

685.621 Algorithms for Data Science
Homework 5
Assigned at the start of Module 10
Due at the end of Module 11
Total Points 100/100

Total Points 100/100

Collaboration groups have been set up in Blackboard. Make sure your group starts one thread for the collaborative problem. You are required to participate in the collaborative problem. Do not directly post a complete solution, the goal is for the group to develop a solution after everyone has participated. There are two parts for this HW; the first part is a theoretical set of problems with one individual problem and second with a collaborative problem; the second part is a programming problem that has a collaborative part and a programming part. You are only required to do one of the two problems.

Part 1

1. [100 points]

(a) [50 points] ***Note this is not Collaborative Problem***

Based on the rules of Depth-First Search draw the tic-tac-toe board with the leaf node starting as an empty board (as shown in the Game Theory document), then expand the your tree to show the first four left leaf nodes. What does this board look like assuming your AI places pieces on the board from top to bottom and left to right?

(b) [50 points] ***Note this is a Collaborative Problem***

From your solution in part a) write an algorithm using a recursive call to search for the next move. Your decision in this algorithm should be based on the ability for the AI to WIN or at a minimum have a DRAW. You will need to pass in the board state at every recursive call. You must ensure that a WIN for your AI is always taken first and a WIN for your opponent is always blocked. Based on your development of the algorithm what is the running time of your algorithm?

Note: When developing a search tree as shown in the Game Theory document, diving down the tree is a recursive call and exploring additional child nodes at a specific level is an iterative move.

Part 2

2. [100 points]

(a) [50 points] ***Note this is not Collaborative Problem***

in the tic-tac-toe code provided add the following method to allow an unbeatable AI in your game.

- Best Move (Provided)
- Implement a method that uses conditional statements to play against a user of your code. You must always check to see if your AI has a WIN and take that move, if not check to ensure your opponent does not have a WIN, if your opponent has a WIN possibility you must block your opponent.
- The following is a bonus problem to make a much smaller code base than your implemented conditional statements.

- MiniMax
- Alpha Beta

(b) [50 points] ***Note this is a Collaborative Problem***

Show how the MiniMax algorithm from the Game Theory document is to be implemented for the tic-tac-toe game. You will need to alter the provided pseudo code to show how the game board is passed in.