Homework 3 – Artificial Intelligence

Teacher: Stefán Ólafsson

January 30, 2023

Time Estimate

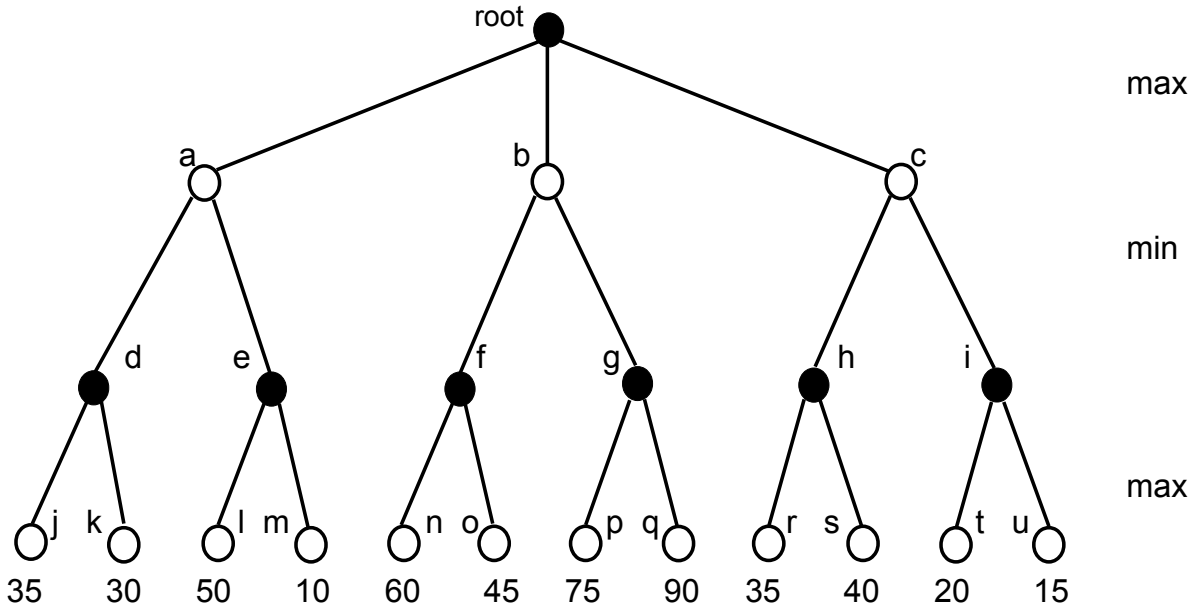
3 hours including reading the relevant parts of chapter 6 on the Minimax and $\alpha - \beta$ algorithms, assuming you have been to the lectures.

Instructions

Hand in a PDF file with the answers to all questions on Canvas.

Questions

1. (60 points) Consider the two-player, turn-taking, zero-sum game with the following game tree:



Apply the $\alpha - \beta$ algorithm (with left-to-right expansion order) to compute the value of the root node! Mark which of the branches can be pruned and why.

1.1 (5 points) What is the best move of the max player in the initial state?

1.2 (5 points) What is the value of the root node?

1.3 (30 points) Which branches of the tree can be pruned and why? Name the branches to be cut and which bounds allow you to cut them (e.g., the branch to x can be cut because $\alpha(y) = 43$ is greater than $\beta(z) = 17$).

1.4 (10 points) Suppose you were using a heuristics to decide on the expansion order of the states. What would be the perfect heuristics? That is, for each node, in which order should child nodes be expanded such that the $\alpha - \beta$ algorithm expands the smallest number of nodes? Write down the names of the all nodes in the order in which they should be expanded. For example, `adjkelmbfnogpqchrsitu` would be the normal left-to-right expansion.

1.5 (10 points) The perfect heuristic as discussed in the previous question is not achievable in practice. Why? Despite that fact, what could be done to increase the amount of pruning that $\alpha - \beta$ search does?

2. (20 points) The Fhourstones Benchmark is an benchmark that measures performance of a CPU on integer arithmetic. The benchmark consists of running an fairly optimized version of alpha-beta search using a transposition table to solve Connect-4, that is to search the complete game tree from the initial position to determine the value of the initial position.

Suppose the algorithm can generate $10^7 = 10000000$ states per second on your computer (you can run the algorithm on your machine to see how fast it would be for you, if you want). In total it needs to generate $1479113766 \approx 1.5 * 10^9$ states to finish the search. That means, the algorithm takes about 2.5 minutes to complete.

The game can last for at most 42 steps. Suppose the algorithm only has random move ordering. That means, the reduction of the tree due to pruning is equivalent to a reduction of the depth of the tree to $\approx 3/4$ of the original depth.

The maximal branching factor of the game is 7. However due to the transposition table, many of the successors of a state will be identical to other states that have already been explored. Those successors are essentially ignored.

2.1 (10 points) Given the assumptions above, what is the effective branching factor of the tree that we get due to using a transposition table? (Hint: use the size and depth of the tree to compute the branching factor)

2.2 (10 points) Suppose we would skip the alpha-beta pruning in the algorithm (but still use a transposition table). How long would it take to solve the game?

2.3 (10 bonus points) Suppose we had a heuristic that we could use for better move order. Adding the heuristic would slow down the search to about half the number of states per second but would lead to almost perfect pruning. Is it worth adding the heuristic to the code?

3. (20 points) Minimax (and therefore also the $\alpha - \beta$ algorithm) only works for two-player, turn-taking, zero-sum games. Why? What might happen if you use Minimax to play a two-player, turn-taking game that is not zero-sum? What might happen if you use Minimax to play a game that is not turn-taking?