**Artificial Intelligence**

Name and ID \_\_\_\_\_Tommy Le U52555883\_\_\_\_\_\_\_\_\_

**Assignment 3**

**Due date: March 10 at 11:59pm**

You may work alone or in groups of two.

**Problem 1** (2 points)

Read Chapters 4 and 5 of the textbook and answer the following questions.

**(a)** Explain the main differences between puzzle-solving search and game playing.

**(b)** Argue that game playing is inherently more difficult than solving one-player puzzles.

1. **Primarily the difference revolves around the number of available agents. In puzzle searching there’s only one agent finding a specific solution or a set of solutions, while game playing both agents are trying to compete with each other. There are situations where it’s based on whoever could reach the solution first, but’s it’s also a win if one person reaches a final state that is still consider more favorable than the other.**
2. **Game playing is more difficult that one player solving puzzles because not only are you including the max outcome for yourself, your also trying to find moves that minimize the second player. Having more agents meaning more functions looking for best outcomes. To win you need ways to diminish other players. This can become more complicated with more than 2.**

**Problem 2 (3 points)**

Consider the zero-sum game tree shown below. Trapezoids that point up, such as at the root, represent choices for the player seeking to maximize; trapezoids that point down represent choices for the minimizer.

(a) Assuming both opponents act optimally, carry out the minimax search algorithm. Write the value of each node inside the corresponding trapezoid and highlight the action the maximizer would take in the tree.

**A graphing on a notebook paper

Description automatically generated with medium confidence**

(b) Now reconsider the same game tree, but use alpha-beta pruning. Expand successors from left to right. Record the alpha-beta pair that is passed down that edge (through a call to MIN-VALUE or MAX-VALUE). Circle all leaf nodes that are visited. Put an `X' through edges that are pruned off.

**A paper with writing on it

Description automatically generated**

(c) True / False. Minimax and alpha-beta pruning are guaranteed to find the same value of the top node.

**True**

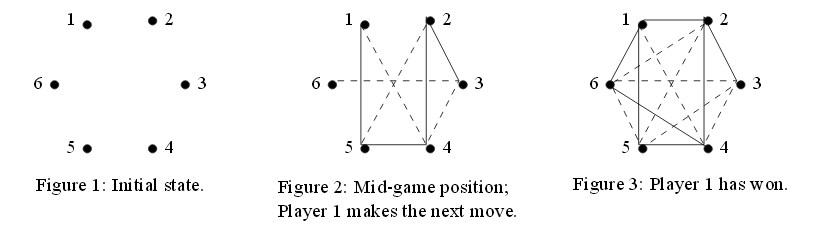
**Problem 3** (15 points)

The *hexagon game* involves two players, who gradually construct a six-vertex undirected graph with solid and dashed edges. Player 1 adds solid edges, whereas Player 2 uses dashes.

The players begin with a six-vertex graph that has no edges (Figure 1), and add new edges, one by one; Player 1 makes the first move. At each move, a player has to add a new edge between two vertices that are not connected by any old edge. Figure 2 is an example of a mid-game position, where Player 1 has to make the next move.

If Player 1 constructs a solid-line triangle, he loses the game; similarly, a dashed triangle means a loss of Player 2. For example, if the game ends as shown in Figure 3, then Player 2 has lost since he has constructed the dashed triangle “3-5-6.”

Implement a program for playing the hexagon game. Your program should prompt the user to enter a player number (1 or 2), and then act as the specified player. For example, if the user enters “1”, the program should make the first move.



**Submitting your assignment**

* Submission via Canvas Assignment.
  + It is your responsibility to submit these assignments in a timely fashion.
* All files should be zipped together.
* Name of your file must include your last name and ID number
* There should be a readme file explaining in detail the exact steps to be taken to compile and execute the code files and the title page
* In case of any code errors, partial credit may be offered based on the code and documentation.
* A report that presents the performance evaluation of your solution.

**Late Submission Policy**

* Late work will be not accepted.

**Rubric for Assignment 3**

**Problem 1 (2 points)**

1. [1/]
2. [1/]

**Problem 2 (3 points)**

1. [1/]
2. [2/]

**Problem 3 (15 points)**

* Compile, run and test (15 points)

[5/] code compile

[5/] Run the program

[3/] Test cases

[2/] Report