1. (25 points) Suppose a search problem is represented by the following graph:

![Graph with nodes and edges](image)

where the weight on each edge is its cost, \( A \) is the start node and \( M \) is the goal. The heurist function is given as follows:

\[
\begin{array}{|c|c|c|c|c|c|c|c|}
\hline
n & A & B & E & F & G & H & K & M \\
\hline
h(n) & 25 & 15 & 12 & 20 & 30 & 10 & 5 & 0 \\
\hline
\end{array}
\]

Assuming the neighbors of a node are explored in the alphabet order of their names, please list the nodes entering the Open and Closed lists as the search goes on using the following strategies, respectively:

- Uniform-cost search: \( f(n) = g(n) \), the total cost of \( n \) from the start node.
  
  **Answer key**: The Closed List is \( A, F, H, B, G, E \).

- Best-first search: \( f(n) = h(n) \), the given heuristic function.
  
  **Answer key**: The Closed List is \( A, H, K \).

- \( A^* \): \( f(n) = g(n) + h(n) \)
  
  **Answer key**: The Closed List is \( A, H, F, B, E, G \).
2. (25 points) Compute the minimax value of each node using the minimax algorithm with alpha-beta pruning on the following example:

Answer key: The value of the root is 8. Only one node (the fourth leaf node from at the bottom from left) is pruned.

3. (25 points) The so-called Scholten’s coffee can problem is stated as follows: In Scholten’s house there are white and black coffee beans. In the morning, when bored Scholten repeatedly does the following: He takes out two beans X and Y from the can. If this is not possible he stops. If they are the same he puts a black bean back in the can. If they are different he puts a white bean back in the can.

Suppose the initial can is given as a prolog list such as

initialCan([b, w, b, w, b, w, b, w, b, w]).

where b and w stand for black and white beans, respectively. If Scholten always takes two beans from the beginning of the list and puts a new bean at the beginning of the list, please write a prolog program to decide the last bean remaining in the can for any given initial can.

Answer key:

initialCan([b, w, b, w, b, w, b, w, b, w]).

scholten([X], X).

scholten([X, X | Z], Answer) :- scholten([b | Z], Answer).

scholten([X, Y | Z], Answer) :- X =\= Y, scholten([w | Z], Answer).

run(Answer) :- initialCan(L), scholten(L, Answer).

The Prolog query is ?- run(A).
4. (25 points) Prove by resolution that the clauses (1)-(6) is unsatisfiable:

\[
\begin{align*}
(1) & \quad a \lor b \lor c \\
(2) & \quad a \lor \neg b \lor \neg c \\
(3) & \quad \neg a \lor b \\
(4) & \quad \neg a \lor \neg b \\
(5) & \quad b \lor \neg c \\
(6) & \quad a \lor \neg b \lor c
\end{align*}
\]

Answer key:

\[
\begin{align*}
(7) & \quad a \lor b \quad \text{from (1), (5)} \\
(8) & \quad b \quad \text{from (3), (7)} \\
(9) & \quad a \lor \neg c \quad \text{from (2), (8)} \\
(10) & \quad a \lor c \quad \text{from (6), (8)} \\
(11) & \quad a \quad \text{from (9), (10)} \\
(12) & \quad \neg b \quad \text{from (4), (11)} \\
(13) & \quad false \quad \text{from (8), (12)}
\end{align*}
\]