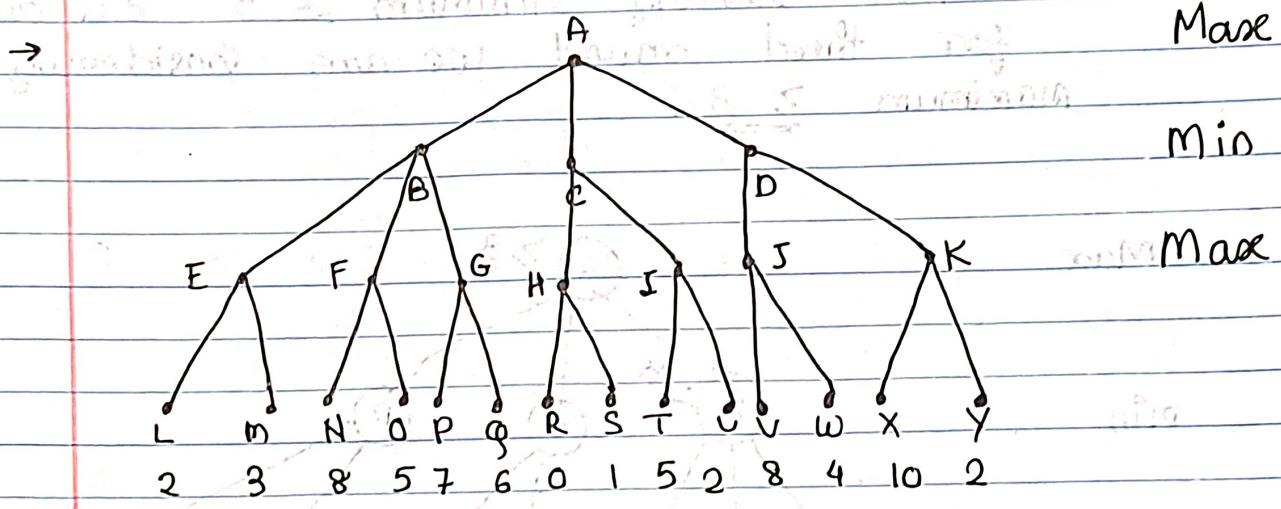


Name : Pauravi Nagarkar
JD : W1650209

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



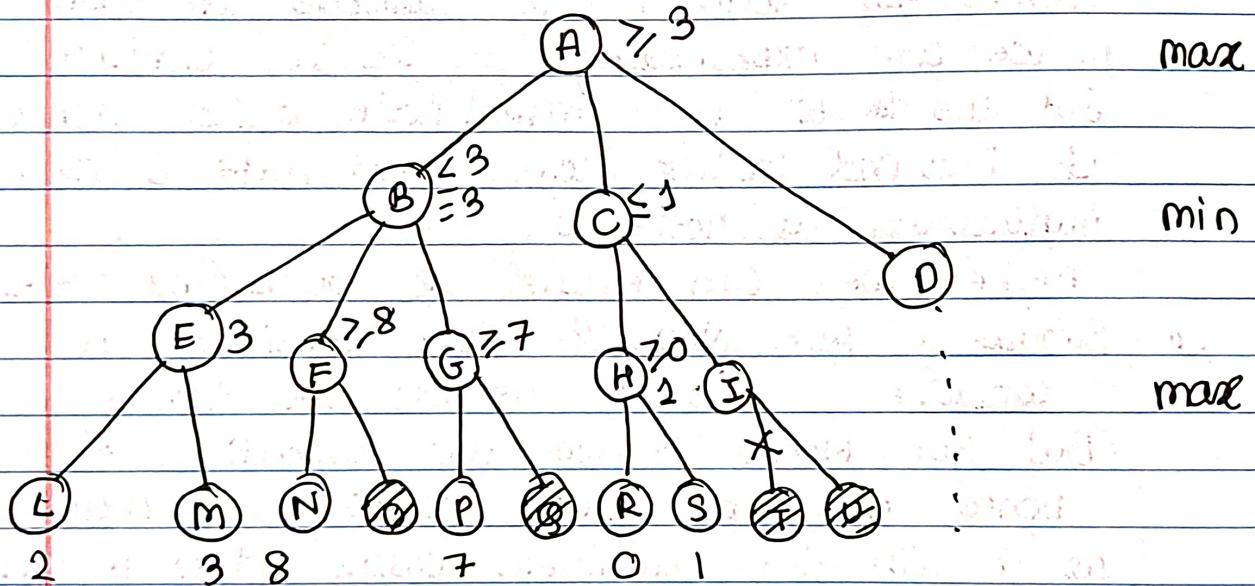
- a] what leaf nodes would not need to be examined using alpha beta pruning algorithm assuming that nodes are examined in left to right order? Show the derivation procedure on graph

→ 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

- 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.

group of nodes hence B will always have value $B \leq 3$.

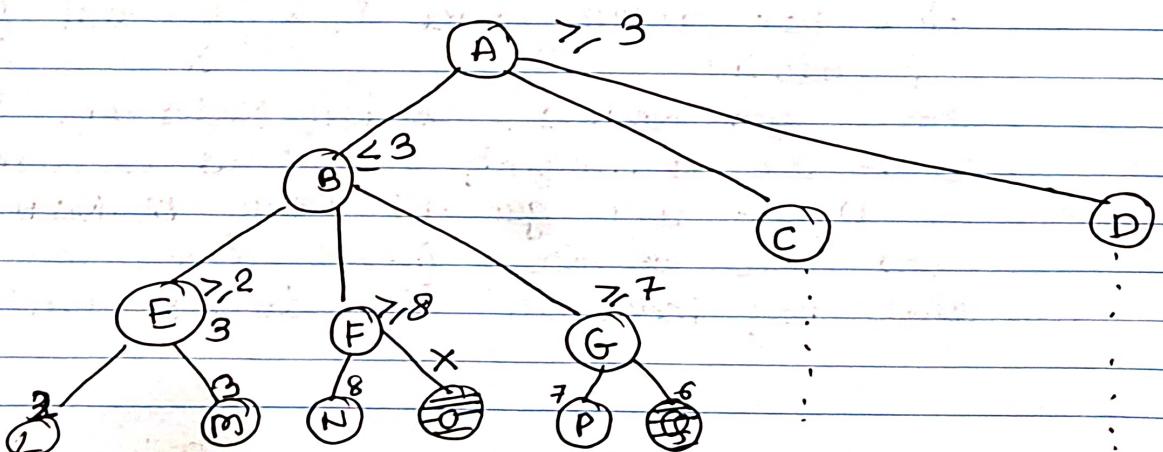
Hence without going from all the nodes we can predict $B \leq 3$ as we initialize minimum value of B amongst Nodes leafs.

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 Initial Info that we have. $A \geq 3$.

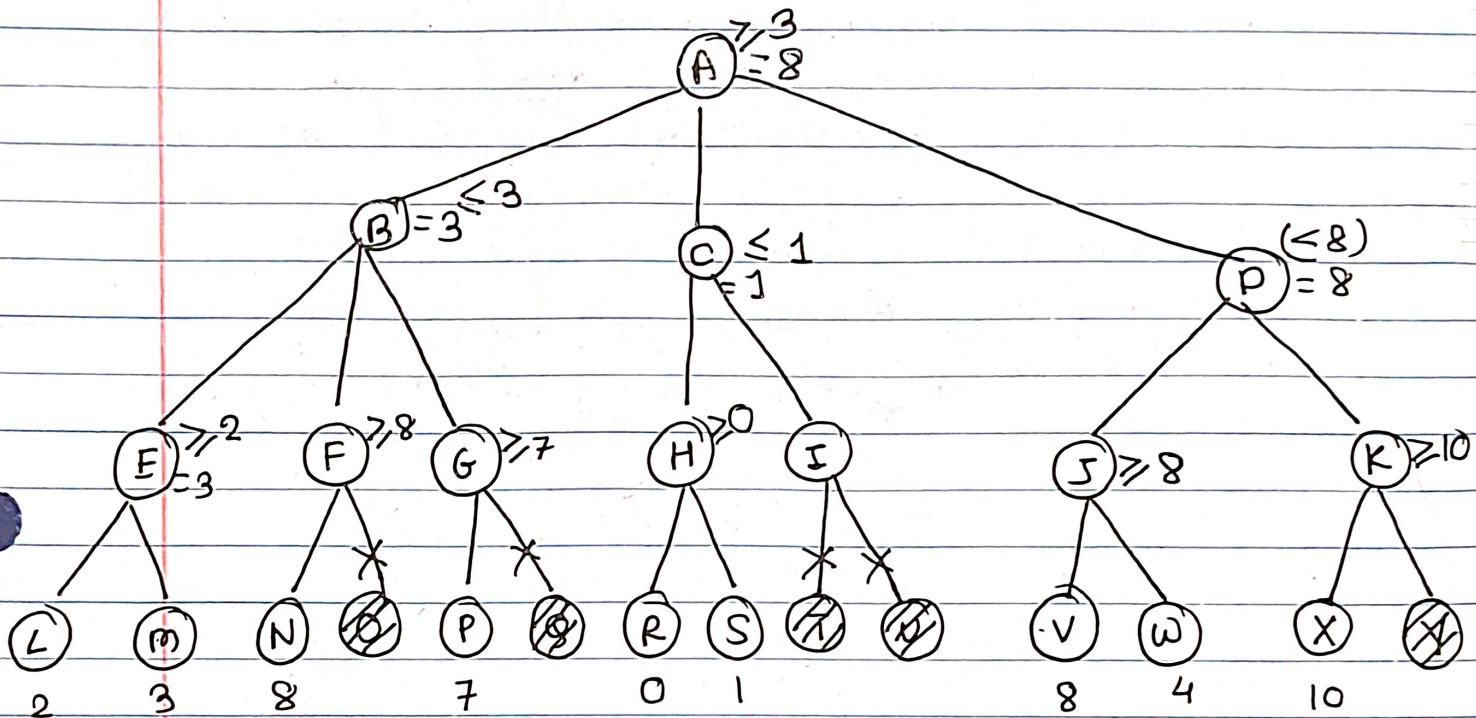
⑥ Consider for Node F we consider its node N: 8 as we are maximizing it we can consider $F \geq 8$ But we know its Parent node $B \leq 3$ hence it does not matter what next node is as we are minimizing at node B.

hence we can Prune its Succeeding 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.

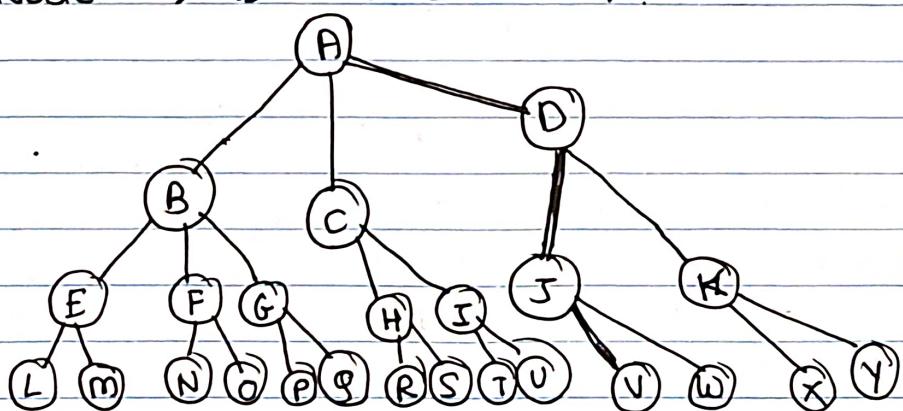


Now $w=4$ hence w can be excluded
 moving to node K . tracing its leaf node X
 with value 10. hence $K \geq 10$. But we are
 minimizing at D hence $D \neq 10$. hence we can
 skip its remaining node.



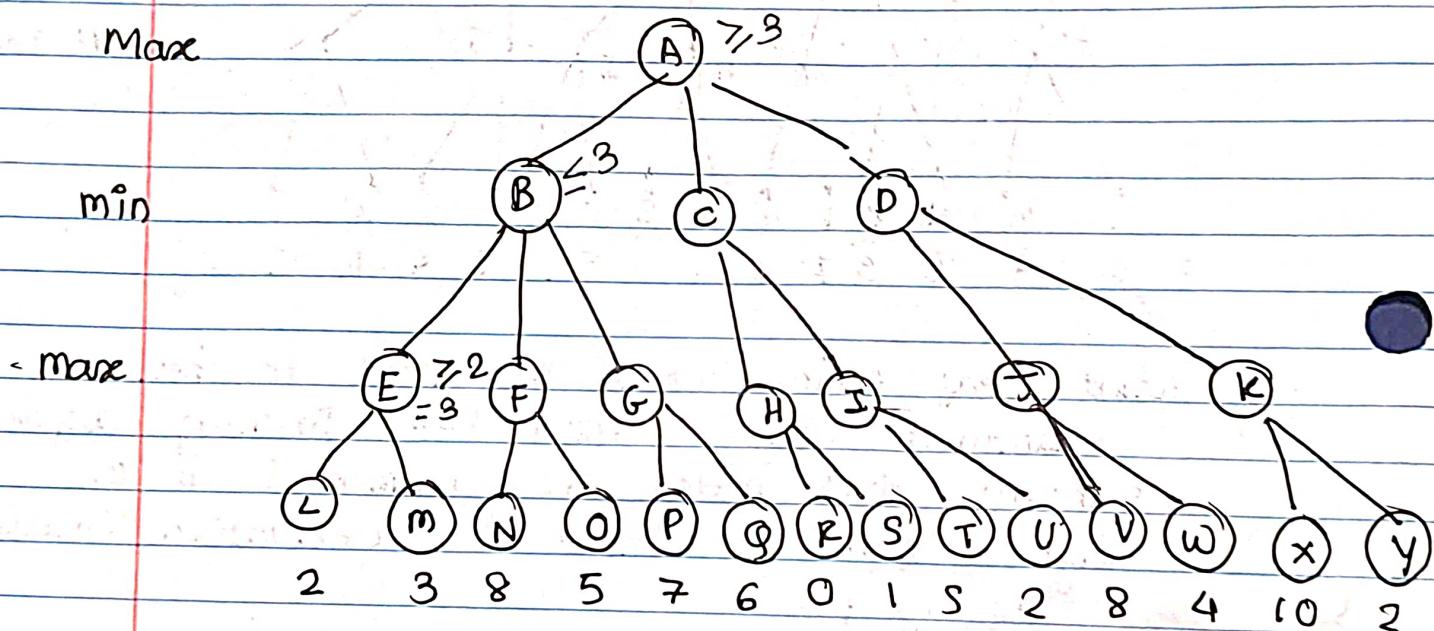
Hence By alpha Beta Pruning Algorithm we get
 that leaf nodes O, Q, T, U, Y can be pruned.

b] As A=8 is optimal actual value of node hence
 first Player Should move towards D. to Right.
 Root Node $\rightarrow D \rightarrow J \rightarrow V$.



To decide which leaves Node to be emitted let's start from lowest/ leftest Node E, m for Node E collect maximum from both.

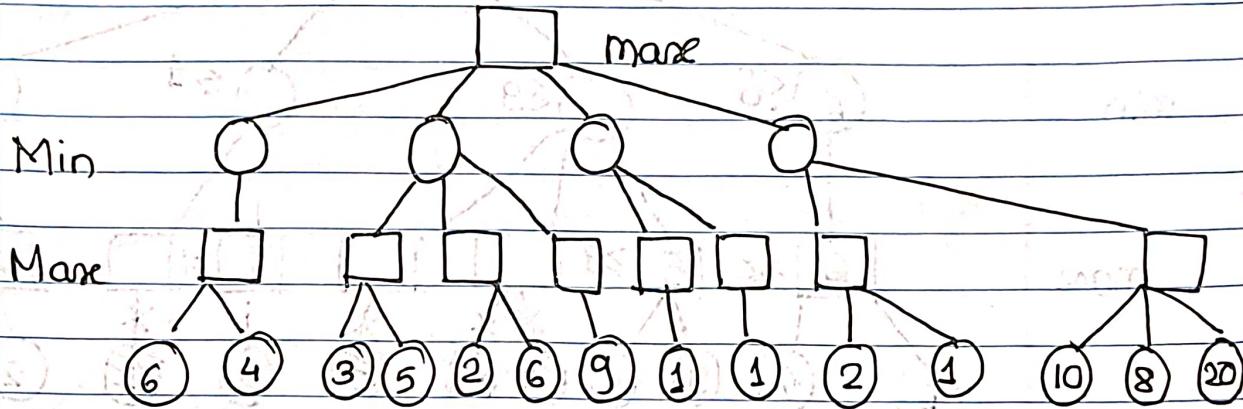
Now maximum node for E but for B we are selecting minimum ≤ 3 But again for third round we are considering maximum ≥ 3 .



- 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 begin 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

c. The final minmax Value for Root Node is 8
 Indicating A → Root Node → 8

Problem 2. Consider the Search tree.

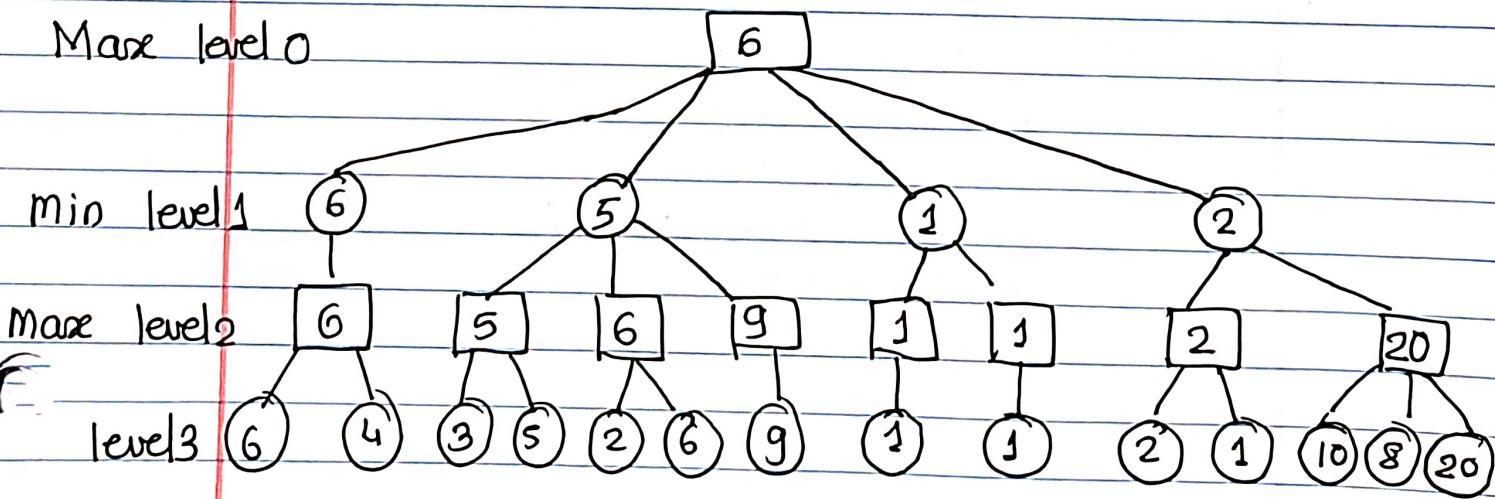


a] fill in Square and circles with backed up values resulting from Regular minimize Search.

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

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

Maze level 0

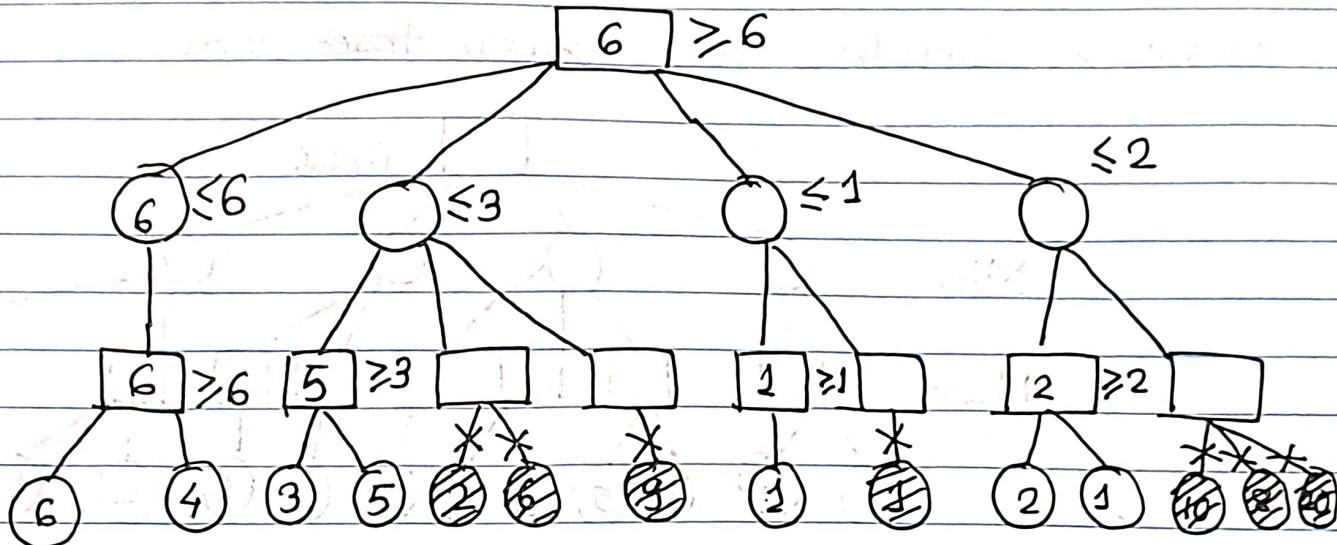


b] What leaf nodes would not be examined using alpha beta Pruning algorithm.

max

min

max

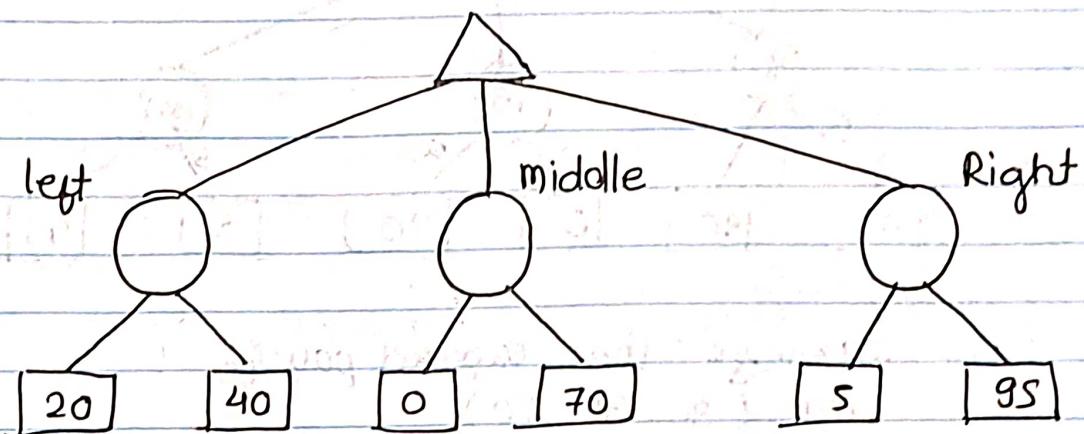


→ The cross marked leaf nodes are removed, they do not need to be evaluated. Using alpha beta pruning we have got 2, 6, 9, 1, 10, 8, 20 nodes which are not required to be evaluated. 8

c] The value of Root Node is '6'.

Using minimax algorithm we get value of Root Node is 6.

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



a] what is Player 1's expected Pay off if she takes optimal action

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

for chance Node = $\text{Value 1} \times 0.5 + \text{Value 2} \times 0.5$

$$= \frac{\text{Value 1} + \text{Value 2}}{2}$$

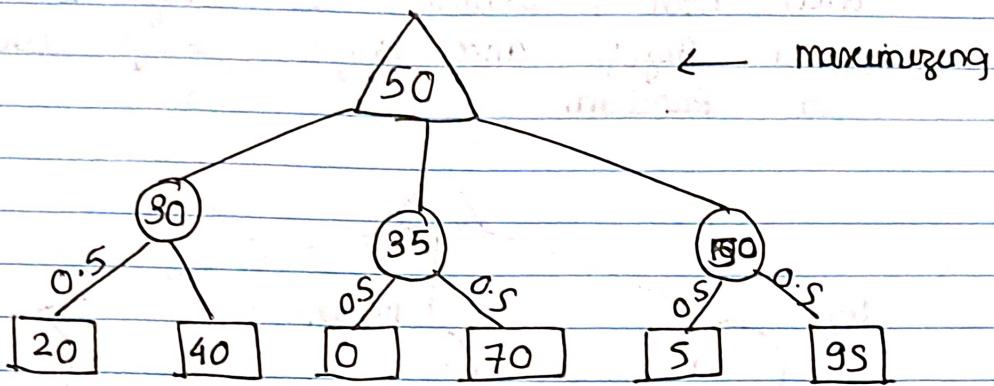
① calculating for left move

$$= 20 \times 0.5 + 40 \times 0.5$$

$$= 30$$

$$\begin{aligned} \textcircled{2} \quad \text{middle node} &= 0 \times 0.5 + 70 \times 0.5 \\ &= 35 \end{aligned}$$

$$\begin{aligned}\text{Right Node} &= 5 * 0.5 + 95 * 0.5 \\ &= 50\end{aligned}$$



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 Player 1's 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' or '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 expectation, Player 1 chooses to perform a modified search. In particular she only chose 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 of 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.

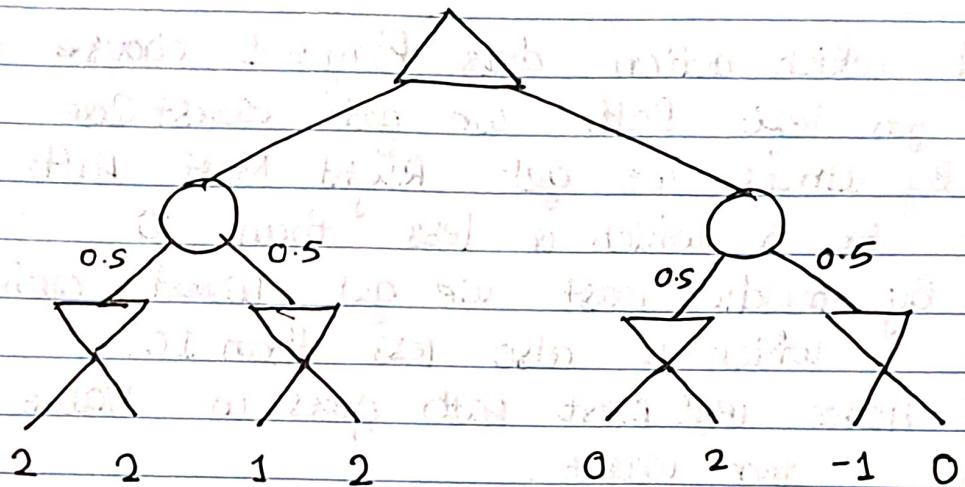
- c.2] What is expected payoff for that action?

→ for left most action / node
The expected payoff is $20 * 0.5 + 40 * 0.5 = 30$

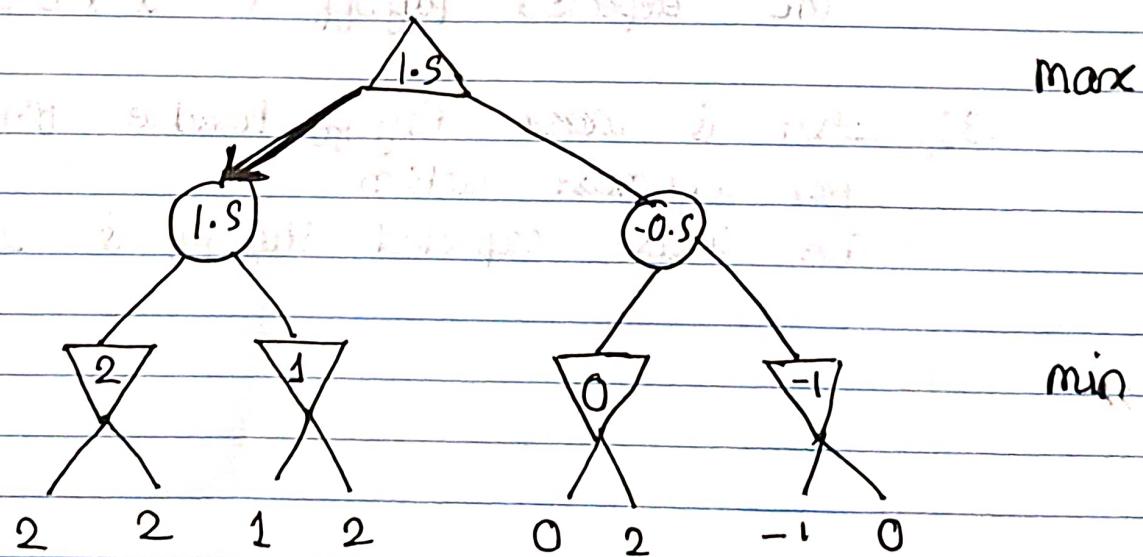
- c.3] What is worst Payoff possible for that action for leftmost action

The worst expected Payoff is 20

Problem 4 This question considers pruning in games with chance 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 & eighth? Given value of first seven leaves do we need to evaluate the eighth leaf?

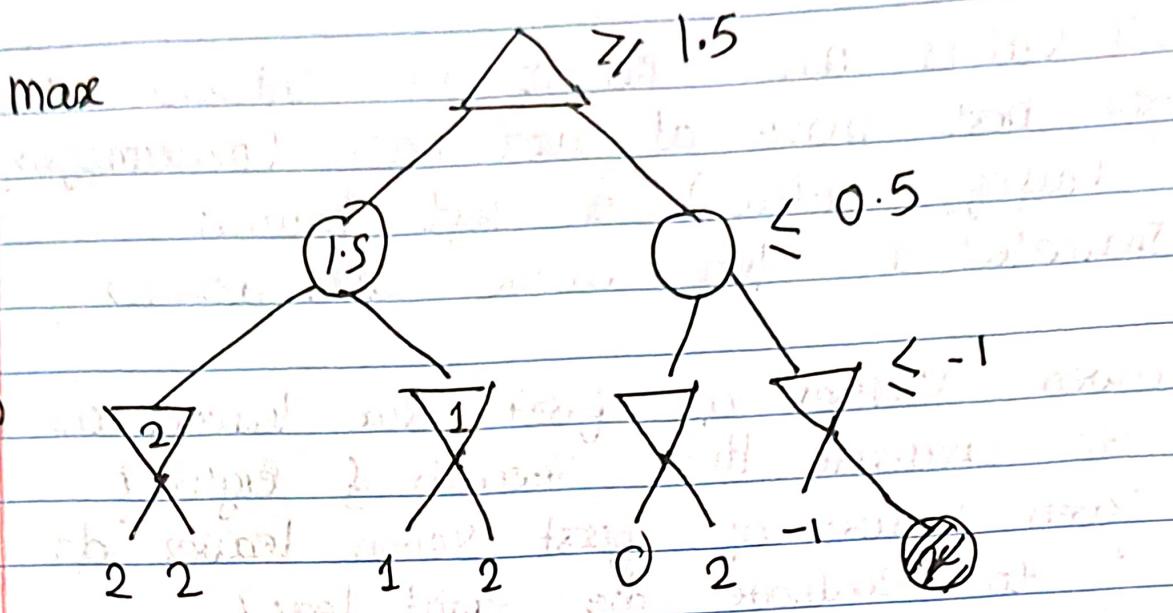
→ With first six leafs we get the idea of one half of the node above that and the other two leaf nodes can have any value ranging from $-\infty$ to ∞ , thus we need to evaluate further.

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 '.

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

where $x \leq 1$. 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 payoff and thus plans to have value greater than 1.5 .

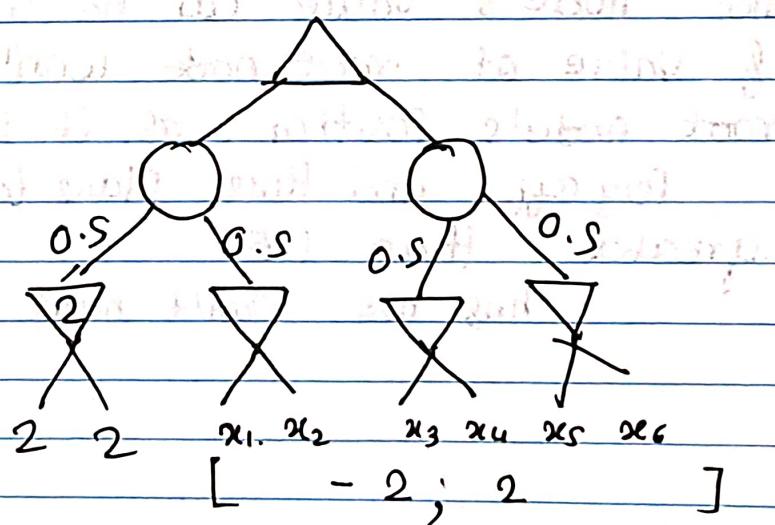
Thus we won't need to evaluate 8th node



As eight node was by any way gonna be less than -1 if 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 2nd min node is x

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

$$= [1 + x/2]$$

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

chance Node can have two values

$$\text{chance Node (min)} = [1 + (-2)/2] \\ = 0$$

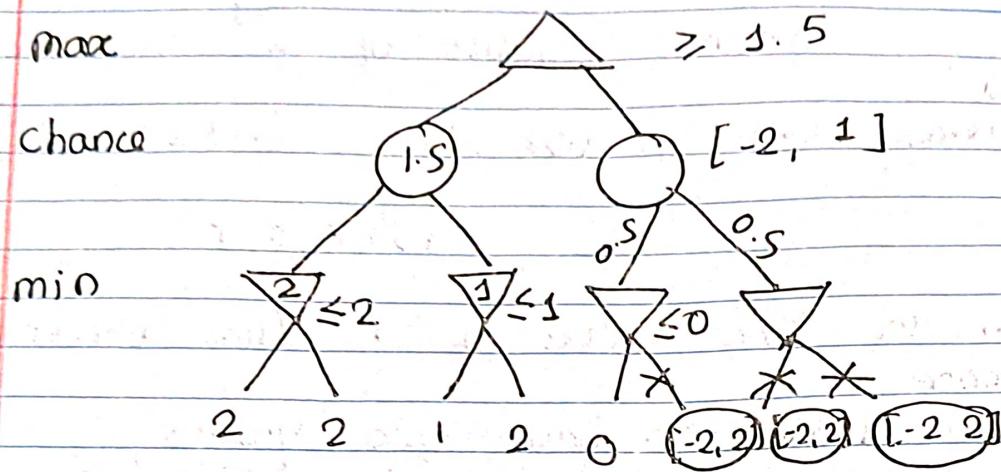
$$\text{chance Node (max)} = [1 + 2/2] \\ = 2$$

Hence for given value of chance Node of left tree there 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)





As shown

The first five nodes are provided with value.

When the fifth node is evaluated i.e '0' it gives an idea of the 'min' value i.e ' ≤ 0 '

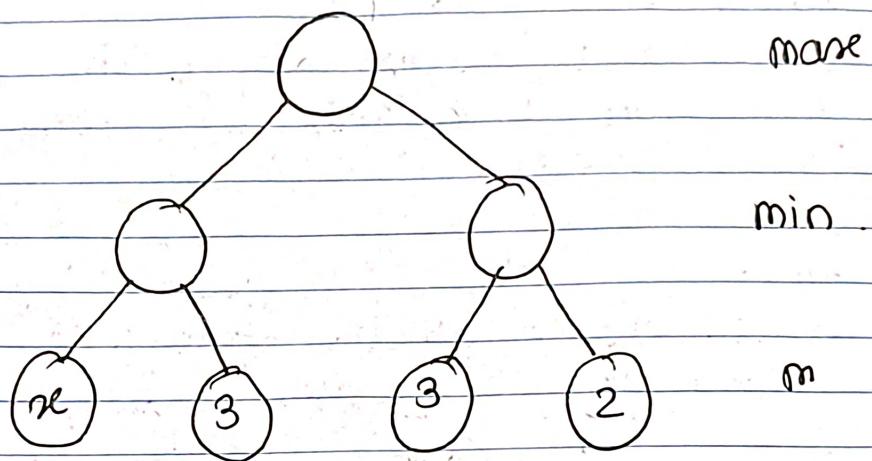
The other leaf nodes value lie between $[-2, 2]$. Even if maximum value '2' is evaluated for three leaves, the chance node will be $(0 * 0.5 + 2 * 0.5) = 1$.

The minimum value of chance node will be -2.

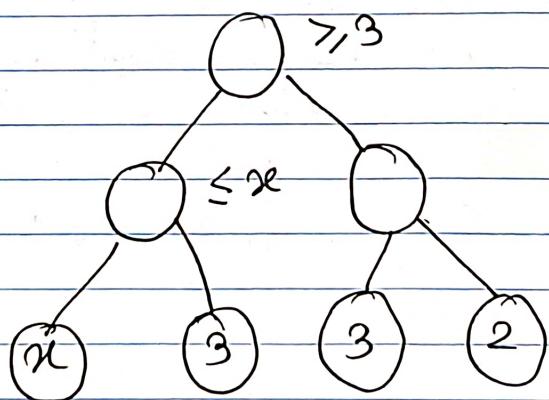
which does not go well with root node equation of ≥ 1.5 .

hence we are okay with last three nodes not being evaluated.

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 requires 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 ~~all~~ the nodes of the tree. It will prune few nodes.

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