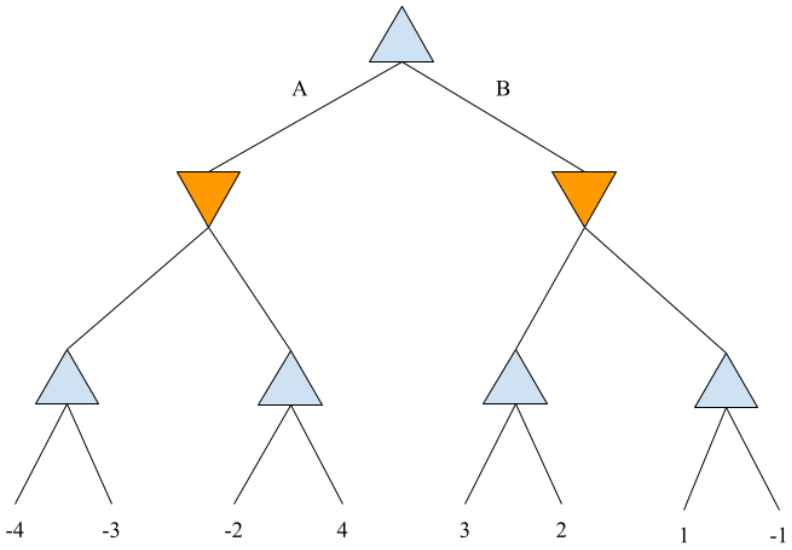
Octavio Almanza
UIN: 628005492

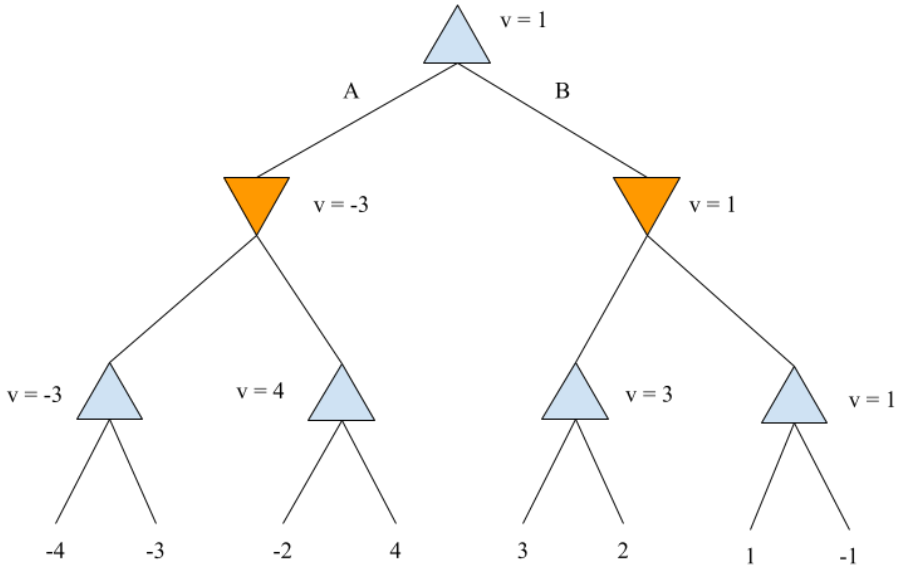
CSCE 420 Section 500

Homework 1 (HW1)

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



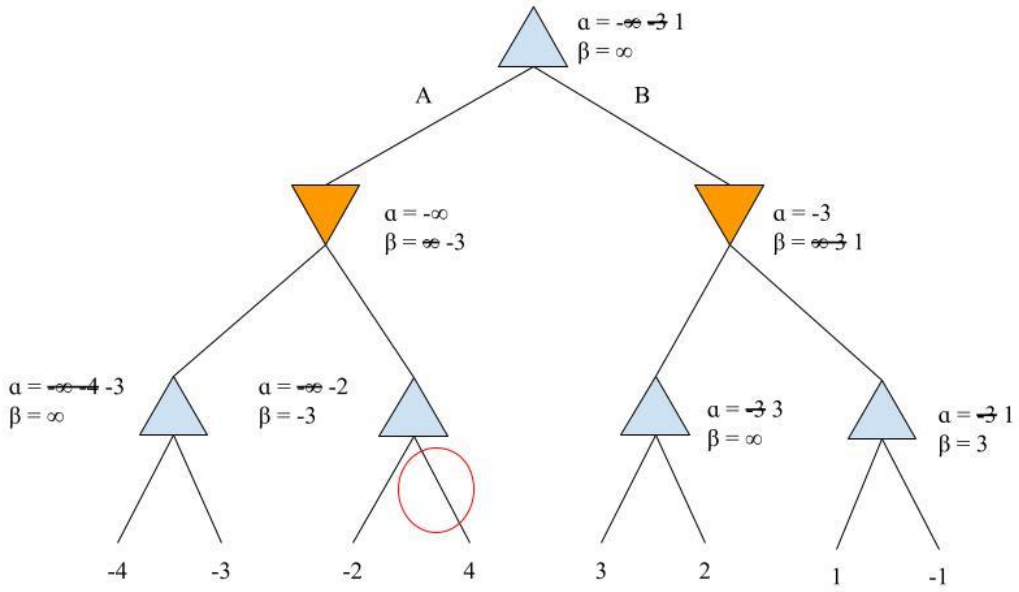
- The leaves are labeled with utility values for player 1 (who is at the root).
- Compute the *minimax* values at the internal nodes (write the values next each node).



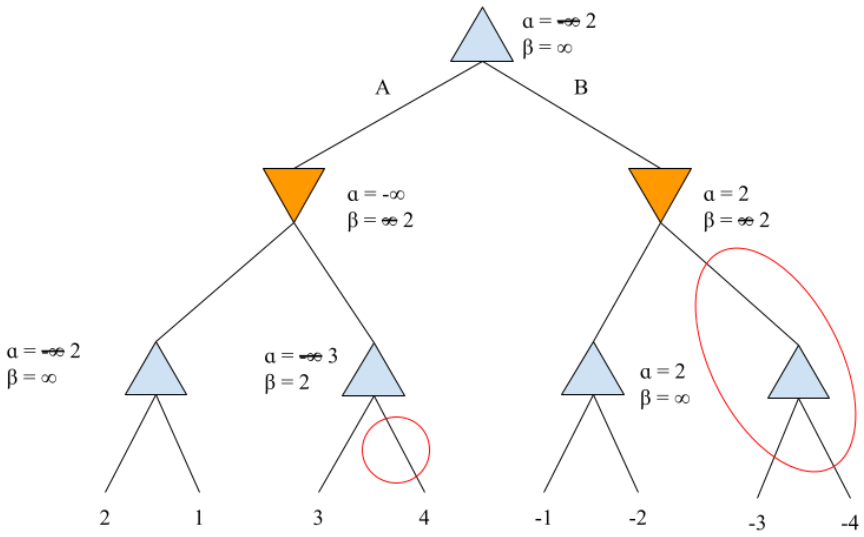
- Should the player 1 take action A or B at the root?
  Player 1 should take action B at the root because it will lead to a positive payoff. If he were to take action A, he would end up with a payoff of -3.

- What is the expected outcome (payoff at the end of the game)?
  The expected outcome, assuming both players play optimally, is +1.

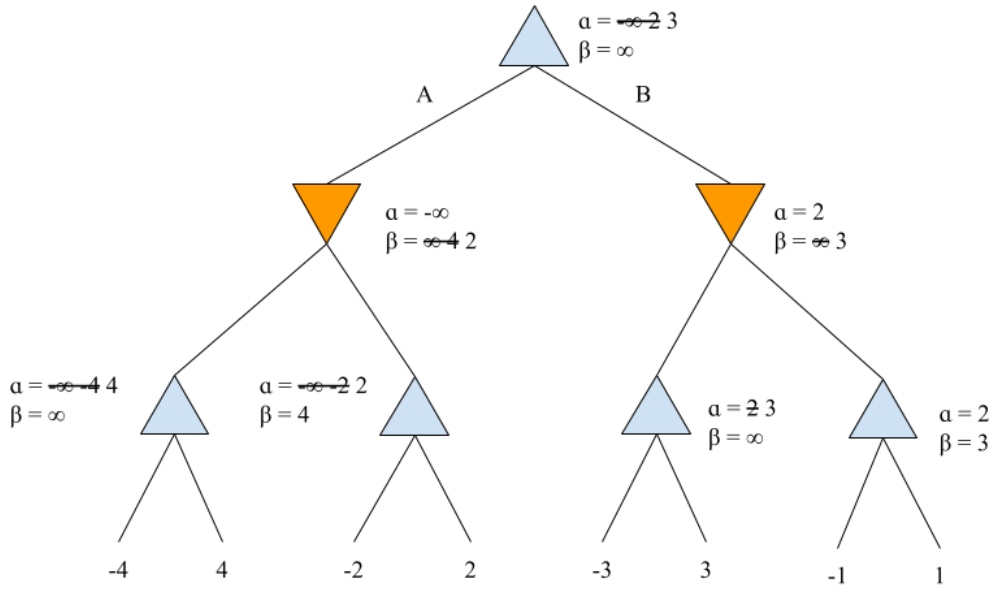
- Which branches would be pruned by alpha-beta pruning? (circle them)



- 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).
  We can relabel the leaves to the following to maximize the number of nodes pruned:



- How could the leaves be relabeled to eliminate pruning?
  We can relabel the leaves to the following to eliminate pruning completely:



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 losing state (-1000).

- **Note:** For the following questions, I will be using the following assumptions:
  1. Player 1 (the root player) and Player 2 (the non-root player) are both playing optimally
  2. The two players take turns making a move.

- Could the player at the root force a win?
  No. Since each player gets 2 moves, and the players take turns making moves, and the root player goes first, the non-root player will always get the last move. Naively speaking, there are four possible cases for the last move:
  1. Both leaf nodes have a utility value of 0
  2. One leaf node has a utility value of 0 and the other is -1000
  3. One leaf node has a utility value of 0 and the other is +1000
  4. One leaf node has a utility value of -1000 and the other is +1000

  Assuming that the non-root player makes an optimal final move, there is no way for the root player to win.

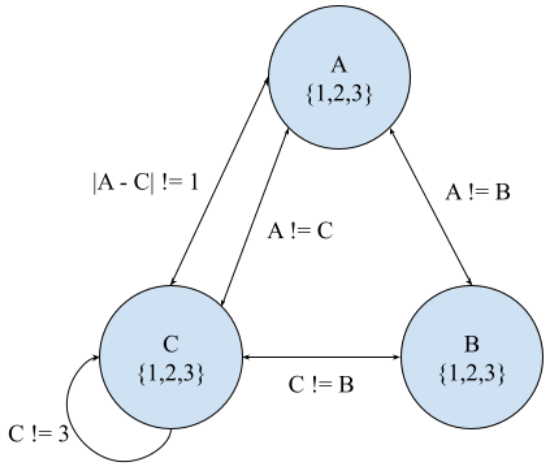
- Does it matter where the 2 non-zero states are located in the tree? (e.g. adjacent or far apart)
  No. As mentioned in my previous answer, assuming the non-root player makes an optimal final move, there is no way for the root player to win the game.

- 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 not, explain why no change would happen.
  No, changing the depth will not allow the root player to win. This is because there will always be a min node in the tree that discards the branch leading to the +1000 leaf node and will instead select the branch that leads to a lower utility value.

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?)
   We must define variables, their respective domain, and the constraints.
   Variables: A, B, C
   Domains: $D_A = D_B = D_C = \{1, 2, 3\}$
   Constraints: $\{A \in D_A, B \in D_B, C \in D_C, A != B, A != C, B != C, C != 3, |A - C| != 1$
   - (Since A and C are hikers with the same degrees, the absolute value of the difference of their position cannot be 1, since that would mean they are next to each other)

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.

Step	A	B	C	Explanation
1	1			Assign 1 to A since we select the lowest position value
2	1	2		Assign 2 to B because A != B and we select the lowest consistent position value.
3	1	2		**Backtrack.** Charlie can only select position 3, but we have the constraint that Charlie cannot be the last hiker (the hiker at position 3).
4	1	3		Change previous choice B->3.
5	1	3		**Backtrack**. Charlie can only get spot 2, meaning Alex and Charlie would be consecutive hikers that have the same degree (|1 - 2| = 1)
6	2			Change previous choice A->2
7	2	1		Since A != B, B can select position 1 or 3. Select the lowest consistent position value.
8	2	1		**Backtrack**. Charlie can only select 3, but we have the constraint that Charlie cannot be the last hiker. It would also violate another constraint since Alex and Charlie would be consecutive hikers with the same degree (|2 - 3| = 1).
9	2	3		Change previous choice B->3
10	2	3		**Backtrack**. Charlie can only pick spot 1, but this would cause Charlie and Alex to be consecutive campers with the same degree (|2 - 1| = 1).
11	3			Change previous choice A->3
12	3	1		B can select from 1 or 2. Select 1 since it is the lowest consistent position value
13	3	1		**Backtrack**. Charlie can only pick spot 2, but this would cause Charlie and Alex to be consecutive campers with the same degree (|3 - 2| = 1).
14	3	2		Change previous choice B->2
15	3	2	1	Charlie can only pick spot 1. This does not contradict any constraints, so we are done.

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

Step	A	B	C	Explanation
1			1	The MRV heuristic states that we should select our variable based on whichever has the minimum remaining values. C only has two remaining values since C != 3, so we assign it a value first.
2	3		1	A has alphanumeric precedence, we assign it a value. If we selected 2, then that would violate the constraint that consecutive hikers cannot have the same degree. Therefore, we select 3.
3	3	2	1	Bob can only pick spot 2. This does not contradict any constraints, so we are done.