Question 1: Living in Another World

Imagine a world that is much more exciting than the one with the three toy blocks. This world consists of four beer bottles A, B, C, and D (Amstel, Becks, Corona, and Duff). They can be arranged in any order from left to right, except that bottle A can never be further to the right than bottle D. For example, ABCD, CBAD, and CADB are possible states of our world, whereas DCBA, CDAB, or BCDA can never occur. The world can be manipulated by the schema swap(x, y), which swaps the bottles in positions x and y. For example, swap(1, 2) turns state BCAD into CBAD. However, swap(1, 2), swap(2, 3), and swap(2, 4) are the only three available operators.

a) Draw the state-space graph of this world. You do not need to draw any bottles; just use four-letter sequences to describe states.
b) Assume that your world is in the state ADBC, but you would like it to be in state CBAD. Use best-first search to find a solution that requires a minimum number of operations. To do this, first define your estimation function \( f'(n) = g'(n) + h'(n) \). Then write down the resulting search tree, indicate the order in which nodes were created, and for each node \( n \) give the value of \( f'(n) \).

Simplest solution:

\[ f'(n) = \text{number of operations already performed} + \frac{\text{number of bottles in incorrect position}}{2} \]

Without the division by two it would not be an optimistic evaluator.

The number to the left of the letters indicates the order of creation, the number in brackets to the right is the value of \( f'(n) \) written as \( g'(n) + h'(n) \):

\[
\begin{align*}
1: & \quad \text{ADBC} \quad (0 + 2) \\
\quad \text{swap}(2, 3) & \quad \text{swap}(2, 4) \\
2: & \quad \text{ABDC} \quad (1 + 1.5) \\
\quad \text{swap}(1, 2) & \quad \text{swap}(2, 4) \\
3: & \quad \text{ACBD} \quad (1 + 1.5) \\
\quad \text{swap}(1, 2) & \quad \text{swap}(2, 3) \\
4: & \quad \text{BADC} \quad (2 + 2) \\
5: & \quad \text{ACDB} \quad (2 + 2) \\
6: & \quad \text{CABD} \quad (2 + 1) \\
\quad \text{swap}(2, 3) & \\
7: & \quad \text{ABCD} \quad (2 + 1) \\
\quad \text{swap}(2, 3) & \\
8: & \quad \text{CBAD} \quad (3 + 0) \\
\quad \text{done!} & \\
\end{align*}
\]

The solution is:

\[
\text{swap}(2, 4) \\
\text{swap}(1, 2) \\
\text{swap}(2, 3) \\
\]

**Question 2: Be a Game-Playing Computer**

It is your turn to do some of the alpha-beta pruning. The tree below indicates the complete Minimax tree for a particular problem (first move by MIN, then MAX, and then MIN again – notice that is different from our previous examples, where MAX started). The number at each leaf $p$ indicates the value of the static evaluation function $e(p)$ if it were computed at that leaf.

a) Now your job is to check the boxes under those leaves that do not need to be created and evaluated thanks to the alpha-beta pruning.

b) Which move (the left or right one) should MIN make, and why, i.e., what exactly is the advantage of making this move over making the other one?

MIN should make the left move, because it guarantees MIN a score of -2 or less, whereas the right move would lead to a score of 1 if MAX plays optimally.