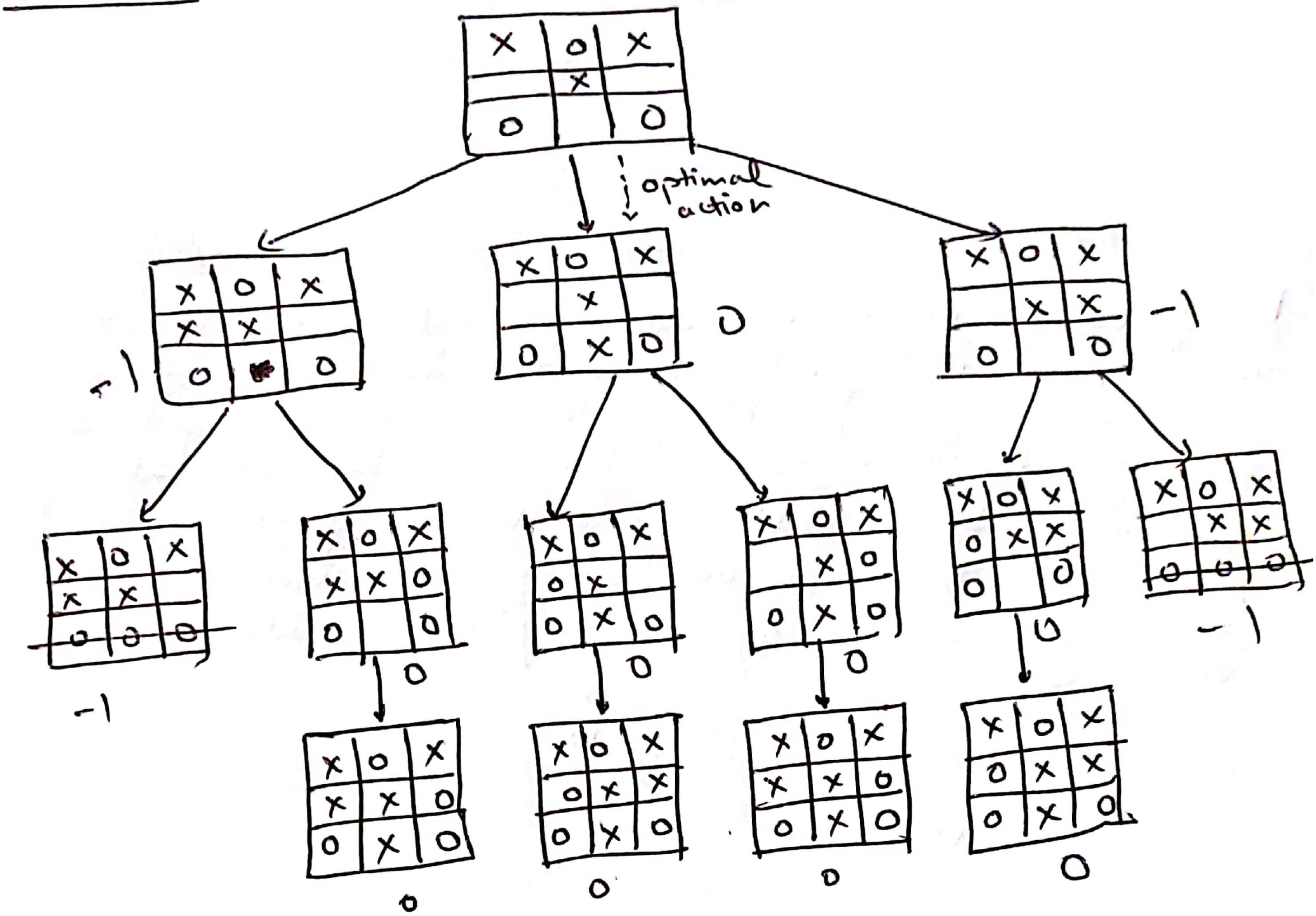
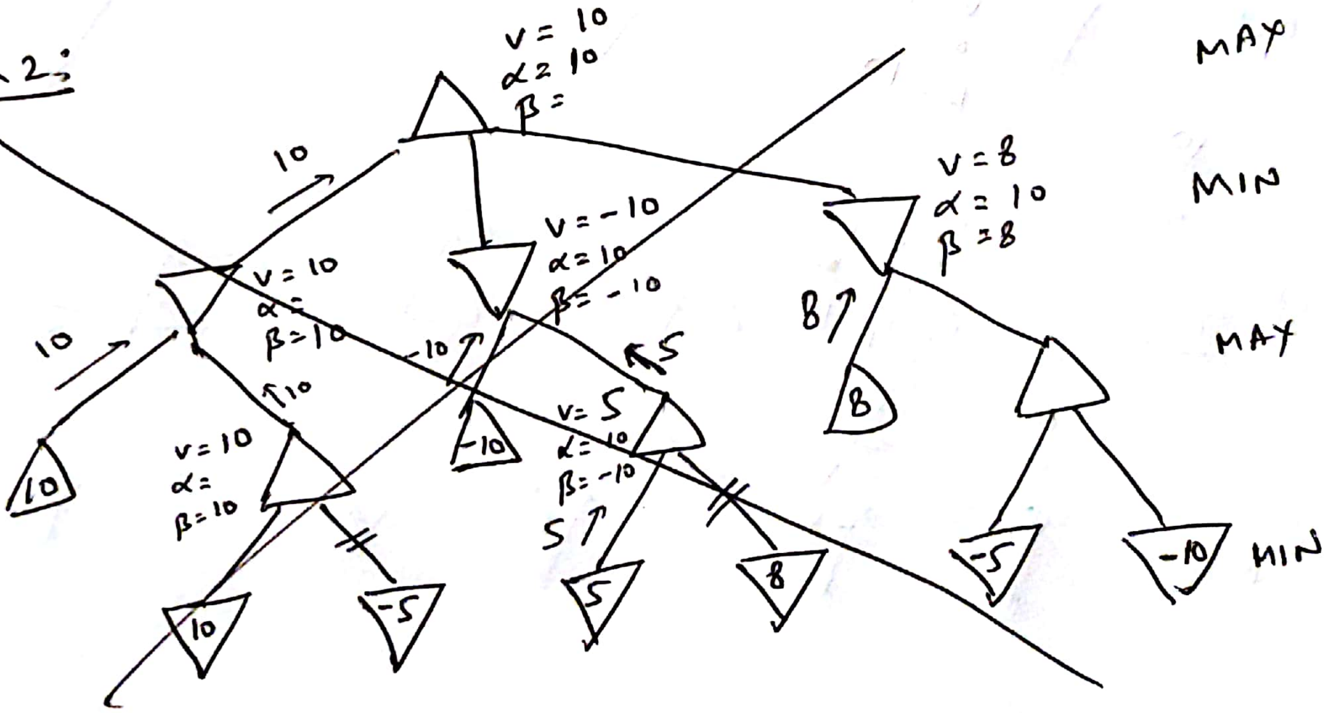
Assignment - 2

Nishant Chaudhary
1001568781

Problem 1:

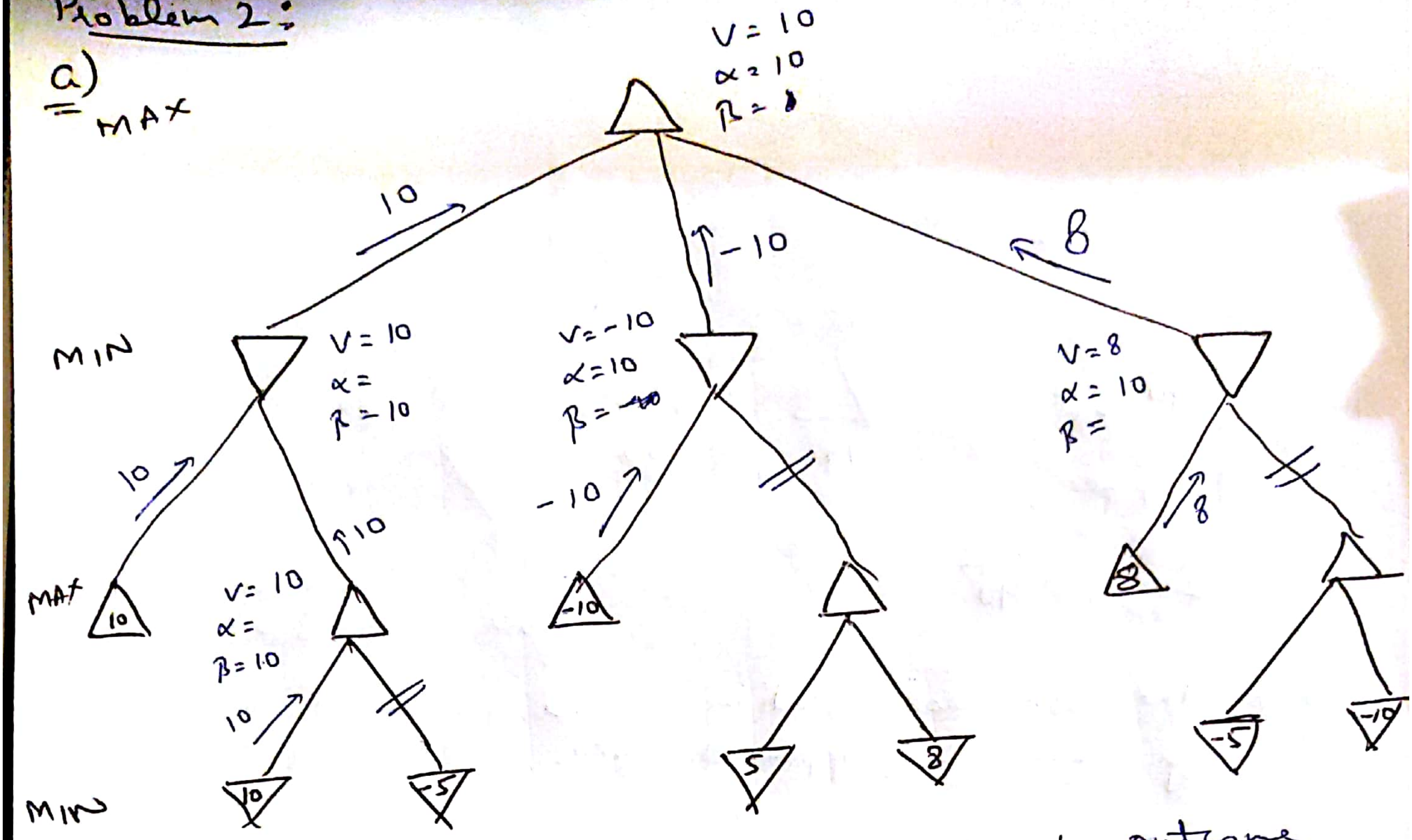


Problem 2:



Problem 2:

a)



α : Max's current lower bound on Max's outcome

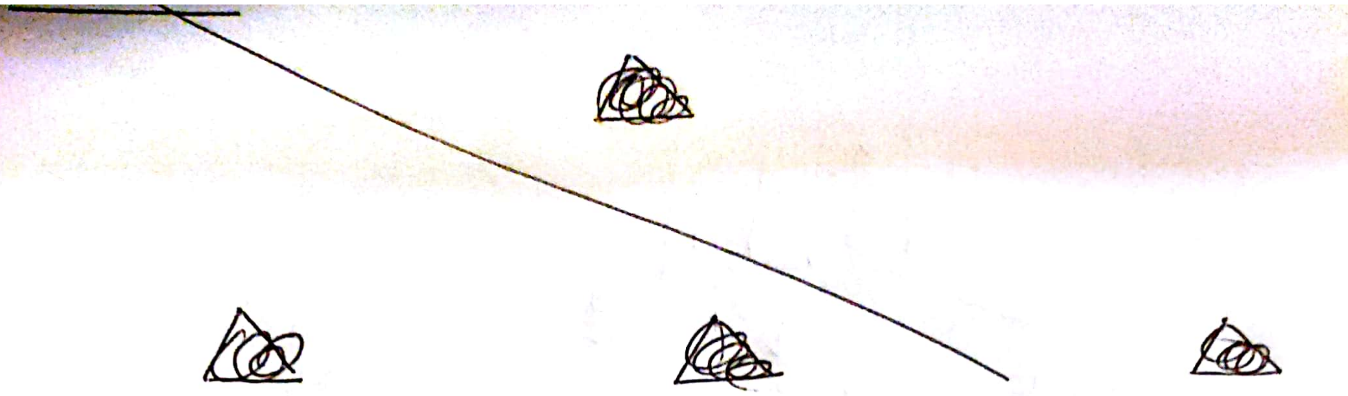
β : Min's current upper bound on Min's outcome

Max will never allow a move that could lead to a worse score (for Max) than α.

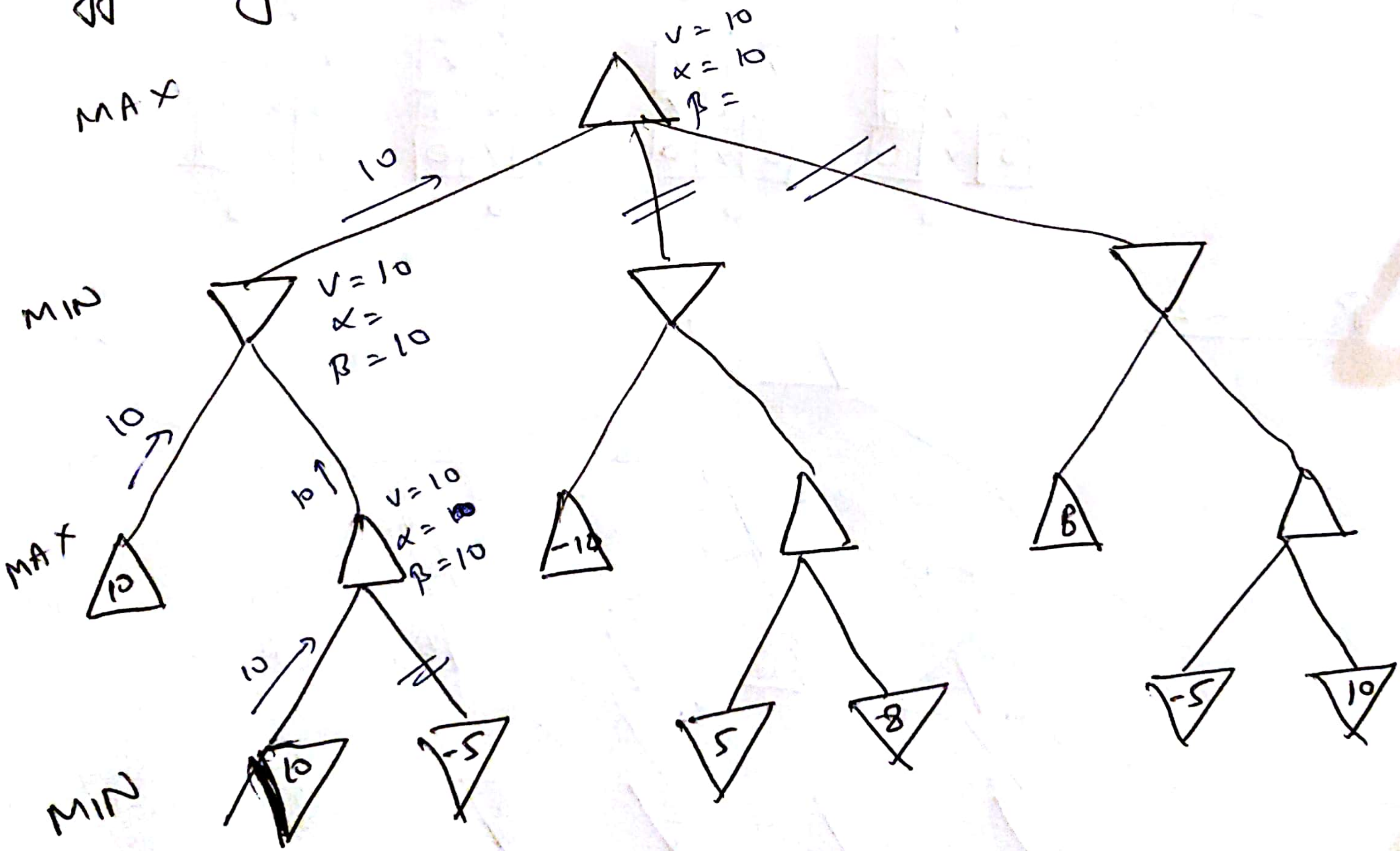
Min will never allow a move that could lead to a better score (for Max) than β.

→ When evaluating Max node: a value V ≥ β is backed-up
- MIN will never select that MAX node.

→ When evaluating MIN node: a value V ≤ α is found
- MAX will never select that MIN node

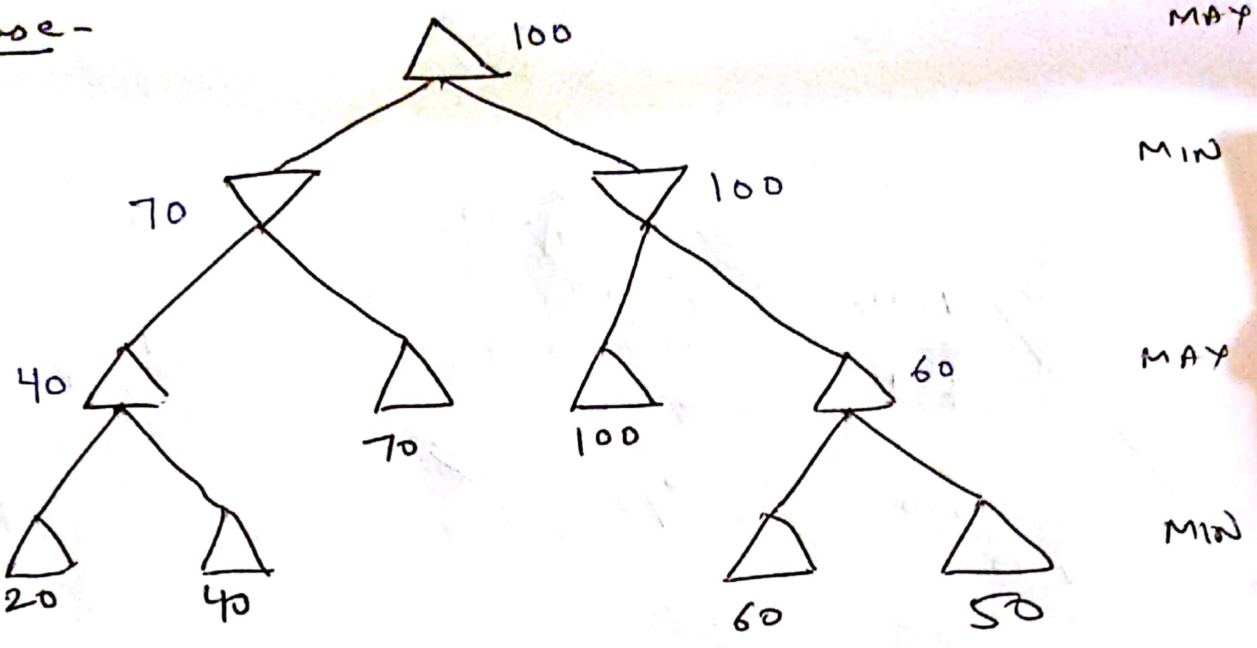


b) If we are given ~~max~~ additional information that the maximum utility value is 10 & the minimum utility -10, then its not possible for the root node to get a value greater than 10. In this case, the ~~left subt~~ leftmost subtree will give a value of 10, so it prunes both the trees on the right. Hence, the efficiency increases by pruning more nodes.

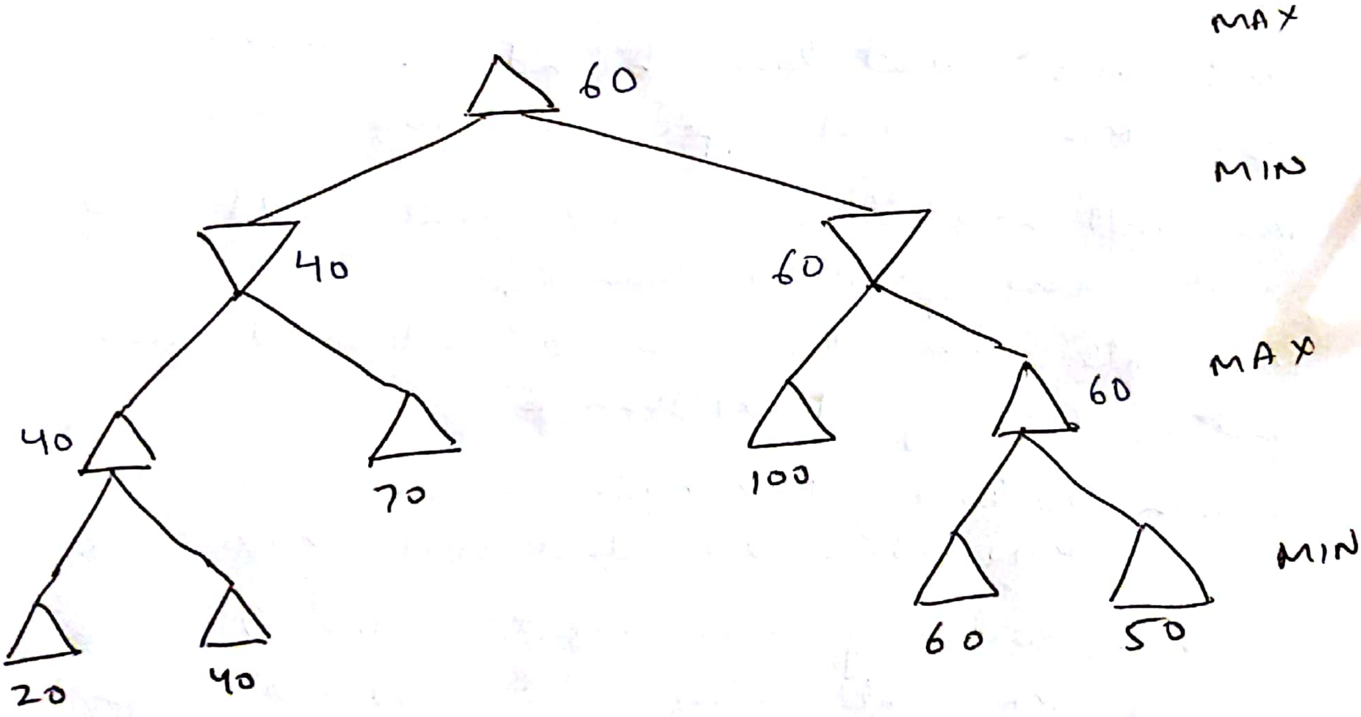


Problem 3:

Best Case-

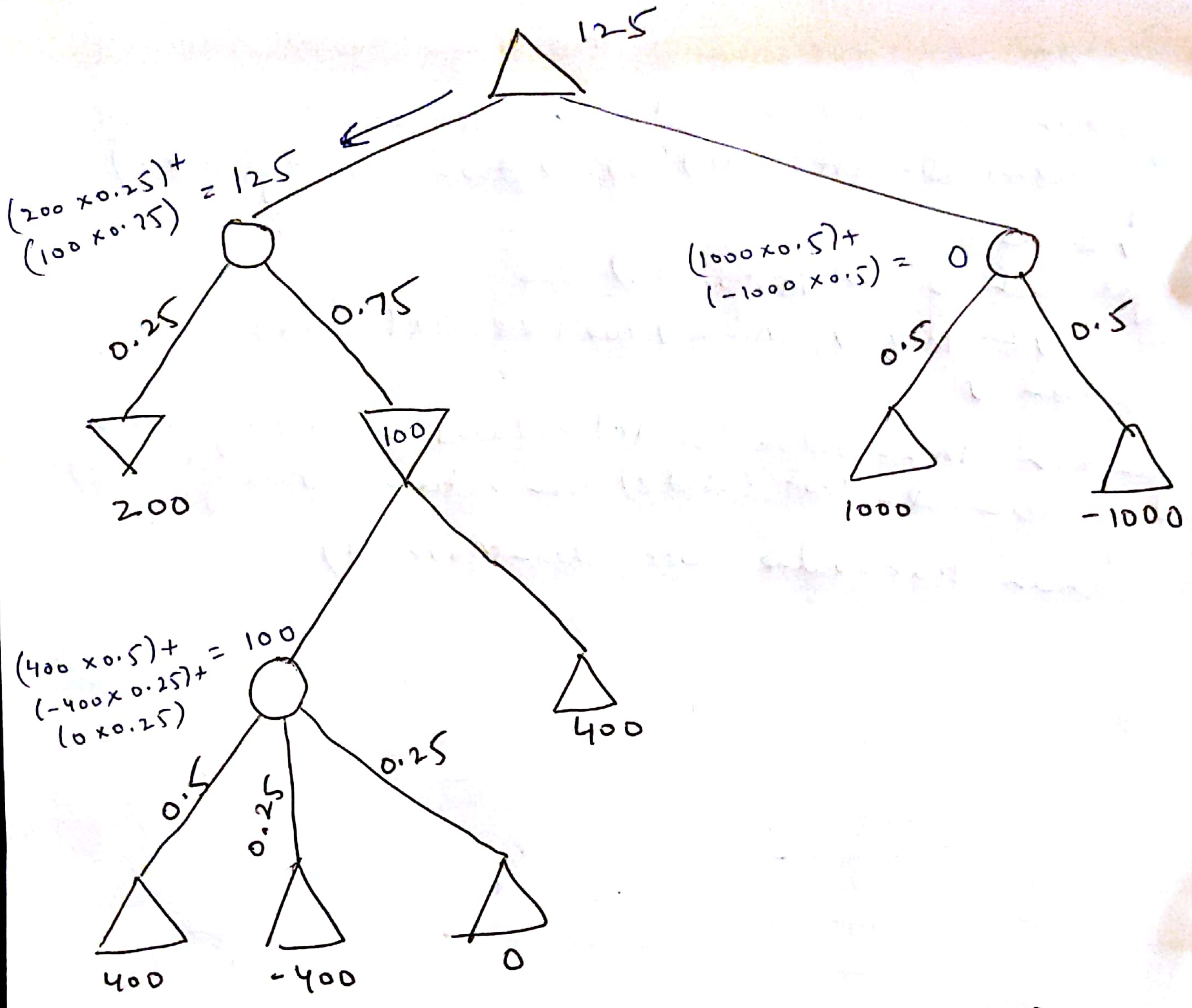


100 is the best case solution because we don't know what algorithm the opponent uses.



60 is the worst case possible, that happens if Min decides to use the optimal algorithm.

Problem 4:


12.5
(200 x 0.25) + (100 x 0.75) = 125
0.25
0.75
200
100
(1000 x 0.5) + (-1000 x 0.5) = 0
0.5
0.5
1000
-1000
(400 x 0.5) + (-400 x 0.25) + (0 x 0.25) = 100
400
0.5
0.25
0.25
400
-400
0

The root node ~~Expect the~~ represent the expectation of the MinMax algorithm for Max player.

If min plays optimally -

Max payoff → 400 Min payoff → -400

If min plays randomly -

Max payoff → 400 Min payoff → -400

Problem-5 :

```
function MinMax-Decision(state) returns an action
  return arg max Actions(s) Min-Value(Result(state, a))
function Max-value (state) returns a utility value
  if Terminal-Test (state) then return Utility(state)
  v ← ∞
  for each 'a' in Actions(S) do
      v ← Max(v, Min-Value (Result(s,a)))
  return v
function Min-value (state) returns a utility value
  if Terminal-Test (state) then return Utility(state)
  return Max-Value (DeepGreenMove(S))
```