INF 4130 Exercises, alpha-beta-cutoff 2014

Exercise 1

Study figures 23.13 and 23.14 (pages 735 and 736 in B&P) where -1 and +1 is used to indicate win and loss, respectively. Look at all nodes and make sure you understand how values for the internal nodes are calculated with the min/max-algorithm. Always keep in mind that values indicate the situation for the player with the opening move – *A*. For *B* smaller values are better. (Note also that in exercise 3 we negate (* – 1) values on every other level so that we can always maximize!)

Exercise 2

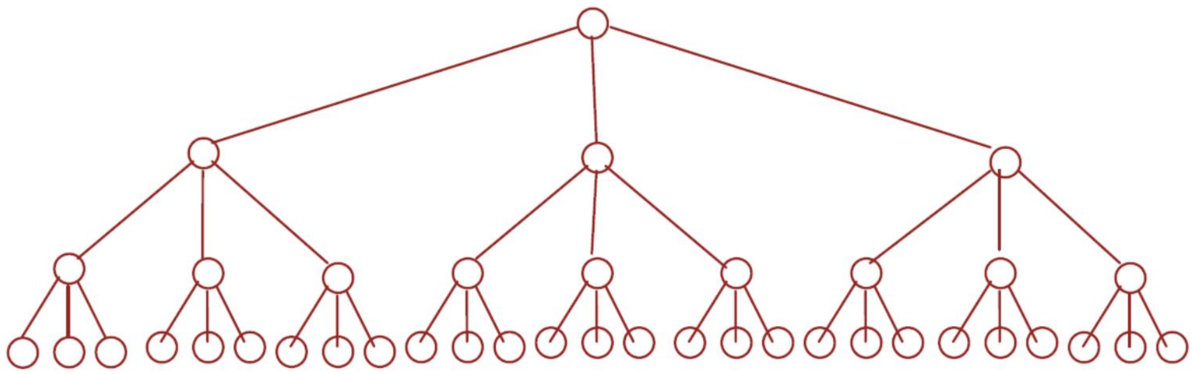
Study figures 23.16 and 23.17 (pages 738 and 740 in B&P) and check that your understanding of alpha-beta-pruning is correct; then solve exercise 23.22 in the text book (B&P).

Exercise 3

Go through the program on page 741 and discuss the solution chosen there, where values are negated (* – 1) on every other level. (Note that there are some typos in the program, see the lecture slides.) Finally assume that the nodes (among them *X*) are objects of a class with attributes `bestMove` (typed with the same class) and `value` (real) indicating the best move from, and the alpha/beta values of *X*, viewed from the player with the move in state *X*.

Exercise 4

Rune Djurhuus, in his lecture 23/9, said that if we are lucky(?) enough to always look at the best move first we get good pruning. He even claimed that if we go down to depth *d*, with a branching factor of *b*, the search time with alpha/beta-pruning is $O(\underbrace{b \cdot 1 \cdot b \cdot 1 \cdot b \ldots}_{d \text{ factors}})$, instead of $O(\underbrace{b \cdot b \cdot b \ldots}_{d \text{ factors}})$. We shall not attempt to prove it, but instead look at a concrete example. We let *d* = 3, and *b* = 3, and get the tree below. Mark the branches you have to evaluate (and thereby the ones you can avoid). The tree has 39 edges, how many do you avoid looking at?



EXTRA: Assume you are unlucky(?) and always look at the best node *last*. Will you get any pruning at all?

Exercise 5 (not central to the course)

Assume you are playing the game of NIM, with two piles, and that it is your move, that no pile is empty, and that the piles are of different size (number of pebbles or matches, or whatever). Try to come up with a strategy that guarantees victory.

Exercise 6 (From the exam, 2009)

We shall look at game trees, and we assume that the root node of the tree in the figure below is representing the current situation of a game (that we do not describe further), and that it is player A's turn to move. The other player is B, and A and B alternately make moves. Player A wants to maximize the values of the nodes while B wants to minimize them.

Player A shall make considerations for deciding which move to make from the root situation, and the tree in the figure below shows all situations it is possible to reach with at most four moves from the current situation. A has a heuristic function (that is, a function that for a given situation gives an integer) that he uses to evaluate how god the situation is for him. A uses this function for situations where he terminate the search towards deeper nodes. For each terminal node in the tree below this function is evaluated and the value appears in the nodes.

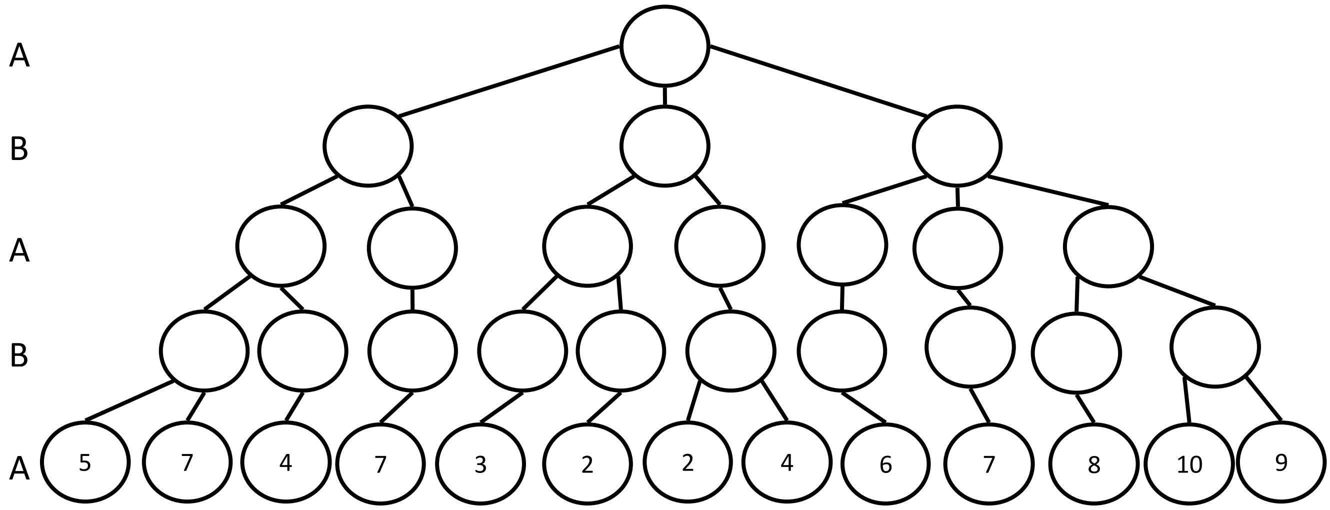


Figure 6.1 A game tree with values in the terminal nodes.

Question 6.a

Using the heuristic values in the terminal nodes, indicate how good the situation is for A in each of the other nodes. What is the best value player A can achieve regardless of how well B plays.

Question 6.b

We now assume that we are back to the start-situation, no nodes have values. We start the algorithm again, and A will then make a depth-first search in the tree from the root node, down to the depth of the tree above. In each terminal node A computes the value of the heuristic function (and thereby gets the value given in the corresponding terminal node in the figure above). The search is done from left to right in the figure above. Indicate which alpha- and beta-cutoffs you will get during this search.

In the drawing at the attached sheet you can simply place a clear 'X' at the root nodes of the sub trees that are not visited because of alpha- or beta-cutoffs. Give a short but explicit explanation for each cutoff (and for this you may suitably give names to some of the nodes in the figure).