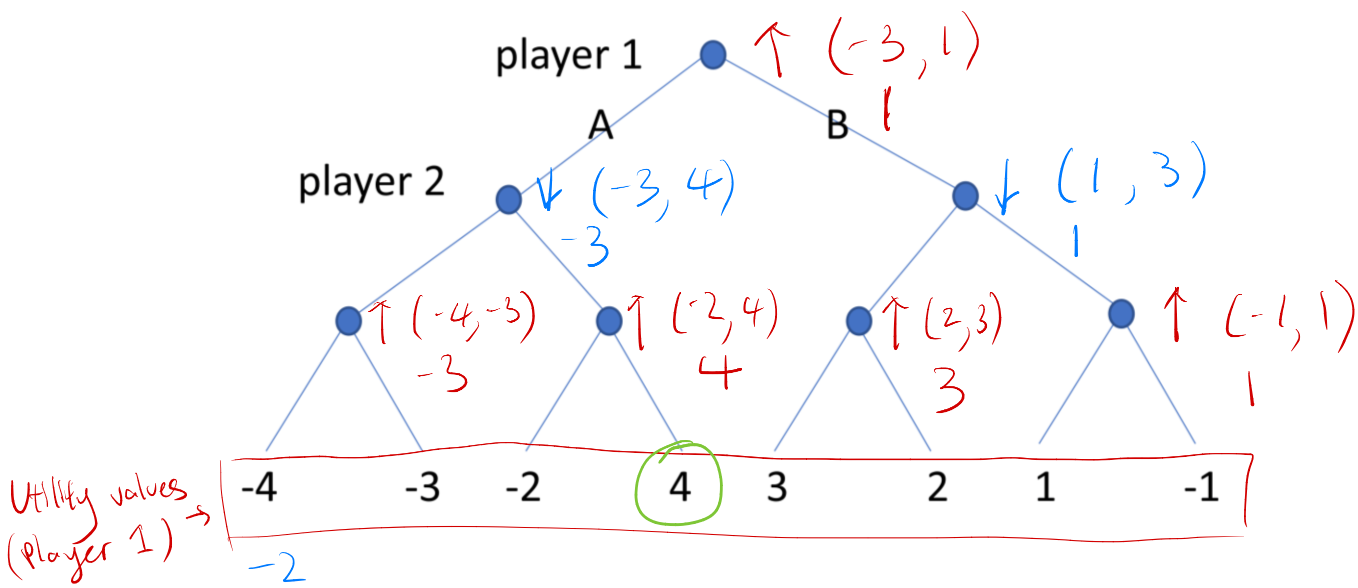
CSCE 420 - Fall 2023
Homework 1 (HW1)
due: Thurs, Oct 5, 11:59pm - Late written homeworks will not receive credit.

Turn-in answers as a Word document (HW1.docx or .pdf) and commit/push it to your class github repo.

When pushing the final version of your HW1, also run the command:
git tag "HW1" && git push origin "HW1"
This will be used to record time of submission for late penalty when applicable.

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



Label the leaves with utility values for player 1 (who is at the root).

Compute the *minimax* values at the internal nodes (write the values next each node).

Should the player 1 take action A or B at the root? B, because it has higher utility.

What is the expected outcome (payoff at the end of the game)?

Payoff is 1 for player 1

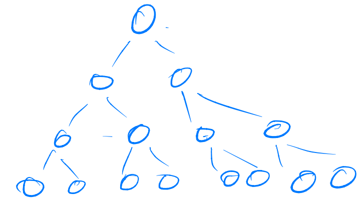
Which branches would be pruned by alpha-beta pruning? (circle them)

How could the leaves be relabeled to maximize the number of nodes pruned? (you can move the utilities around arbitrarily to other leaves, but you still have to use -4,-3,-2,-1,+1,+2,+3,+4)

A: 2, 3, 4, 1 B: -1, -2, -3, -4

How could the leaves be relabeled to eliminate pruning?

A: -2, 4, -2, -3 B: 3, 2, 1, -1



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).

- Could the player at the root force a win?

  No, the player two can choose diff path

- Does it matter where the 2 non-zero states are located in the tree? (e.g. adjacent or far apart)

  No, the final choice is made by player 1 so its always maximized.

- 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.

  No, root can never force a win and it doesnt matter where the nodes are. The end will always be 0.

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.

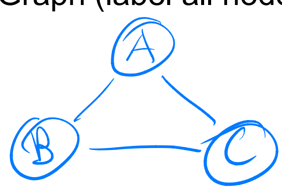
a) Show how to set this up as a Constraint Satisfaction Problem. (what needs to be defined?)

   Variables: Alex A bob B Charlie C

   Domains: [1, 2, 3]

   Constraint: 1≠2 , 2≠3 , C≠3

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.

	A	B	C
1	1		
2	1	2	
3			1
4		2	1

Backtrack

	A	B	C
5	3	2	1

d) Trace how BT would solve this problem using the MRV heuristic.

Start with C since it has least options

	A	B	C
1			1
2	2		1
3		2	1
4	3	2	1
5			

Backtrack