1. Given the simple game tree (binary, depth 3) below, label the nodes with up or down arrows, as discussed in the textbook

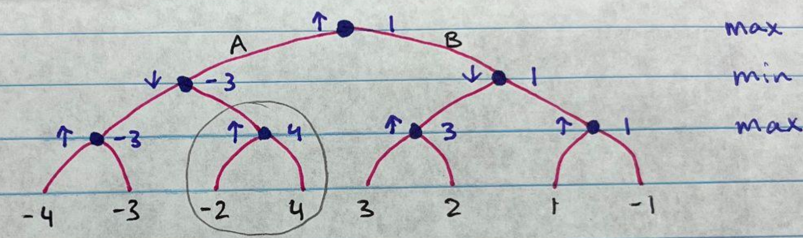
p1 = max, p2 = min

Player 1 should take action B at the root.

Expected outcome is 1.

↓ = min node/ min turn

↑ = max node/ max turn



e) relabeling - if we label as

-4 -3 -2 -1 1 2 3 4

we can prune 3 branches

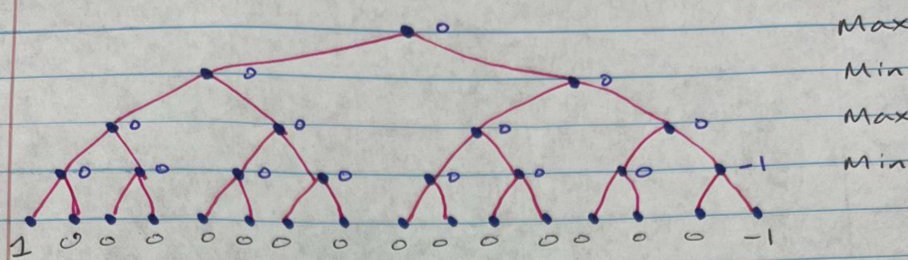
f) relabel to eliminate pruning

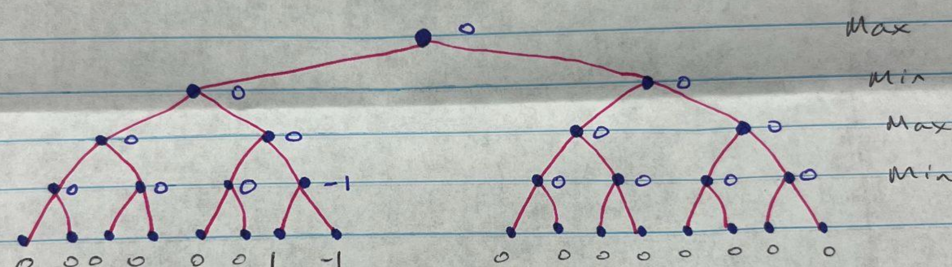
4 3 2 1 -1 -2 -3 -4

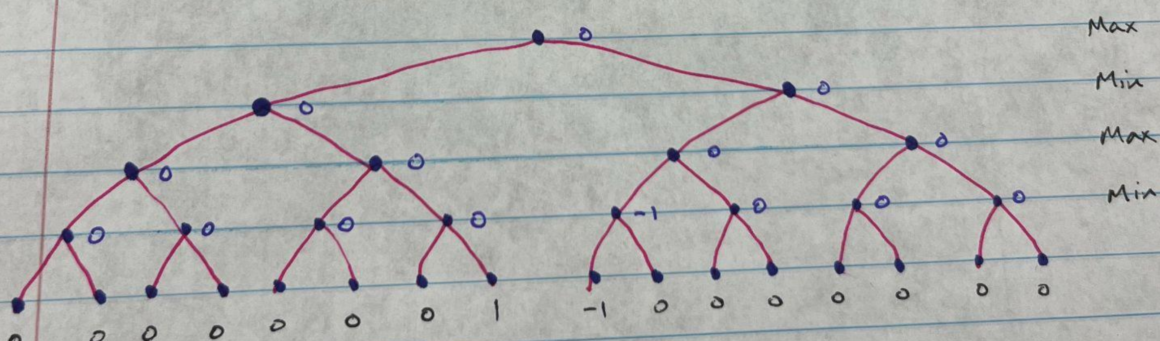
then we have to check all.

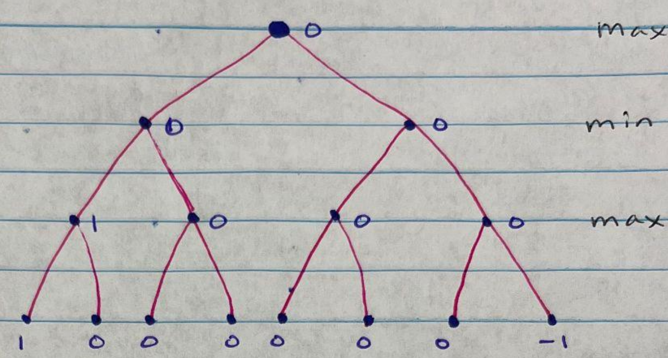
2. In a simple binary game tree of depth 4 (each player gets 2 moves), suppose all the leaves have utility 0 except one winning state (+1000) and one loosing state (-1000).
    a. Could the player at the root force a win?
        i. No. The best they can do is force a draw (0).
        ii. We can see this by drawing the binary game tree with different configurations for the leaf utility values.
    b. Does it matter where the 2 non-zero states are located in the tree? (e.g. adjacent or far apart)
        i. No. Regardless of the location, the result will always be a 0.
        ii. We can see this by drawing the binary game tree with different configurations for the leaf utility values.
        iii. Essentially, Min will always get the last move and be able to move away from the +1000 node. Max will always be able to predict that Min's last move will move to either a 0 or a -1000, and Max will always make the move that results in Min's only choice being to play towards 0.
    c. If this question was changed to have a different depth, would it change the answers to the two questions above? If yes, how do the answers change? If no, explain why no change would happen.
        i. No, answers do not change.
        ii. There is only 1 path that Max can take to force the win. As long as Min has at least 1 turn, Min can use that turn to force the game along any of the other paths that aren't this win-forcing route. Basically, as long as Min can take at least 1 turn, Min can force the game away from the single route that results in Max winning.
        iii. Only if depth is 1 can player 1 force a win.

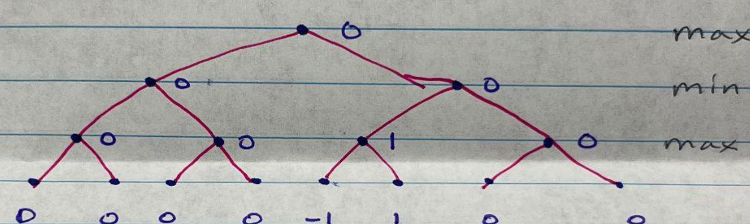
Below is work for #2

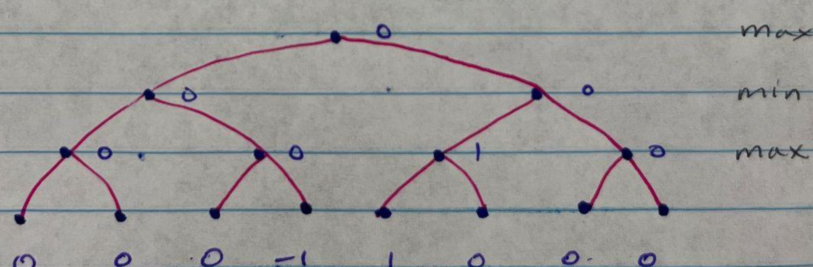

Max
Min
Max
Min
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1

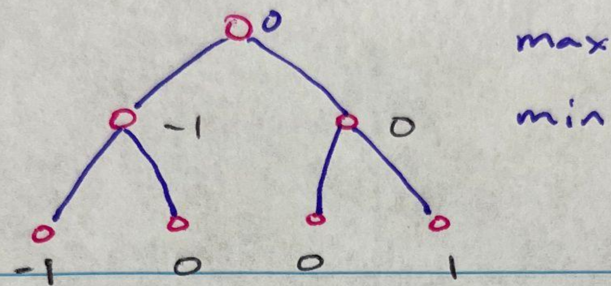

Max
Min
Max
Min
0 0 0 0 0 0 1 -1 0 0 0 0 0 0 0 0

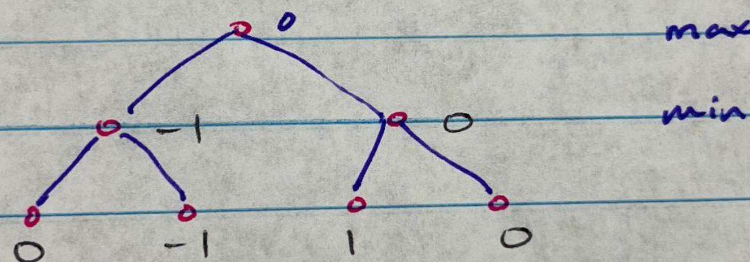

Max
Min
Max
Min
0 0 0 0 0 0 0 1 -1 0 0 0 0 0 0 0

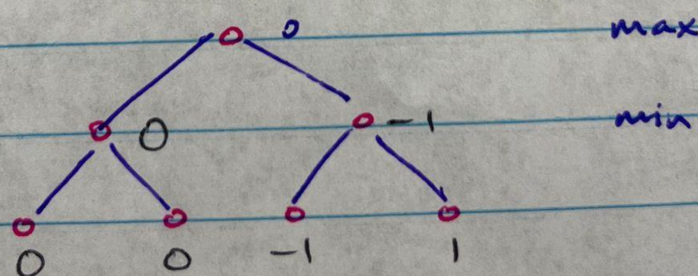

0
max
0 0
min
1 0 0 0
max
1 0 0 0 0 0 0 −1


0
max
0 0
min
0 0 1 0
max
0 0 0 0 −1 1 0 0


0
max
0 0
min
0 0 1 0
max
0 0 0 −1 1 0 0 0


0
max
-1
0
min
-1
0
0
1


0
max
-1
0
min
0
-1
1
0


0
max
0
-1
min
0
0
-1
1

3. Hiking Philosophers. Three philosophers, Alex (A), Bob (B), and Charlie (C), are going on a hike and need to decide the order in which they will hike. Alex and Charlie have PhDs, while Bob has a MS degree. Adjacent hikers in the sequence have to have different degrees. Finally, Charlie does not want to be last.
    a. Show how to set this up as a Constraint Satisfaction Problem. (what needs to be defined?)
    b. Draw the Constraint Graph (label all nodes and edges)
    c. Trace how plain Backtracking (BT) (with no heuristics) would solve this problem, assuming values are processed in alphanumeric order. Identify instances where back-tracking happens.
    d. Trace how BT would solve this problem using the MRV heuristic.

Work/solutions is below

Alex position / Bob position / Charlie position

3. a) must define X : set of variables : $\{A, B, C\}$

D : set of domains : possible positions for each person

C : set of constraints : $\{C_1, C_2, C_3, C_4, C_5, C_6\}$

Domain of A: $\{1, 2, 3\}$

Domain of B: $\{1, 2, 3\}$

Domain of C: $\{1, 2\}$

C_1: $A \neq C+1$ or $C \neq A+1$

C_2: $A \neq C-1$ or $C \neq A-1$

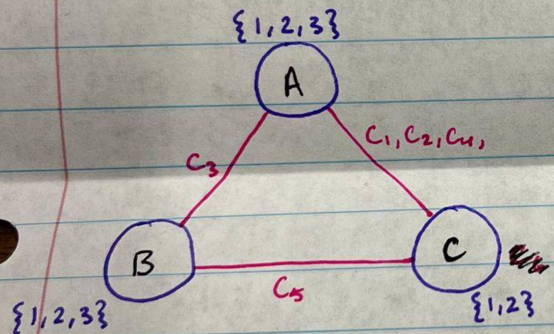
C_3: $A \neq B$ or $B \neq A$

C_4: $A \neq C$ or $C \neq A$

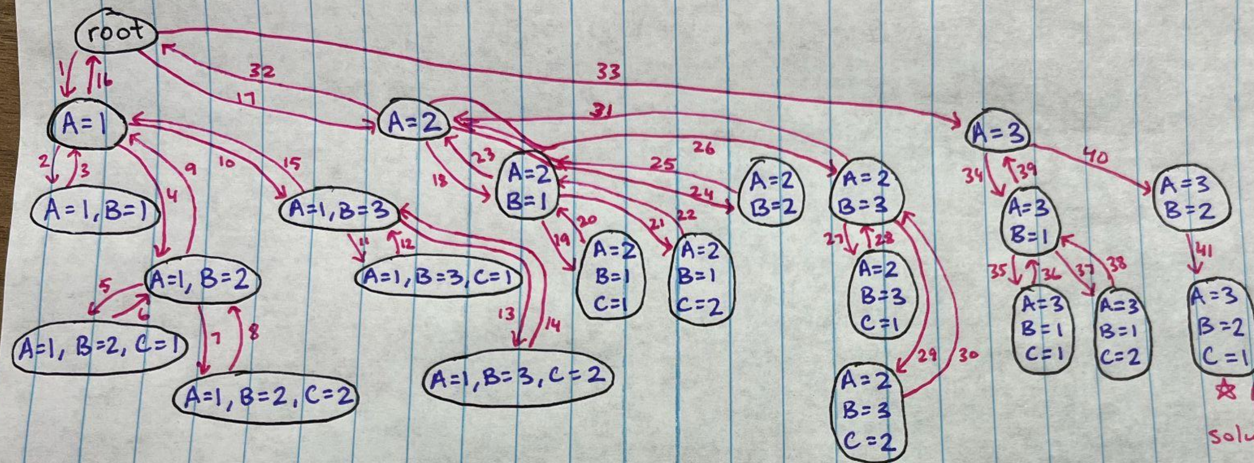
C_5: $B \neq C$ or $C \neq B$

subject to constraints $C_1 - C_5$

b) Constraint Graph



c) BT, no heuristic, values processed in alphanumeric order



☆ All constraints satisfied

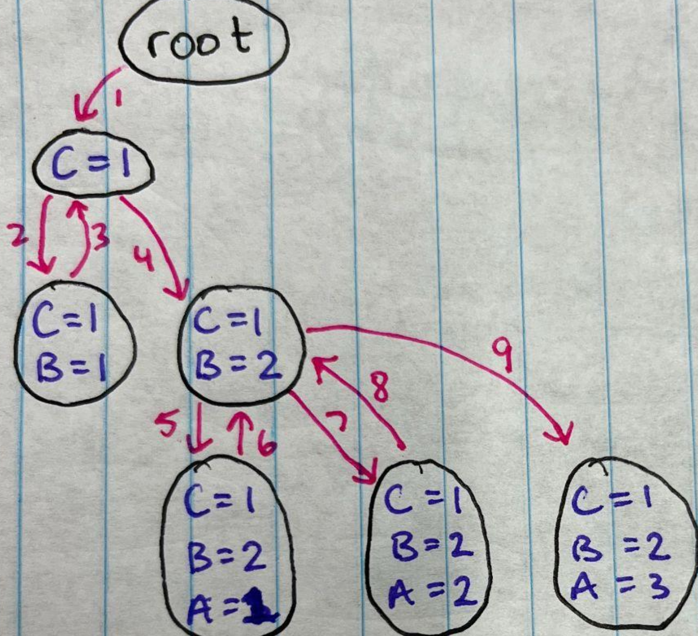
solution is A=3
B=2
C=1

BT occurs on steps: 3, 6, 8, 9, 12, 14, 15, 16, 20, 22, 23, 25, 28, 30, 31, 32, 36, 38, 39

↳ if solution space exhaustively searched w/o solution e.g. 9, 15, 16, 23, 31, 32, 39

↳ if constraint violated (every other BT step) want $C=1, B=2, A=3$

d) BT with MRV Heuristic



BT on steps: 3, 6, 8,

↳ constraint violated.