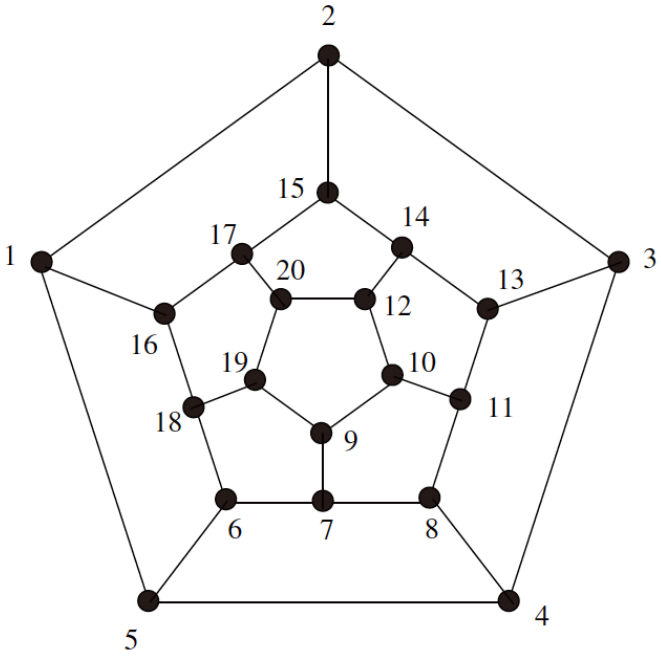
SI252 Reinforcement Learning **Name (Print):** _____
Spring 2022
Final Project
Deadline: 11:59am, June 26, 2022

This is the final project of **undergraduate students**. The following rules apply:

- Write the project report with LaTeX and output it as a PDF file.
- Programming with Python (**Do NOT accept other languages**).
- Submit your report (PDF File) and your programming files to the corresponding TA before 11:59 am of June 26, 2022.
- **Later project report receives No credit**.
- You may work in groups of up to three people. Each group will submit the project report together. The expectations for the project scope will increase depending on the number of students in each group, and for groups of two or three, we will also expect a short paragraph to explain the role of each group member along with the final report.

Problem	Points	Score
1	40	
2	30	
3	80	
Total:	150	

1. (40 points) **Maximal Cuts**. A dodecahedron graph is shown in the following figure:



Suppose that all edges have the same reward 1. We wish to partition the node set into two disjoint subsets such that the accumulated reward across the cut is maximized. This is the classical max-cut problem in graph theory, a typical combinatorial optimization problem.

(a) (10 points) Adopt the MCMC method to solve this problem.

(b) (10 points) (**Bonus**) Adopt the Cross-Entropy method to solve this problem.

(c) (10 points) (**Bonus**) Adopt the Monte Carlo Tree Search (MCTS) Method to solve this problem.

(d) (10 points) (**Bonus**) Discuss the pros and cons of each method, and then compare them.

2. (30 points) **Paper Summary**. To understand state-of-the-art results, it is important to read and parse research papers. If this is your first time reading a research paper, this link may be helpful. This resource may be even useful for veterans in research. It describes a multi-pass approach to reading papers.

   The papers that you can choose to read may vary in difficulty, and we do not expect anyone to understand all of the content in the papers. Nonetheless, it is important to start building your ability to read research papers, especially research papers out of your comfort zone. The notation, mathematics, and jargon of the paper matter less than the big ideas and context of the paper. Read papers, take notes, and answer the questions for each paper individually (see below).

   Questions for each paper:

   - What are the papers main contributions? Describe the core idea.
   - What was surprising, difficult, or confusing in the paper?
   - Were the experiments (if there are any) convincing?
   - How can these methods be applied in ways not described in the paper? Have you seen these ideas in other papers? Feel free to cite relevant work.

   Paper list:

   (a) (5 points) Human-level control through deep reinforcement learning, by Mnih et al, Nature, 2015.

   (b) (5 points) Mastering the game of Go with deep neural networks and tree search, by Silver et al, Nature 2016.

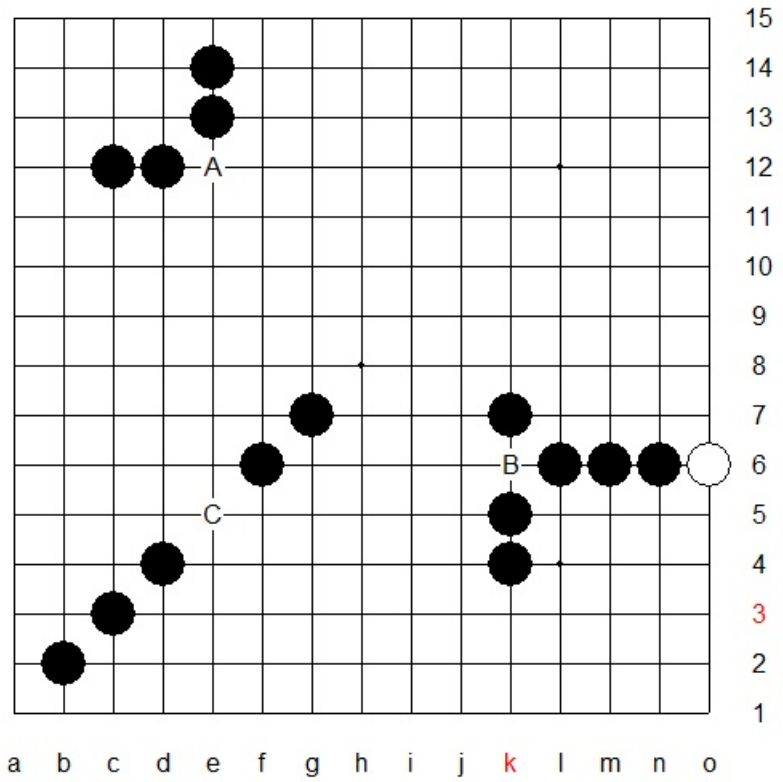
   (c) (5 points) Mastering the game of Go without human knowledge, by Silver et al, Nature 2017.

   (d) (5 points) A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play, by Silver et al, Science 2018.

   (e) (5 points) (**Bonus**) DeepStack: Expert-level artificial intelligence in heads-up no-limit poker, by Moravcik et al, Science, 2017

   (f) (5 points) (**Bonus**) Safe and Nested Subgame Solving for Imperfect-Information Games, by Brown et al, NeurIPS, 2017

3. (80 points) **Gomoku**. Gomoku is a two-person game with black stone and white stone. The forbidden rules for the black stone are shown in the following figure:



Read the GitHub reference codes of AlphaZero for Gomoku without forbidden rules. The corresponding details of implementation in Chinese can be found here.

(a) (20 points) Based on reference codes, implement the AlphaZero for Gomoku **with forbidden rules** in a 8 x 8 board. Then think about how to improve this version using RL knowledge obtained from SI252 course.

(b) (40 points) Without using neural network, implement Gomoku with forbidden rules in a N x N ($N \geq 8$) board with reinforcement learning techniques. Can it beat the AlphaZero version or not? Why?

(c) (10 points) Share experiences, lessons and insights from your implementation.

(d) (10 points) (**Bonus**) List your innovative parts such as user-interface and novel algorithms with better performances with supporting evidences.