1. 15 points
   Consider a problem in which you have a table and three blocks, all of the same size. Each block is indicated by a letter, A, B, or C. A block can be on the table or on top of another block. At start, block A and block B are on the table, block C is on top of A. One block at the time can be moved if nothing is on top of it. A block can be moved either to the table or on top of another block. Your goal is create a stack with the three blocks in the order A, B, C from bottom to top.

   1. Describe the state space representation of the problem (i.e., what are the states, initial state, goal condition, and actions).

   2. Propose a non trivial (i.e. not \( h(n) = \text{constant} \)) heuristic. Is your heuristic admissible?

2. 15 points
   Suppose you decide to remove the CLOSED list (i.e. the place where the explored nodes are saved) from the A* algorithm.

   1. If the heuristic used is admissible, will the new algorithm still be guaranteed to find the optimal solution? Explain why (or why not).

   2. Will the new algorithm be more or less efficient? Explain.

   3. Will the new algorithm still be complete? Explain.

3. 15 points
   A drawback of A* is its memory requirement since the OPEN list (i.e., the frontier) might get very large. Suppose you modify A* as follows: You keep in OPEN only the best N nodes, with N > 1 (i.e, the nodes with lower costs). When the OPEN list is full and a new node has to be stored, the worst node is deleted from OPEN and removed from consideration.

   1. If an admissible heuristic is used, will the modified algorithm find the optimal solution? Explain why (or why not).

   2. If a perfect heuristic is used (i.e. \( \forall n \ h(n) = h^*(n) \), the cost of the optimal path from \( n \) to goal), will the modified algorithm find the optimal solution? Explain.
4. 15 points
Show, examining the branches in the following game tree from left to right:

1. the backed-up values for the all nodes,
2. the branches that are pruned by alpha-beta pruning. For each branch pruned, write the condition that is used to do the pruning.

5. 15 points
Suppose you traverse a game tree (not just the one above, any game tree) by visiting children in right-to-left order instead of left-to-right order.

1. Could the minimax value computed at the root change? Explain why (or why not).
2. Will the number of nodes pruned by alpha-beta pruning change or not? Explain your reasoning.
3. Explain in general what ordering of the nodes in the game tree will increase pruning.

6. 25 points
Answer these questions briefly but precisely. Always explain your reasoning.

1. Are there conditions under which depth-limited search is guaranteed to find an optimal path? What about iterative deepening depth-first search?
2. Will the path returned by the uniform cost search algorithm change if we add a positive constant $C$ to the cost of each action? What if we multiply the cost of each action by a positive constant $C$?
3. Is it possible for alpha-beta pruning and minimax to return different values at the root of the game tree for the same problem?
4. Some search algorithms use randomness for avoiding getting trapped in local maxima. Describe briefly two of them and explain how they use randomness.
5. Explain what is the difference between a rational agent and an omniscient agent and explain why the difference is important.