Isfahan University of Technology
Game Theory
Dr. Manshaei
Homework #7
Due date: Friday, 6th Bahman 1402, at 23:59

We only accept the homework **delivered via *Yekta*, before the deadline**. If you have any questions or concerns about this homework, feel free to contact Mr. Mirdamadiyan via *Telegram* (Preferred) or *Email*.
Please be aware that this homework is entirely optional and serves as a bonus assignment. If you've missed any previous assignments or would like to increase your grade with bonus points, feel free to submit it. Best of luck!

This assignment comprises two programming questions, and you are free to use any programming language to implement the algorithms. It is essential to note that using code from other sources or previously implemented by users is not acceptable.

Problem 1. Write a program that obtains the preference matrix of boys and girls and finds a stable marriage. Note that if you are a boy, you should implement the boy-optimal approach, and if you are a girl, you should implement the girl-optimal approach. Consider the following sample:

Boys preference				
1	1	2	4	3
2	1	2	3	4
3	3	1	2	4
4	2	3	1	4

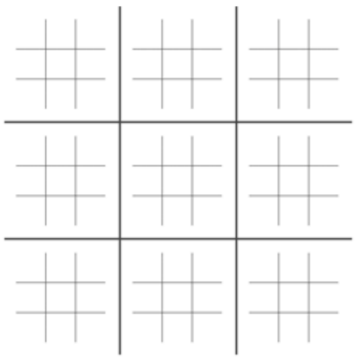
Girls preference				
1	3	4	1	2
2	2	3	4	1
3	1	2	3	4
4	3	4	2	1

Boy Optimal	
Boys	Girls
1	4
2	2
3	3
4	1

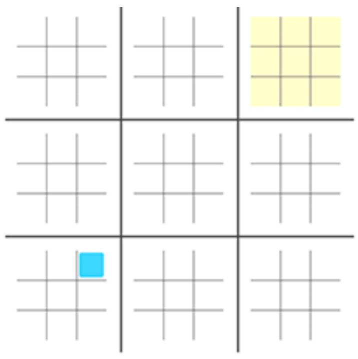
Girl optimal	
Girls	Boys
1	3
2	2
3	1
4	4

Problem 2. Consider the following variation of Tic-Tac-Toe:

The board consists of nine small 3-by-3 grids, collectively forming a larger 3-by-3 grid.



Players take alternating turns, and a player wins a small grid, similar to regular Tic-Tac-Toe, by placing three of their symbols in a row. Upon winning a small grid, the player marks their symbol in the corresponding position on the larger grid. The game concludes when a player achieves three symbols in a row on the larger grid or ends in a tie if all squares are exhausted. To enhance gameplay, a player must place their symbol in the small grid corresponding to the position in the small grid of the previous move.



In this example, the blue player played the top-right square in the bottom-left small grid, so the red player must now take their turn on the top-right board (colored in yellow). If all squares in the small grid have been exhausted or if the grid has already been won, the player may choose to make his move anywhere on the board.

Here is an example provided to illustrate the gameplay.

You must write a minimax-based algorithm to solve this game. You are open to using any programming language or framework, such as Pygame, to build a UI for this game. Additionally, it is advisable to incorporate alpha-beta pruning in your algorithm to expedite decision-making at each step. You may also choose to employ a heuristic function for depth limiting in tree traversal.

Finally, record a 5-10 minute video describing how your algorithm works, the tools you utilized for this question, and run your program to demonstrate a round of this game.

Good Luck.