Solution 2
CIS 471/571, Fall 2014

1. Consider the problem of solving two 8-puzzles in two boards. The two boards have different initial positions. Each step can move one tile in one board.
   a) Give a complete problem formulation based on search.
   b) How large the reachable state space is? Give an exact numerical expression.
   c) Suppose we make the problem adversarial as follows: the two players take turns moving; a coin is flipped to determine the board on which to make a move in that turn; and the winner is the first to solve one 8-puzzles problem in one board. Is there an algorithm can be used to choose a move in this setting? If yes, explain why. If no, also explain why.

Ans:
   a) State: A state description specifies the location of each of the eight titles of the two eight-puzzle and the blank in one of the nine squares of the two eight-puzzle
   Initial state: Any state can be designated as the initial state except the state that the two puzzle are the same.
   Actions: The formulation defines eight actions as movements of the blank space Left, Right, UP, or Down. Different subsets of these are possible depending on the blank position of the first puzzle and the second puzzle.
   Transition model: Given a state and action, this returns the resulting state.
   Goal test: This checks whether the state matches the goal configuration.
   Path cost: Each step costs 1, so the path cost is the number of steps in the path.

   b) \[
   \frac{9! \times 9!}{2 \times 2} = 3,292,047,360
   \]

   c) Expectiminimax is the best for “win the game.” This setting is similar to backgammon with two players. When a move happens, we need to know the movement of next step and who will play this move. However, if the goal is “not to lose the game,” then no one will take the move to one step away from the completion of one board, because there is 50% chance that the opponent get the turn for that board next time.

2. Prove that alpha-beta pruning takes time O(b^m/2)) with optimal move ordering, where m is the maximum depth of the game tree and b is the branch factor.

In an optimal move ordering, for even depth, only the children of first node in the last level will all be visited. For the other nodes in the last level, only the first child will be visited and the others will be pruned. So the time it takes will be b*1*b*1*b*1... so the time complexity is O(b^(m/2))
3. Programming assignment:
Create a program that can solve a cryptarithmetic problem. Each letter must be assigned a unique digit from zero to nine, and no term is allowed to have leading zeros. (For example, F and T cannot be zero in the following:)

\[
\begin{align*}
\text{F O U R} \\
\quad \text{-} \quad \text{T W O} \\
\hline \\
\text{T W O}
\end{align*}
\]

The program should allow a user to enter one minuend, one subtrahend, and one difference term. For simplicity, you may limit the minuend term to four letters and the subtrahend and difference terms both to three letters. The program needs to output all valid assignments of the variables and digits, or output that no valid assignment exists.

Solve the problem with a method you think as the best fit. It could be CSP or other methods. Submit your program as an attachment file (e.g., java source code) and report the number of steps (number of assignment of values) to solve the above (FOUR-TWO=TWO) problem. Note for any other combinations of input minuend term (4 letter), subtrahend term (3 letter), and difference term (3 letter), your program also needs to either find all valid assignments, or output no valid assignment exists.

Sample of solvable combination: four-two=two

\[
\begin{align*}
1468 - 734 &= 734 \\
1530 - 765 &= 765 \\
1672 - 836 &= 836 \\
1692 - 846 &= 846 \\
1734 - 867 &= 867 \\
1856 - 928 &= 928 \\
1876 - 938 &= 938 \\
\end{align*}
\]

Sample of unsolvable combination: abbc - abb = aab

The program needs to need to record the number of steps/assignments. For example, backtracking with forward checking takes 23 steps to solve four – two = two.