# Minimax Search

1. Values:
   * Max (topmost value) = 7
   * Min (from left to right) = 3, 7, 5, 6
2. 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).
3. Max: , Min: in that order

# Pruning

1. First, there is no value yet. So, all branches need to be explored. This will result in (, also because best Min value as well so far).

Then, search for next values in branches, since they are all bigger than , none of them are pruned. This yield and is updated to .

When branch is search it gives , which is so 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.

are searched because the score they give since they are all bigger than . Then, is searched giving which is ( now, not anymore). Thus, is pruned.

# Expecti-Minimax Search

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