

AI HW4: Analysis of the Game of Nim using Minimax and Alpha-Beta Pruning

Daniela Cruz Perez

11/6/2025

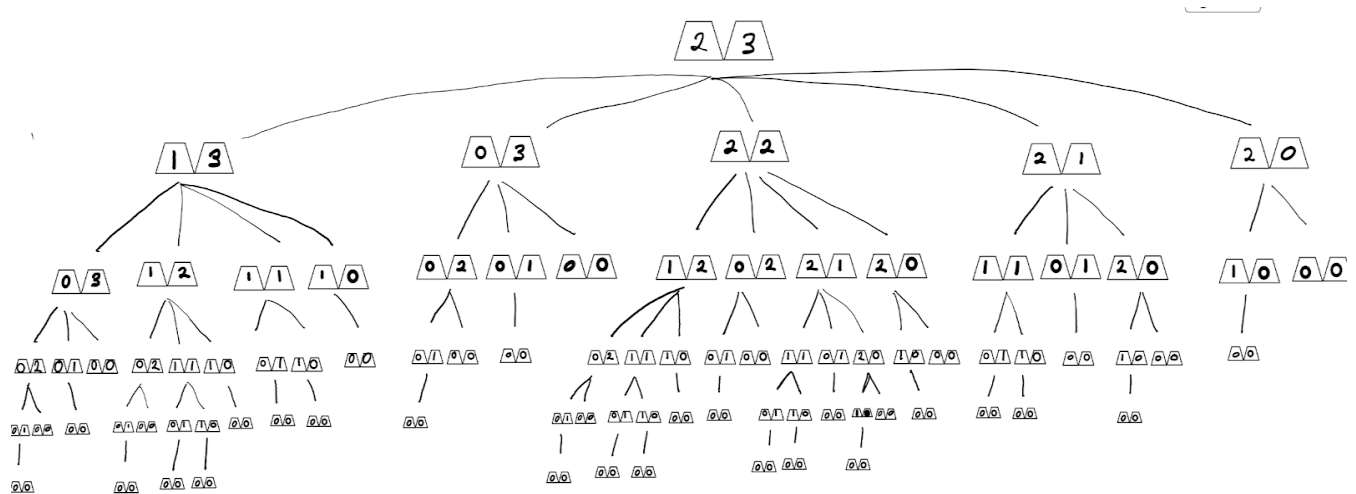
Problem 1

a.) Game Description:

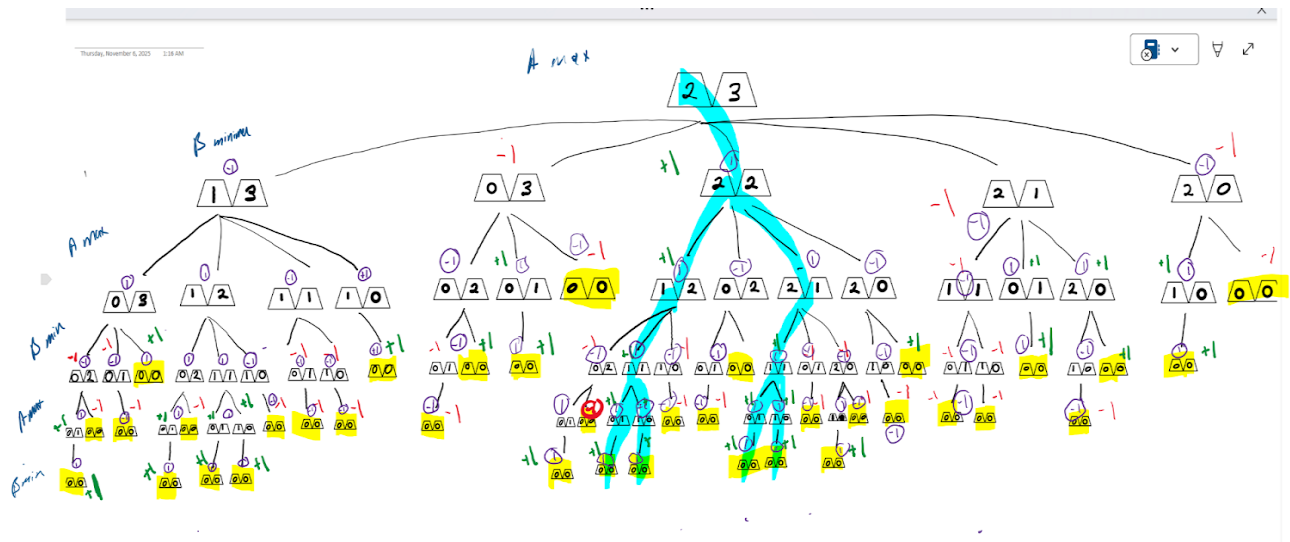
The game of Nim consists of a fixed number of heaps, each containing a variable number of objects, such as a stone or coin. Each player chooses a heap to take from on their turn and can remove any amount of objects from their chosen heap, however they must at least take one. This continues until all objects are removed from the heaps and the heaps are emptied. The player that removed the last object remaining, wins. For instance in a 2-ply game of Nim, where there are 2 heaps and 2 and 3 objects respectively, this is the starting state. Each player will remove any amount of objects from their chosen heap in their turn. In the end, the heaps will be emptied.

b.) Game Tree:

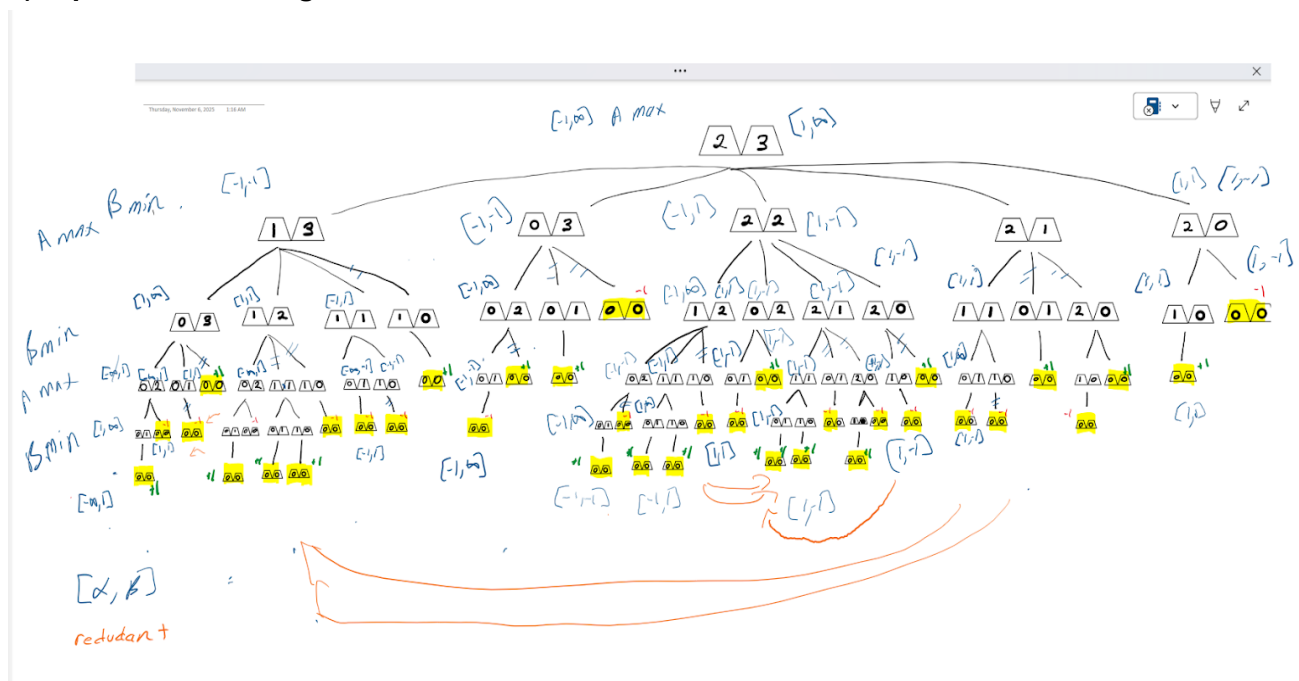
Initial State: 2 heaps, 2 and 3 objects respectively



c.) Minimax Algorithm:



d.) Alpha-Beta Pruning:



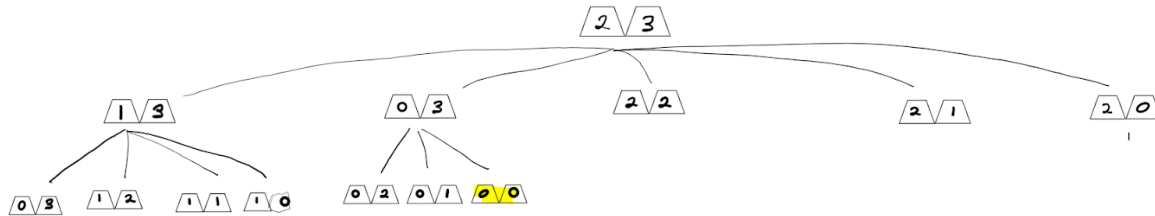
e.) Analysis

The minimax algorithm was effective in finding the optimal path that guarantees a win for player 1. This algorithm iterates through each node, thus is less effective in time. Whereas the alpha-beta pruning algorithm compares the alpha and beta values and decides whether a node can be pruned, thus minimizing time. Especially, in a large tree, alpha-beta pruning is optimal

Problem 2:

a.) Optimization Techniques:

The following map depicts the iterative deep learning algorithm. The algorithm searches through a 2-ply game to find the goal state. At depth 2, the goal state is reached for Player 2. In a two ply-game the algorithm will miss other goal-state nodes deeper into the tree. In a game of nim, the algorithm will choose the closest goal state effectively, and guarantees a win for a player.



Heuristic Evaluation:

The heuristic function used is the number of direct winning outcomes for each node. If the number is equal to -1, the algorithm will not explore the node, and if it equals 1 or 0 (no direct node), the algorithm will continue to explore. The algorithm then applied the minimax function to nodes that can be explored. The tree was able to find the optimal path for player 1 as it eliminated states that resulted in a win for player 2.

