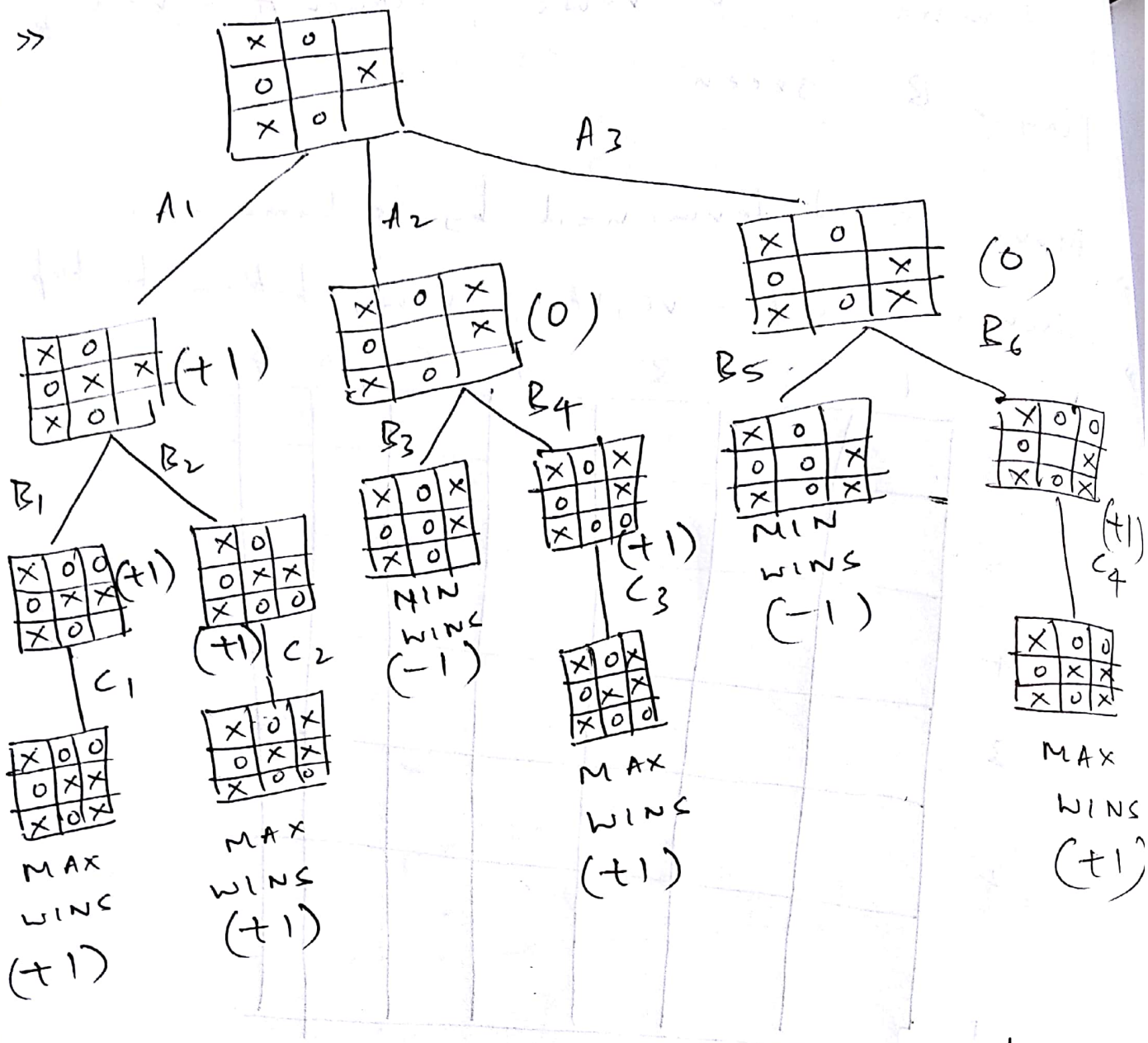
Problem-1

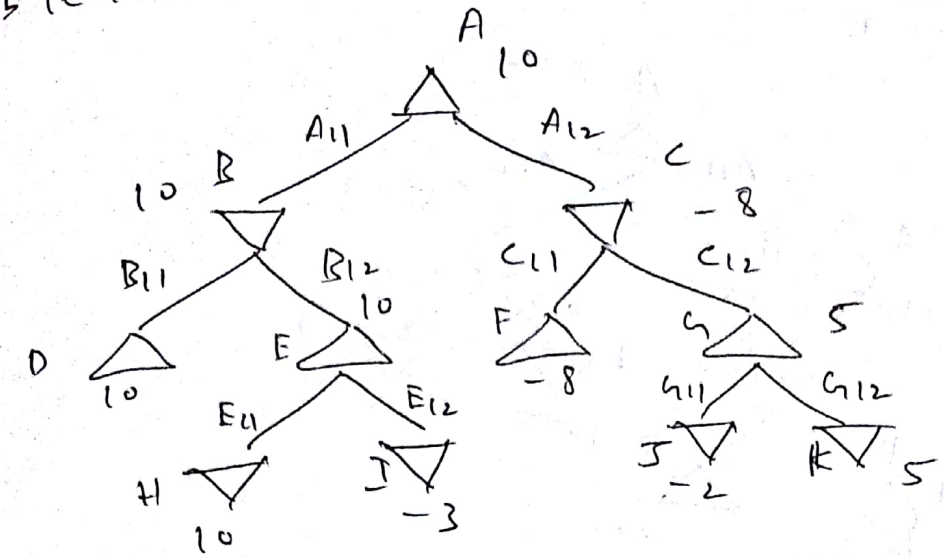
»



» Minmax algorithm will decide to play A_1, B_1 / B_2, C_1 / C_2. since B_1 & B_2 & C_1 & C_2 have the same utility value.

» Utility value of non-terminal node. & the terminal nodes are shown in the diagram

Problem-2



» Node I [E_{12}] & node K [G_{12}] will be pruned.

» A = 10 ; B = 10 ; C = -8 ; D = 10 ; E = 10 ; F = -8 ; G = 5 ; H = 10 ; I = -3 ; J = -2 ; K = 5 are the utility values.

» MAX player will play A_{11} & then it can E_{11} if min player chooses B_{12}.

» If the maximum utility is given, (10) then the whole of right side of the tree i,e node C will be pruned which inturns prunes F, G, J, K since we would have got the highest utility from ~~right~~ left part of the tree.

Problem-3

>> DeepGreen does not use minimax

>> DeepGreen.MoveTo(s) is the function we know

>> The required Pseudo code is

function MINIMAX-DECISION (state) returns an action

inputs : state.

return the a in ACTIONS (state) maximizing DeepGreenMove (state)

function MAX-VALUE (state) returns a utility value

if TERMINAL-TEST (state) then return UTILITY (state)

v ← -∞

for a, s in SUCCESSORS (state) do

v ← MAX (v, DeepGreenMove (s))

return v

function DeepGreenMove (state) returns a utility value

if TERMINAL-TEST (state) then return UTILITY (state)

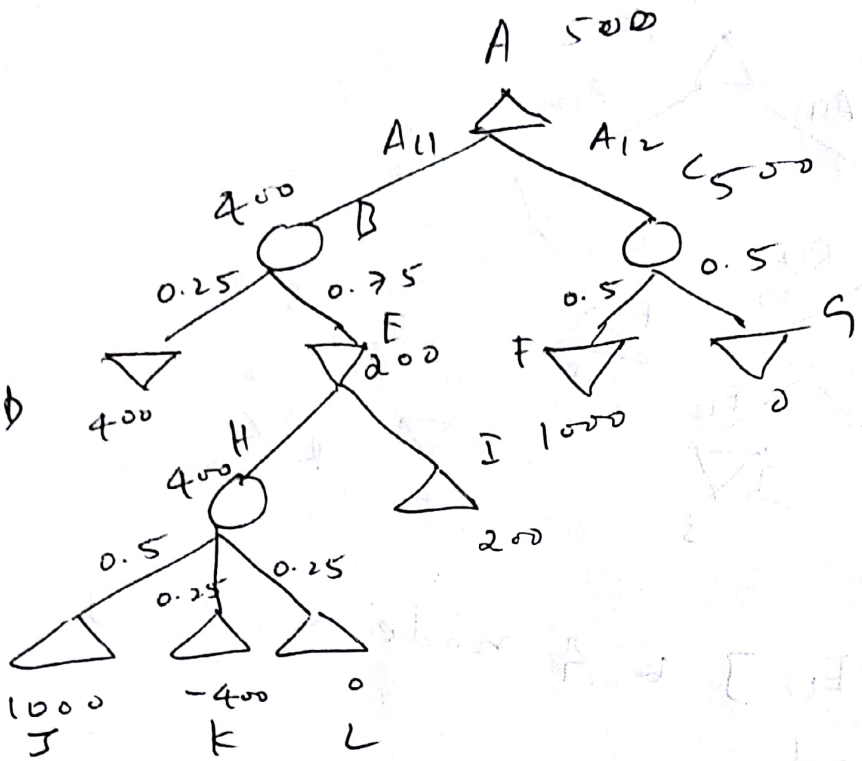
for a, s in SUCCESSORS (state) do

v ← DeepGreenMove (MaxValue (s))

return v.

» Since the DeepGreenMove (s) can provide us enough information about the next move, it is much more optimal to use this algorithm since it will definetly have less time complexity. since it need not calculate the values of the opponent.

Problem - 4



» A : 500 ; B : 400 ; C : 500 ;

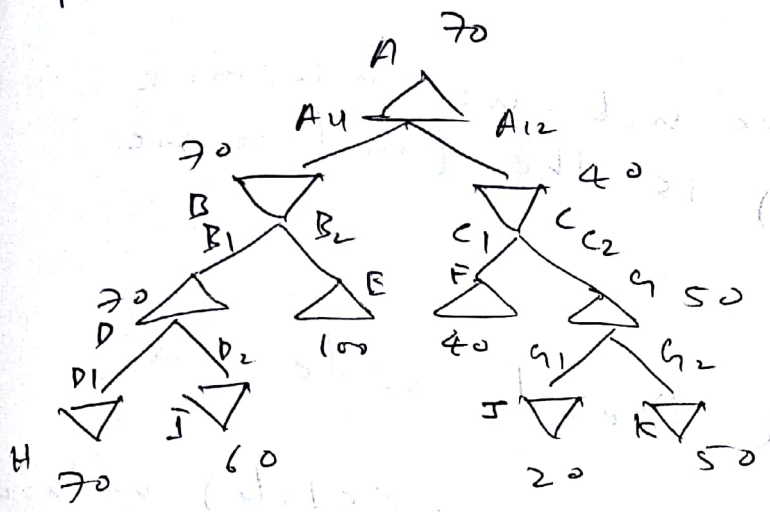
D : 400 ; E : 200 ; F : 1000 ; G : 0

H : 400 ; I : 200 ; J : 1000 ; K : -400

L : 0

» A_{12} will be performed by the algorithm. since it results in maximum utility value of 500.

Problem-5



» Suppose the opponent also uses minimax algorithm, then the best move A could make was A_{11} where he would be guaranteed 70 utility value.

» Best possible outcome might be if B chooses B_2 which results in the utility value of 100 for MAX player.

» Worst possible outcome is 70 since MAX player uses MINIMAX algorithm, he would go with A_{11} since he doesn't know about the opponent's algo. B would choose {70} B_1 which is the worst case.