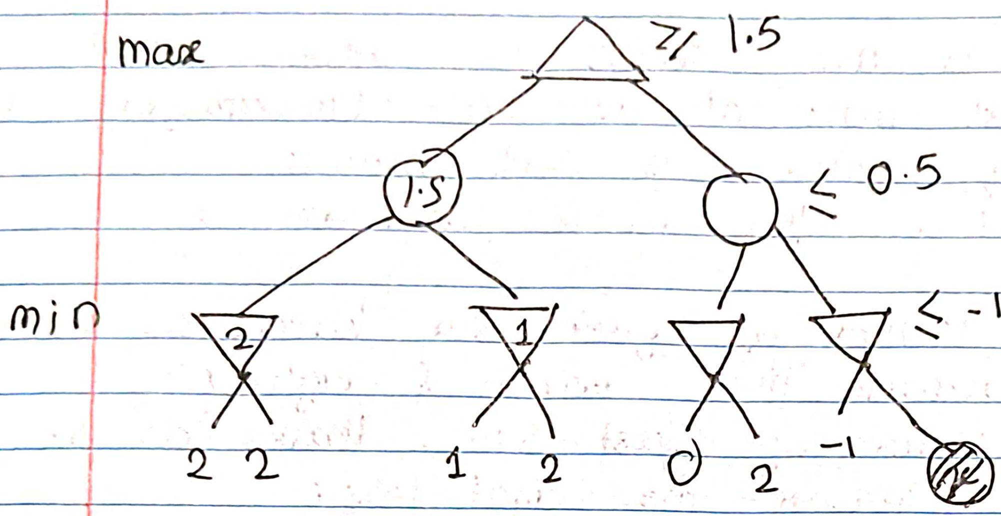
If value of first Seven (7) nodes are given as node 7 & 8 share common parent.
The min will be less or equal to '-1'.
i.e The chance node will be

$[0 * 0.5 + x * 0.5] = x/2$.

where $x \le 1$ because of minimum

hence node's value can be at max (-0.5) & value at root node won't affect much at root node's Position as it is Maximizing its Pay off and thus Plans to have value greater than 1.5.

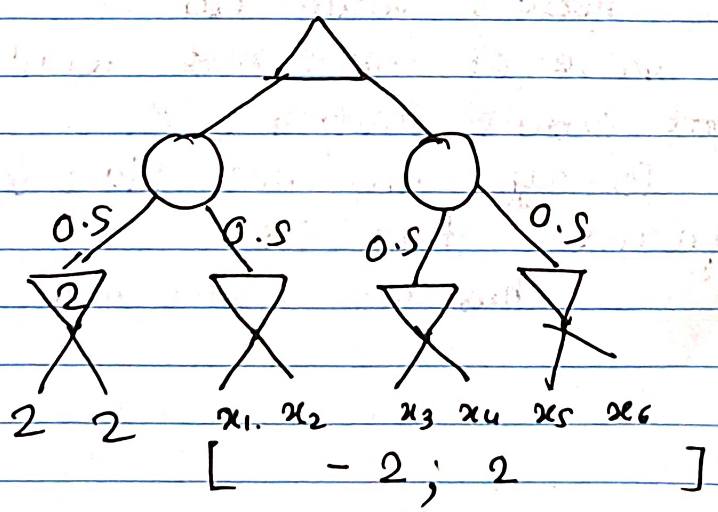
Thus we won't need to evaluate 8^{th} node



As right node was by any way gonna be less than -1 & 0.5 it would never have affected the root node.

c] leaf node value are known to lie between -2 and 2. After first two leaves are evaluated what is value range for left hand chance node?

→ The value of first two leaf nodes stays the same. The value of 3rd, 4th, 5th, 6th, 7th, 8th lie between [-2, 2].



Now we Assume value of 2^{nd} min node is 'x'.

$\therefore$ chance Node for left side $= (0.5 * 2) + (x \times 0.5)$

$= [1 + x/2]$

given we know value of x lies between -2 to 2 hence

Chance Node can have two values

Chance Node (min) $= [1 + (-2)/2]$
$= 0$

Chance Node (max) $= [1 + 2/2]$
$= 2$.

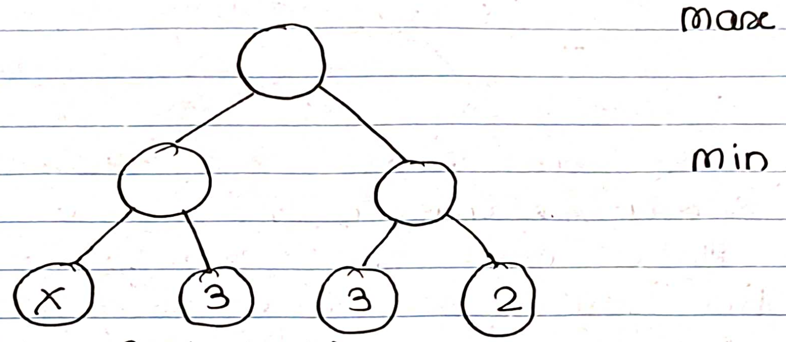
Hence for given value of chance Node of left tree are [0 & 2]

hence value can be anything between 0 & 2 for left chance Node

4.d] Circle all leaves that need not be evaluated under assumption in question c. (i.e the leaf node values are known to lie -2 & 2 inclusive)

→

Problem 5 for the following game tree, we have Prior information that all nodes values are integers that are ≥ 1 and ≤ 6. find all values for X that require the algorithm to visit all leaf nodes of game tree. we assume that alpha - beta pruning perform a depth - first Search that always generates the leftmost Successor first



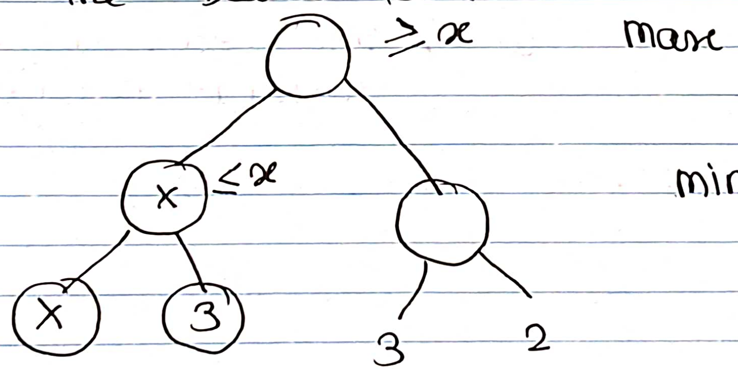
give $x \in (1, 2, 3, 4, 5, 6)$

→ for mini-max Algorithm.

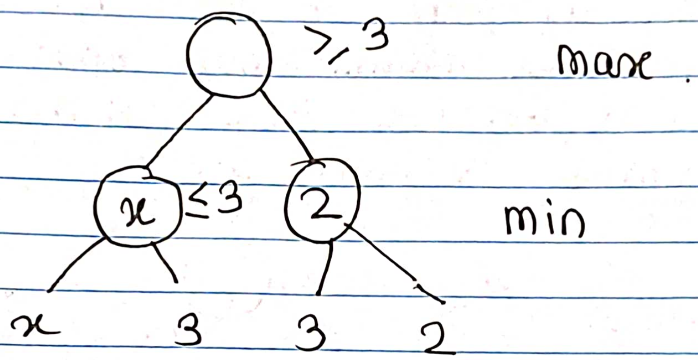
we take the minimum node from two leftmost node

So for node x.

we get the Search tree.

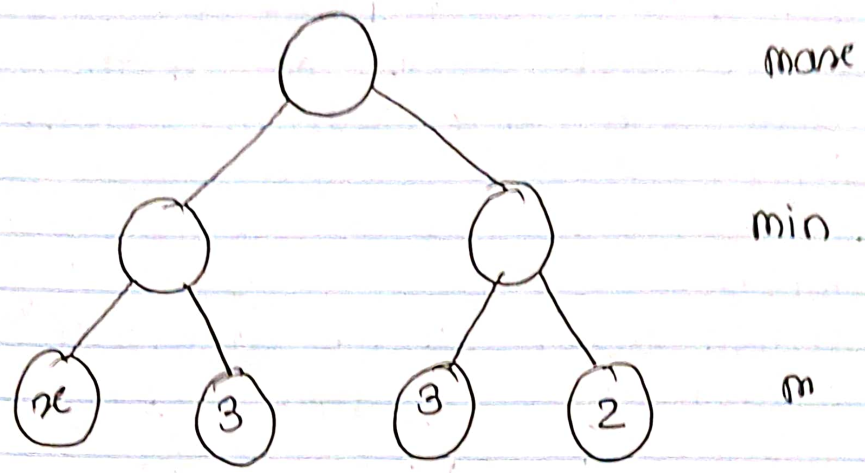


moving on we get node value as 3. as we don't know the Value we will let it Preceed moving one we will have the Search tree the value of x should be less than 3 as anyway the Player will be taking minimum value. hence correct value of x should be $x \leq 3$.

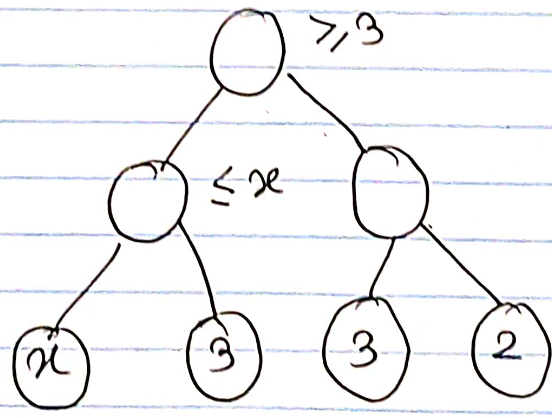


for the root node the Player takes value from $x \in (1, 2, 3)$.
As Player need to maximize it the value of Root node will be 3/2 depending on the value of x.
So the value of $x \in (1, 2, 3)$
The maximum &

Problem 5 for following game tree, we have Prior information that all nodes value are Integers that are ≥ 1 & ≤ 6. find all values for x that require algorithm to reach all the nodes.



→ We are given $x \in [1, 2, 3, 4, 5, 6]$
we need to find out the value of x
for which we will visit all the nodes.



for $x = 1$.
we get the search tree with minimum node = 1
& maximum node any node greater than 1
hence for this value it will search for few
the nodes of the tree. It will Prune few no

for $x = 2$.
It will Search for all the nodes, it won't Prune any nodes.
As it will Search till last for best Path.

for $x = 3$
It will Not Search for all the nodes, as it will find best Path in Second traversing.
hence $x < 3$.

for $x > 2$.
It will choose all the Path. with Right Branch to Reach the best Path.

for $x = 4, 5, 6$.
It will Search for all the Paths to find the best Path. hence will go to all the nodes