1. What is the advantage of adding alpha-beta pruning to a minimax algorithm? [4]
   It on average speeds up minimax algorithms by reducing the number of nodes that need to be examined. This is achieved by “pruning” nodes which have been found not to change the result produced by the algorithm.

2. What is the advantage of adding a move-ordering heuristic to a minimax algorithm with alpha-beta pruning? [5]
   It on average speeds up the algorithm by reducing the number of nodes that need to be examined. This is achieved by increasing the chance that “good” moves are examined early on, which results in increased alpha-beta pruning.

   I) Solution availability: i.e., you always have the solution of the previous iteration available during the execution of the current iteration (this is particularly useful when under a time constraint).
   II) Information gleaned during the current iteration can be employed to increase pruning in successive iterations (e.g., history table). Because successive iterations require exponentially more CPU time, the overhead of searching at lower depths is typically insignificant while increased pruning at higher depths can be very significant.

The next three questions are about the following adversarial “chance” tree.

4. Calculate the EXPECTIMINIMAX values for nodes B, C and D in the above adversarial “chance” tree. Show your calculations! [3]
   - EXPECTIMINIMAX(B) = 0.1 * 2 + 0.9 * 1 = 0.2 + 0.9 = 1.1
   - EXPECTIMINIMAX(C) = 0.2 * 3 + 0.2 * 2 + 0.6 * 5 = 0.6 + 0.4 + 3 = 4
   - EXPECTIMINIMAX(D) = 0.7 * 1 + 0.3 * 8 = 0.7 + 2.4 = 3.1
5. Which action will MAX choose, \(a_1\), \(a_2\), or \(a_3\)? Explain your answer! [2]

MAX will choose action \(a_2\) because it has the highest EXPECTIMINIMAX value.

6. If the utility values given for MIN were multiplied with a positive constant \(c\), which action would MAX then choose? Explain your answer! [3]

MAX would still choose action \(a_2\) because multiplying with a positive constant is a positive linear transformation and such transformations do not change decisions made on the basis of EXPECTIMINIMAX values.

The remaining questions are about the following adversarial search tree. State evaluation heuristic values for the max player are provided in the form of numbers following the letter labels of the states (e.g., B5 indicates that the heuristic value of state B for the max player is 5). The order in which successors are generated is from left to right. Example: A generates first B, then C, and finally D.

You are also provided the following history table (note: non-listed moves default to a HT value of zero):

<table>
<thead>
<tr>
<th></th>
<th>AC</th>
<th>AD</th>
<th>DK</th>
<th>BE</th>
<th>BF</th>
<th>BG</th>
<th>FN</th>
<th>FO</th>
</tr>
</thead>
<tbody>
<tr>
<td>HT</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

7. Give the execution trace of HTQSABDLM(A,3,2,\(-\infty,\infty\)). [20]

Define: \(DLM = HTQSABDLM\), \(MinV = HTQSABMinV\), \(MaxV = HTQSABMaxV\)
8. What is the PV found by HTQSABDLM(A,3,2,−∞,∞)? [3]
   \textit{ADKT}

9. Which nodes (if any) are pruned by HTQSABDLM(A,3,2,−∞,∞)? [4]
   \textit{E,L,M,V,F,N,W,AF,O}