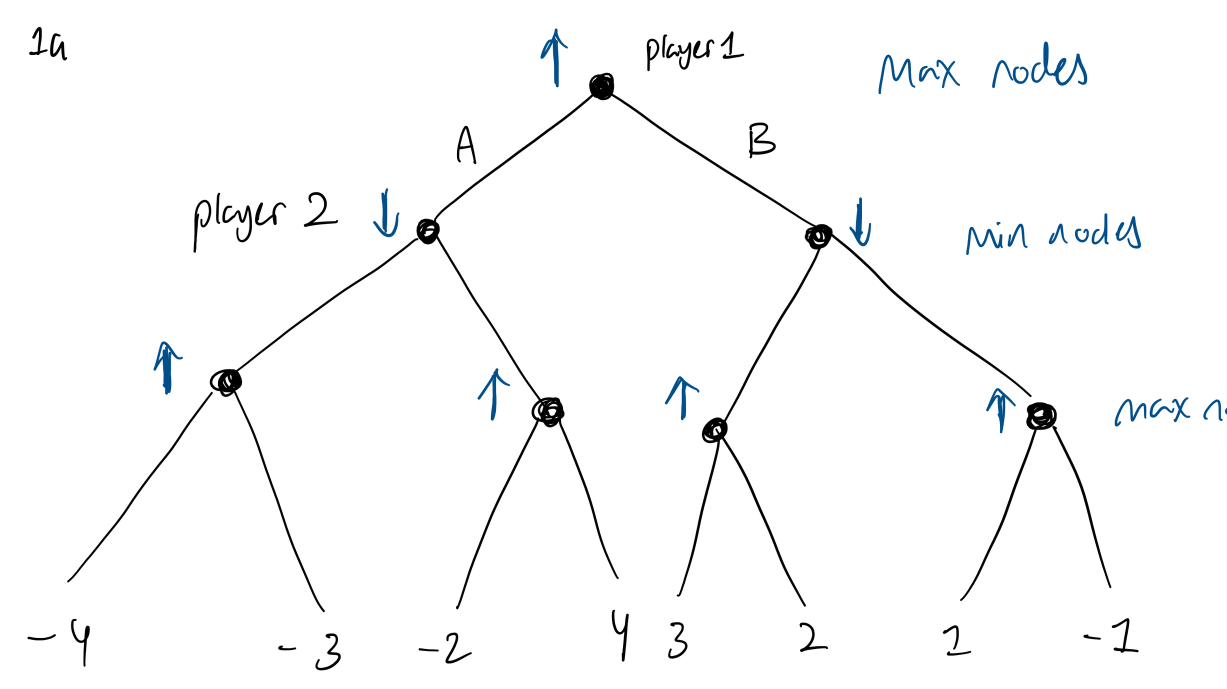
HW2

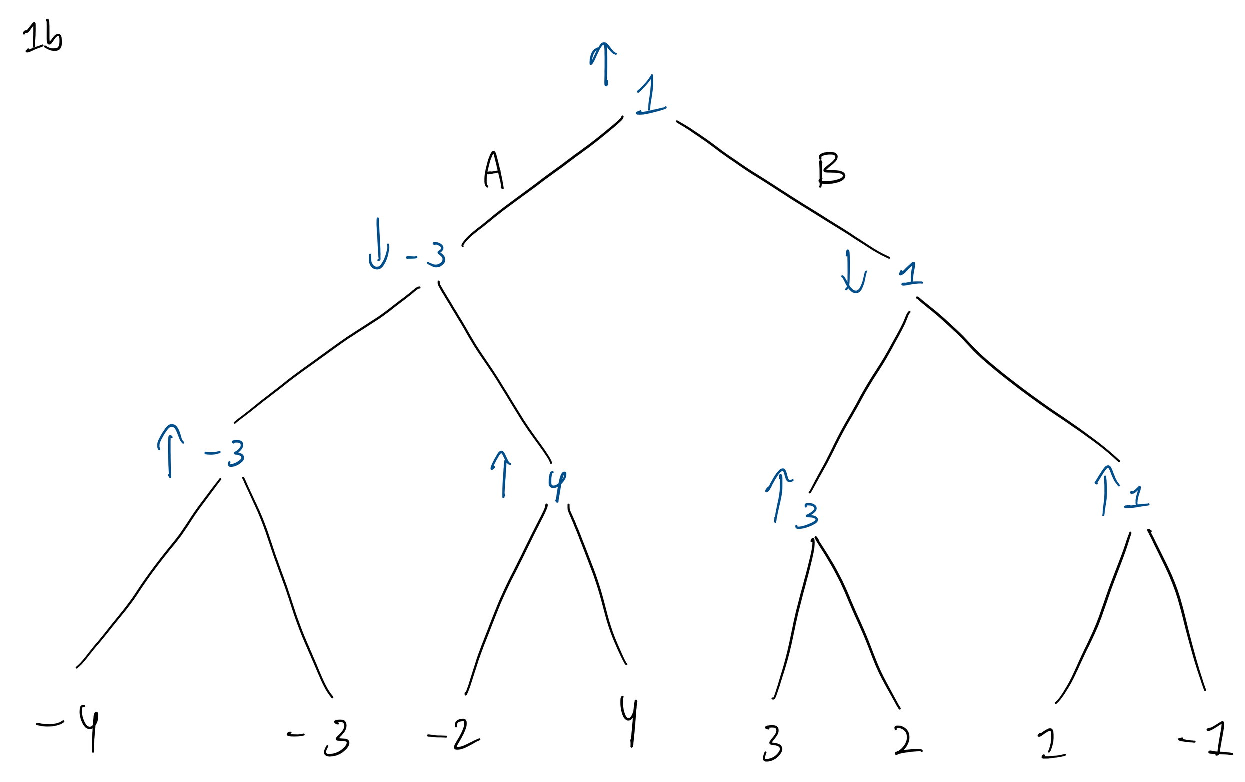
Wednesday, October 1, 2025 3:24 PM

Question 2

1a



1b

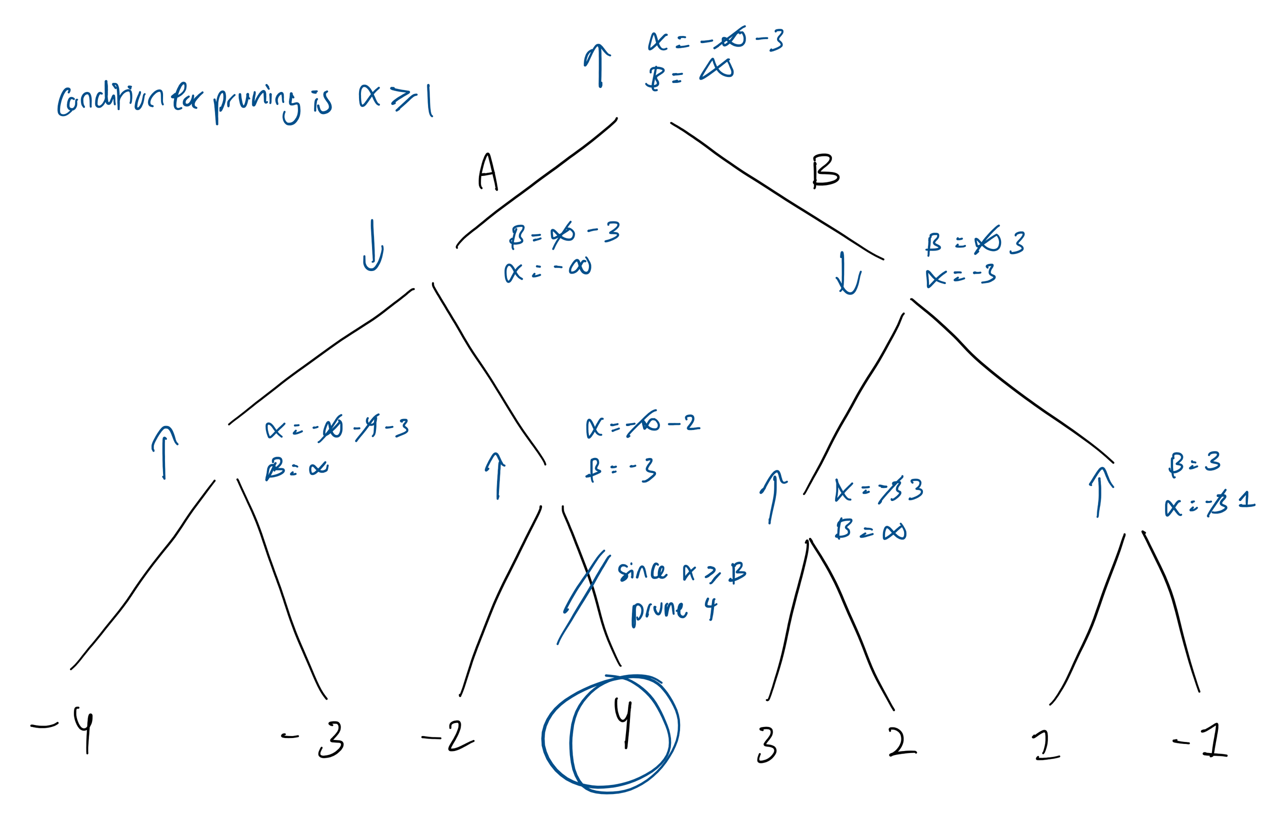


1C) option B is optimal for player 1 as 1 > -3 and root node is a max node so play will choose higher of two values therefore option B is optimal for player 1

1 d) expected outcome is minimax value at root node which is 1

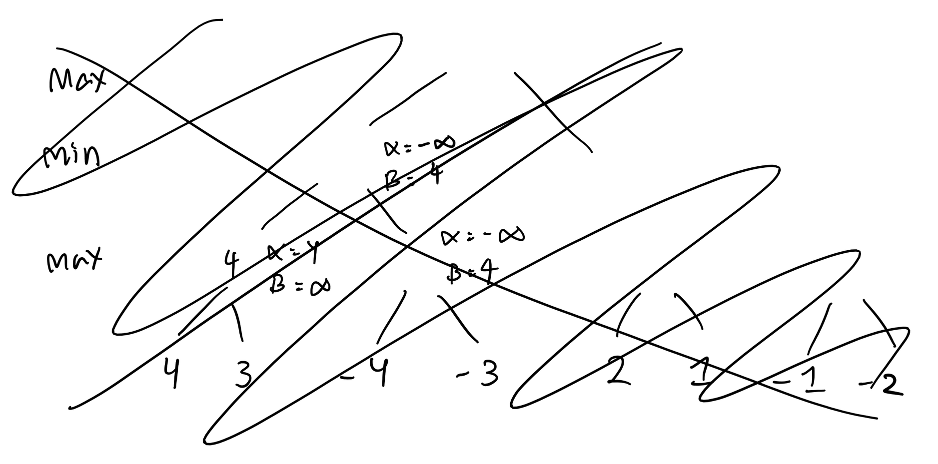
1e) For $\alpha = -\infty$ and $\beta = \infty$

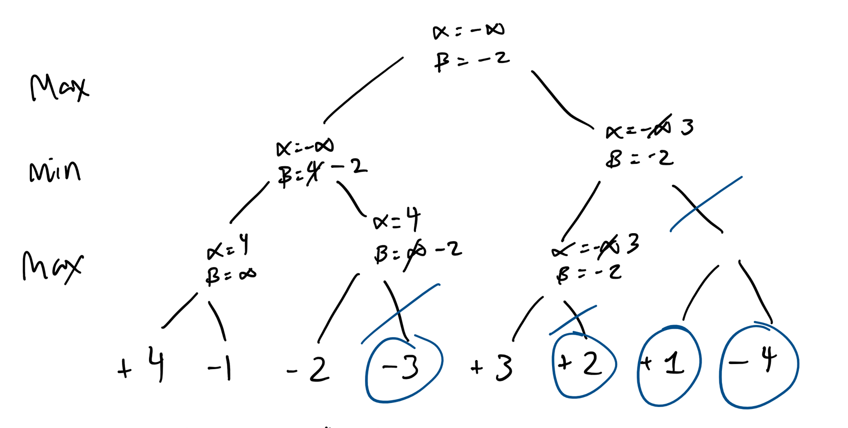
Condition for pruning is $\alpha \geq \beta$



1f) For max nodes put best values early & for min nodes put worst values early

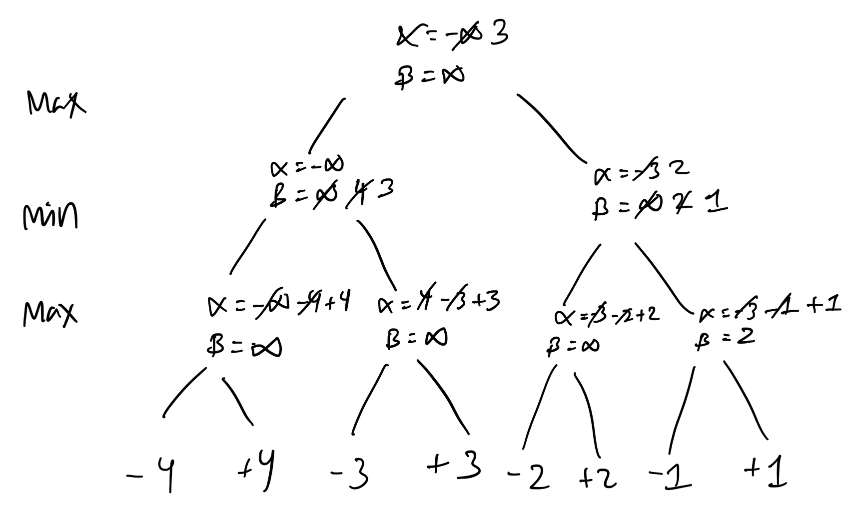
For max pruning try and have max values on left n min on right best case is branch A right n B right can be pruned





this prunes 4 values

1g)



2) a)

Variables

P1 → person in position 1

P2 → person in position 2

P3 → person in position 3

Domain

each var can take any 3 hikers {A, B, C}

but no hiker can be repeated

Constraints

1) each person needs appear at least once and $P1 \neq P2 \neq P3$

2) Adjacent degrees must differ

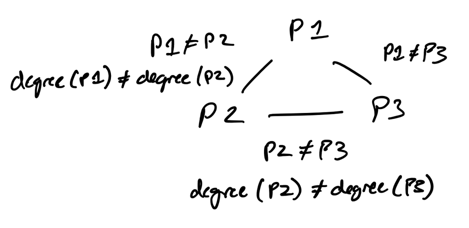
degree (P1) ≠ degree (P2)

degree (P2) ≠ degree (P3)

3) Carol ≠ P3

b) Nodes = variables

Edges = constraints



c) Try P1 = A ✓

→ Try P2 = A × same as P1

Try P2 = B → degree constraint met ✓

→ Try P3 = A × same as P1

→ Try P3 = B × same as P2

→ Try P3 = C × Carol is last violates constraint

→ Try P2 = C × violates degree constraint

→ Arrows indicate backtracking

→ Try P1 = B ✓

Try P2 = A ✓

→ Try P3 = C × Carol last violates constraint

→ Try P2 = B × ~~violates~~ P1 ≠ P2

Try P2 = C ✓

→ Try P3 = A × degree (P2) ≠ degree (P3)

→ Try P1 = C ✓

→ Try P2 = A × degree (P1) ≠ degree (P2)

Try P2 = B ✓

Try P3 = A ✓

d) MRV = Min remaining value

A (cannot touch other PHD)

B

C (cannot touch other PHD) (cannot be last)

P1 P2 P3

A cannot go P2

B can go anywhere

C cannot go P2 or P3

So start assigning one with most constraints

Try P1 = C

remaining domains

P2 ∈ {A, B}

P3 ∈ {A, B}

Try P2 = A (A has less options left) × degree constraint

back → Try P2 = B ✓

Try P3 = A