

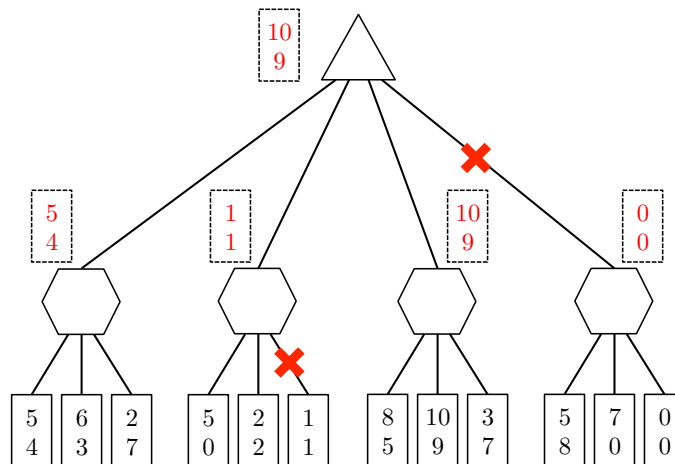
Q1. Game Trees and Pruning

You and one of the 188 robots are playing a game where you both have your own score.

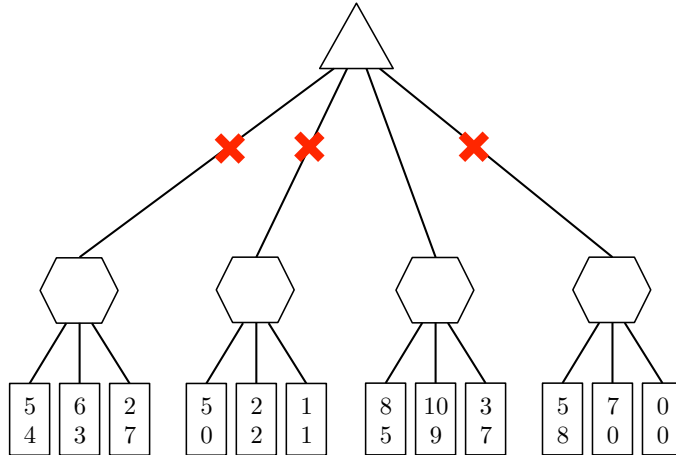
- The maximum possible score for either player is 10.
- You are trying to maximize your score, and you do not care what score the robot gets.
- The robot is trying to minimize the absolute difference between the two scores. In the case of a tie, the robot prefers a lower score. For example, the robot 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. The scores are shown at the leaf nodes with your score always on top and the robots score on the bottom.

(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.



Q2. Game Trees

(a) Alpha-beta pruning true/false For each true/false question, circle the correct answer. Missing choices and wrong choices with no explanation are worth zero.

(i) [*true* or *false*] Minimax search with alpha-beta pruning may not find a minimax optimal strategy.

False. Alpha-beta will always find the optimal strategy for players playing optimally.

(ii) [*true* or *false*] Alpha-beta pruning prunes the same number of subtrees independent of the order in which successor states are expanded.

False. If a heuristic is available, we can expand nodes in an order that maximizes pruning.

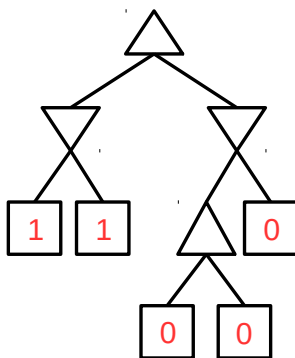
(iii) [*true* or *false*] Minimax search with alpha-beta pruning generally requires more run-time than minimax without pruning on the same game tree.

False. Alpha-beta will require less run-time than minimax except in contrived cases, which is the whole point of pruning.

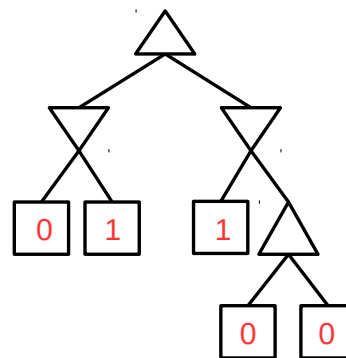
(b) For each of the following minimax game trees (max is at the root), fill in the leaf nodes with 0 or 1 as utility values. For the left tree (i), fill in the leaves so that as many leaves will be pruned by alpha-beta as possible. For the right tree (ii), fill in the leaves so that no leaves are pruned. Note that the two trees are slightly different.

Assume (1) left to right traversal while pruning, and (2) leaf values are potentially drawn from $[-\infty, \infty]$ (not from $[0, 1]$).

(i) Maximum Pruning



(ii) No Pruning



These answers are one example, and other solutions are also possible. For (i), the values chosen should cause the last leaf node to be pruned.