Assignment 3:
Hashi Game

Due Sunday, 5 May, 11:30pm

Hashi game[1] is a two-player zero-sum game played on a square grid of cells where some cells contain circles, which are called islands. Each island can have a label from 1 to 4, or be empty.

Each player must do one of the following **moves** when their turn comes:

- place a horizontal bridge to connect two labelled islands,
- place a vertical bridge to connect two labelled islands, or
- label an empty island with 3 or 4.

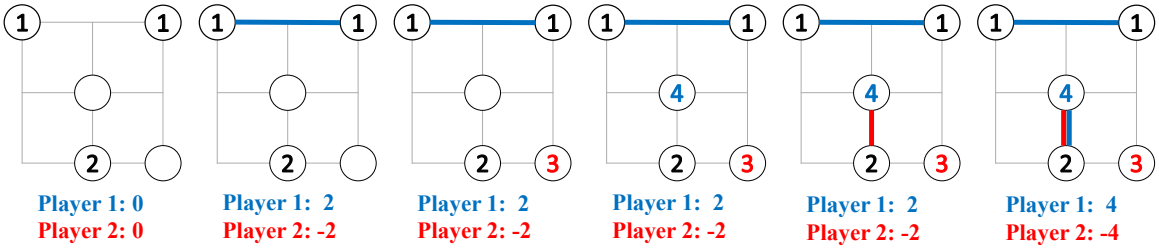
While making their move, players must follow the rules below:

- A bridge must not cross any other bridges or islands.
- At most two bridges connect a pair of islands.
- The number of bridges connected to each island cannot be greater than its label.

After a player's move, for every island affected by this move, if the number of bridges connected to the island is equal to its label n, the player gets n points and the opponent loses n points. Note that in this case, a player can get points from an island at most once, but they can get points from two islands with one move.

The game ends when there are no possible moves. The player with more points wins the game.

As an example, a possible play between two agents is presented below.



Note that, this execution might not contain the optimal moves for each player. It is only included here to show the rules of the game.

The goal is to design and implement an AI agent that plays Hashi game against a human.

[1]The rules of this game are different from the Hashi puzzle in Assignment 2. Please read this document accordingly.

The assignment consists of two parts:

1. Model this game as a game tree search problem (i.e., define the players, the states, the initial state, terminal states, the state transition function, the payoff function).
2. (provided that part 1 is completed) Implement the game in PYTHON using alpha-beta pruning.

Submit the following two files at SUCourse+:

- a pdf file (at most 2 pages) containing a description of your model,
- a zip file containing your PYTHON code, and the traces of two different runs of the program. In one of the runs, you should start the game, and in the other run, your AI agent should start the game.

Demos If your submitted implementation runs correctly and you can demonstrate it with your test instances successfully (without any bugs), then you will be invited to make a demo of your implementations. The 2nd part of the assignment will be graded at the demos.[2] The demos are planned for the week following the deadline and will be scheduled later on.

Collaboration You are allowed to work with another classmate. In that case, each team should submit one report in pdf, and one zip file at SUCourse+. Both team members should be present at the demos, if the team would like to make a demo of their implementation.

[2] Attention: If your implementation does not run correctly with any of your instances, you will not be invited to the demos and thus no credit will be given to the 2nd part.