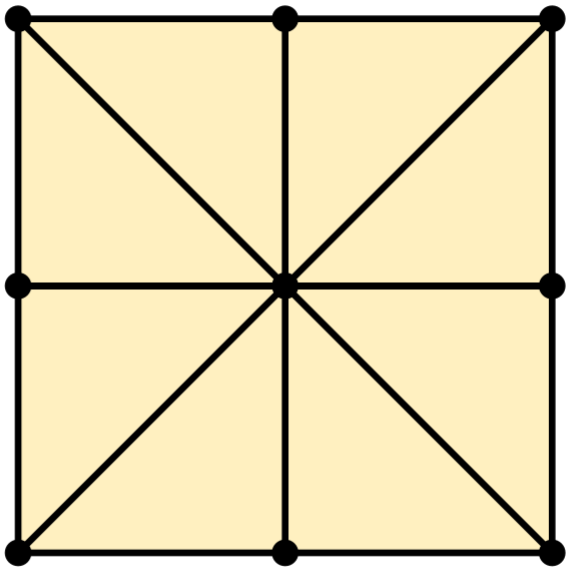
Homework Assignment 3

Submit Assignment

Due Mar 26 by 11:59pm **Points** 25 **Submitting** a text entry box or a file upload
Available Mar 17 at 12am - Mar 26 at 11:59pm 10 days

The purpose of this homework assignment is to provide you some practice in using the minimax search algorithm to design an automated agent that can play a 2-agent game. Your goal is to design an agent that can play the version of the Three-men's Morris game (https://en.wikipedia.org/wiki/Three_men%27s_morris (https://en.wikipedia.org/wiki/Three_men%27s_morris)) described below:

- The game board has 3 horizontal lines, 3 vertical lines and 2 diagonal lines as shown below.
- 

- Each player has three pieces at the start of the game.
- The board is empty to begin the game, and players take turns placing one piece on each turn. A piece can only be placed on an empty intersection.
- Once all pieces are placed (assuming there is no winner by then), play proceeds with each player moving one of their pieces per turn. A piece may move to any adjacent linked position that is currently unoccupied. If no adjacent positions are empty, the player loses its turn and the other player makes their move.
- The winner is the first player to align their three pieces on a line drawn on the board.

- A draw is declared if the board returns to a previous state.

1. Your first task is to write a computer program in the programming language of your choice that implements the standard minimax algorithm to figure out a plan to play the game. Run your program in following two ways: (a) assuming that Max starts first, (b) assuming that Min starts first. Compute the average running time and the average search tree size for both runs. (60 points)
2. Your second task is to modify the program you wrote in response to Question 1 to implement the minimax algorithm with alpha-beta pruning. Again note the average running time and the search tree size. How much shorter is the new search tree compared to the old one from Question 1? (40 points)

Please submit the running times and search tree sizes along with your fully documented code. The documentation must include:

- a description of the representation you chose to represent the problem states, and
- a brief pseudocode description of each procedure/function along with a description of the input and output for the procedure/function.