CS 188
Spring 2023

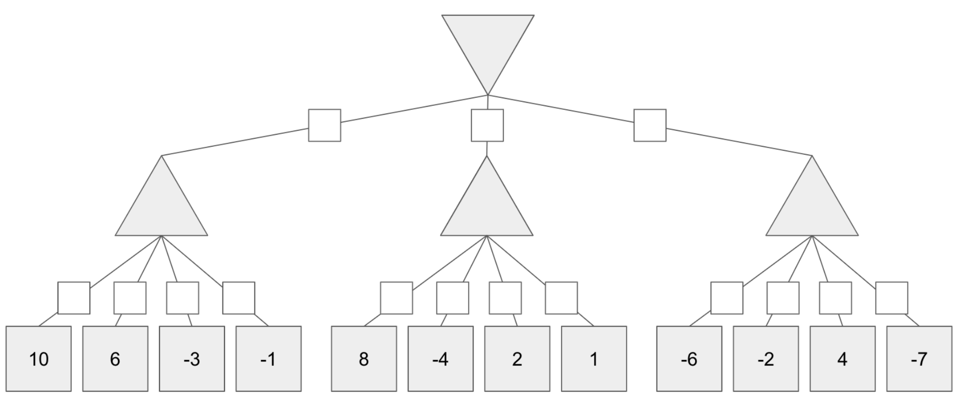
Final Review: Games Solutions

Q1. Coin Stars

In a new online game called Coin Stars, all players are walking around an M x N grid to collect **hidden coins, which only appear when you're on top of them**. There are also power pellets scattered across the board, which are visible to all players. If you walk onto a square with a power pellet, your power level goes up by 1, and the power pellet disappears. Players will also attack each other if one player enters a square occupied by another player. In an attack, the player with a higher power level will steal all the coins from the other player. If they have equal power levels, nothing happens. Each turn, players go in order to move in one of the following directions: {N, S, E, W}.

In this problem, you and your friend Amy are playing Coin Stars against each other. You are player 1, and your opponent Amy is player 2. Our state space representation includes the locations of the power pellets (x_{p_j}, y_{p_j}) and the following player information: (1) Each player's location (x_i, y_i); (2) Each player's power level l_i; (3) Each player's coin count c_i.

(a) Suppose a player wins by collecting more coins at the end of a number of rounds, so we can formulate this as a minimax problem with the value of the node being $c_1 - c_2$. Consider the following game tree where you are the maximizing player (maximizing the your net advantage, as seen above) and the opponent is the minimizer. Assuming both players act optimally, if a branch can be pruned, fill in its square completely, otherwise leave the square unmarked.



● None of the above can be pruned

If you traverse the tree with $\alpha - \beta$ pruning, you will see that no branches can be pruned

(b) Suppose that instead of the player with more coins winning, every player receives payout equal to the number of coins they've collected. Can we still use a multi-layer minimax tree (like the one above) to find the optimal action?

○ Yes, because the update in payout policy does not affect the minimax structure of the game.

○ Yes, but not for the reason above

○ No, because we can no longer model the game under the updated payout policy with a game tree.

● No, but not for the reason above

No, because the game is no longer zero-sum: your opponent obtaining more coins does not necessarily make you worse off, and vice versa. We can still model this game with a game-tree, where each node contains a tuple of two values, instead of a single value. But this means the tree is no longer a minimax tree.

An example of using the minimax tree but not optimizing the number of coins collected: when given a choice between gathering 3 coins or stealing 2 coins from the opponent, the minimax solution with $c_1 - c_2$ will steal the 2 coins (net gain of 4), even though this will cause it to end up with fewer coins.

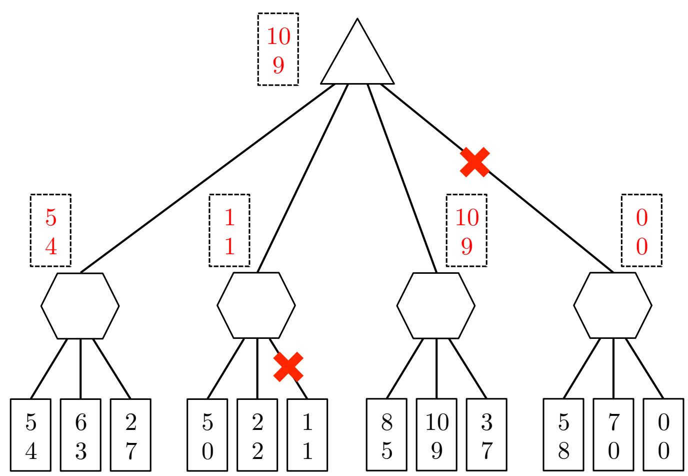
Q2. Game Trees and Pruning

You and one of the 188 robots are playing a game where you both have your own score. In the leaf nodes of the game tree, your score is on top and the robot's score on the bottom.

- The maximum possible score for either player is 10.
- You are trying to maximize your score, and you do not care what score the robot gets.
- The robot is trying to minimize the absolute difference between the two scores. In the case of a tie, the robot prefers a lower score. For example, the robot prefers (5,3) to (6,3); it prefers (5,3) to (0,3); and it prefers (3,3) to (5,5).

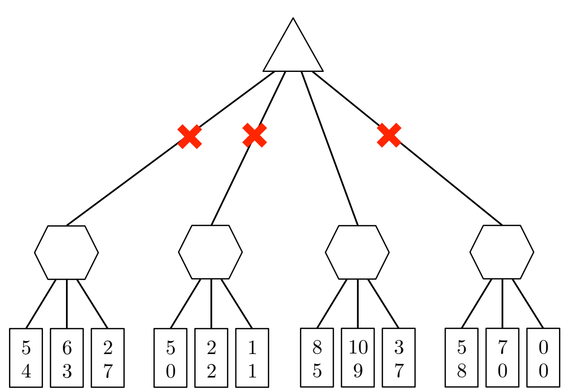
The figure below shows the game tree of your max node followed by the robots nodes for your four different actions.

(a) Fill in the dashed rectangles with the pair of scores preferred by each node of the game tree.



(b) You can save computation time by using pruning in your game tree search. On the game tree above, put an 'X' on line of branches that do not need to be explored. Assume that branches are explored from left to right.

(c) You now have access to an oracle that tells you the order of branches to explore that maximizes pruning. On the copy of the game tree below, put an 'X' on line of branches that do not need to be explored given this new information from the oracle.

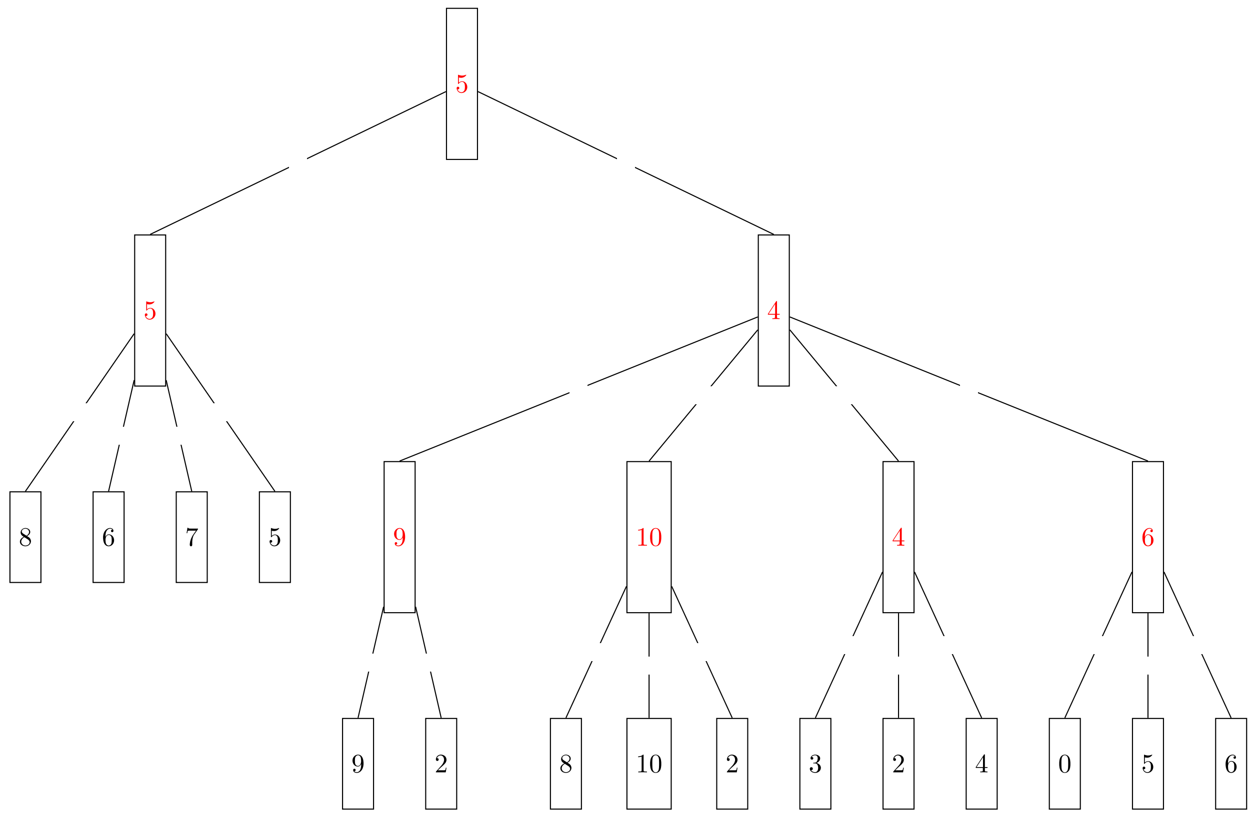


Q3. Game Trees

The following problems are to test your knowledge of Game Trees.

(a) Minimax

The first part is based upon the following tree. Upward triangle nodes are maximizer nodes and downward are minimizers. (small squares on edges will be used to mark pruned nodes in part (ii))



(i) Complete the game tree shown above by filling in values on the maximizer and minimizer nodes.

(ii) Indicate which nodes can be pruned by marking the edge above each node that can be pruned (you do not need to mark any edges below pruned nodes). In the case of ties, please prune any nodes that could not affect the root node's value. Fill in the bubble below if no nodes can be pruned.

◯ No nodes can be pruned

Edges that can be pruned: (parent-child) 10-2, 4-6, 6-0, 6-5, 6-6. So mark: 10-2, 4-6

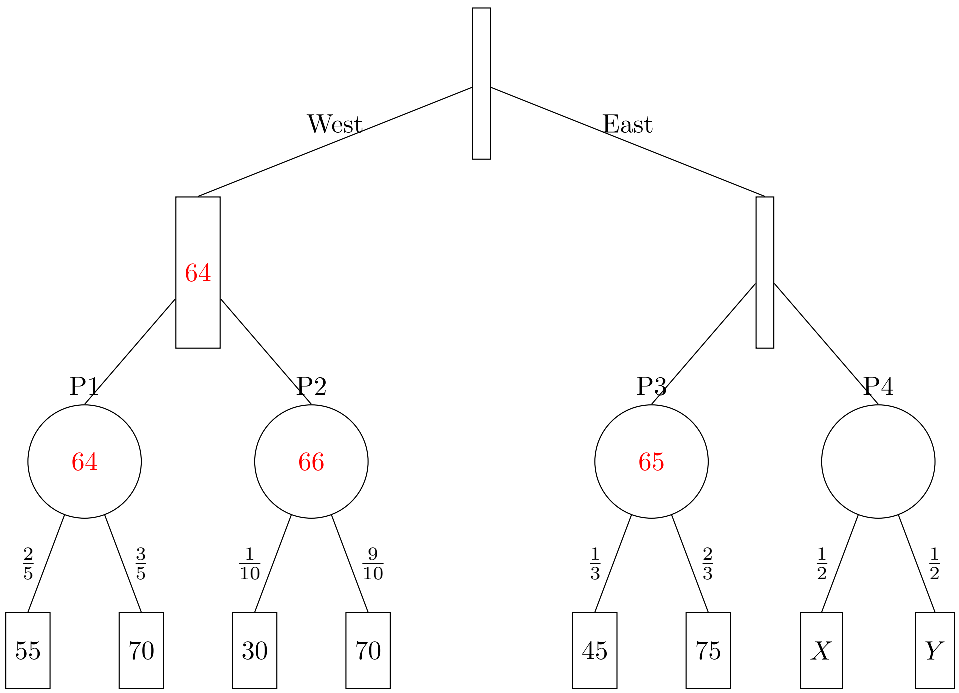
(b) Food Dimensions

The following questions are completely unrelated to the above parts.

Pacman is playing a tricky game. There are 4 portals to food dimensions. But, these portals are guarded by a ghost. Furthermore, neither Pacman nor the ghost know for sure how many pellets are behind each portal, though they know what options and probabilities there are for all but the last portal.

Pacman moves first, either moving West or East. After which, the ghost can block **1** of the portals available.

You have the following gametree. The maximizer node is Pacman. The minimizer nodes are ghosts and the portals are chance nodes with the probabilities indicated on the edges to the food. In the event of a tie, the left action is taken. Assume Pacman and the ghosts play optimally.



(i) Fill in values for the nodes that do not depend on X and Y.

(ii) What conditions must X and Y satisfy for Pacman to move East? What about to definitely reach the P4? Keep in mind that X and Y denote numbers of food pellets and must be **whole numbers**: $X, Y \in \{0, 1, 2, 3, \dots\}$.

To move East: $\boxed{X + Y > 128}$

To reach P4: $\boxed{X + Y = 129}$

The first thing to note is that, to pick A over B, $value(A) > value(B)$.
Also, the expected value of the parent node of X and Y is $\frac{X+Y}{2}$.

$\implies min(65, \frac{X+Y}{2}) > 64$

$\implies \frac{X+Y}{2} > 64$

So, $X + Y > 128 \implies value(A) > value(B)$

To ensure reaching X or Y, apart from the above, we also have $\frac{X+Y}{2} < 65$

$\implies 128 < X + Y < 130$

So, $X, Y \in \mathbb{N} \implies X + Y = 129$