

Minimax Search

- a. Values:
 - Max (topmost value) = 7
 - Min (from left to right) = 3, 7, 5, 6
- b. For max value, player will attempt to maximize the value chosen (hence, 7). However, the options that he can choose from is based on the decision of the min player, and the min player will always choose the lowest score of each option that he will present to the max player (hence: 3, 7, 5, 6).
- c. Max: a_2 , Min: c_1 in that order

$\alpha - \beta$ Pruning

- a. d_2, d_3, d_4, e_4
- b. First, there is no α value yet. So, all b branches need to be explored. This will result in $\alpha = 3$ ($\beta = 3$, also because best Min value as well so far).

Then, search for next β values in c branches, since they are all bigger than α , none of them are pruned. This yield $\beta = 7$ and α is updated to $\alpha = 7$.

When d branch is search it gives $\beta = 5$, which is $\beta \leq \alpha$ so d_2, d_3, d_4 are pruned because the Min player will always give 5 if this option is chosen, and the max player will never take it over the previous action which give the score of 7, thus subsequent search is useless.

e_1, e_2 are searched because the score they give since they are all bigger than α . Then, e_3 is searched giving $\beta = 6$ which is $\beta \leq \alpha$ ($\alpha = 7$ now, not $\alpha = 3$ anymore). Thus, e_4 is pruned.

Expecti-Minimax Search

- a. Values:
 - Max (topmost value) = 1.5
 - Chance (from left to right) = 1.5, -0.5
 - Min (from left to right) = 2, 1, 0, -1
- b. a_1