Object-Oriented Programming, Junhee Seok, Korea University

Programming Project 3

Bingo Game

Due: 11:59pm 5/19 (Sun)

In this programming project, you will implement the strategy in a Bingo game.

Bingo Game

A bingo game is a board game using an N x N matrix board that is filled with numbers from 1 to N^2. In this project, you and the opponent play the game according to the below sequence.

1) You and the opponent randomly distribute numbers on the matrix boards.
2) At each turn, you and the opponent pick a number to mark, and open them at the same time.
3) You and the opponent mark the picked numbers on the board. You will mark two numbers unless you and the opponent pick the same number.
4) The players repeat turns of 2) and 3) until one wins or the game is drawn.
5) One wins if one marks all numbers of in M rows, columns, diagonals, or in their combinations. In other words, when M=3, the winning condition includes marking all numbers in 3 rows as well as marking all numbers in 2 rows and 1 diagonal.
6) If you and the opponent reach the winning condition at the same time, the game is drawn.

You get 3 point for a win, 1 point for a draw, and 0 for a defeat.

As you expect, you can easily win if you know the bingo matrix of your opponent. In this program, the bingo matrices of you and the opponent are generated by hidden parameters. By finding the hidden parameters from the calls of your opponent, you might be able to guess the bingo matrix of the opponent and use the information to win.

In the program, the bingo matrix size N is defined by BINGO_SIZE, and the winning condition M is defined by BINGO_CNT. By default, N = 20 and M = 5. The full league will be played for all possible pairs of players. Two players will play 100 games. All students will play in the league, which means you will play against your classmates. Player "Junhee Seok" and "Colin Seok" will also join the league.

Important Note

1. You should change the filename of `Bingo_2016000000.h` to `Bingo_<your student id>.h`, and properly include to your program. The class name also should be `Bingo_<your student id>`. It is very important to include all students in the league.
2. You should implement your codes in `myCall()` of `Bingo_<your student id>.h`. Do not create an additional cpp file. It is to make it easy to manage the codes of many students.

3. Whenever you modify a header file, you should "rebuild" to compile the program. Just "compile" won't reflect your changes.

Please answer the following problems.

1. Describe how a bingo matrix is generated by hidden parameters? What are hidden parameters?

2. Explain the strategy of Junhee and Colin in the given program.

3. Build a strategy that does not guess the opponent matrix. This strategy does not use the history of calling. Explain your strategy here. What is the result when this strategy is played with Junhee Seok and Colin Seok. Please include the code of this strategy in your report as well as the screen shot of the game result.

4. Build a strategy that does guess the opponent matrix. This strategy may want to use the history of calling. Explain your strategy here. What is the result when this strategy is played with Junhee Seok, Colin Seok, and the strategy of problem 3. Please include the code of this strategy in your report as well as the screen shot of the game result.

5. Build a strategy that guesses the opponent matrix as well as tries to confuse the opponent's guess assuming your opponent also tries to guess. This strategy may want to use the history of calling. Explain your strategy here. What is the result when this strategy is played with Junhee Seok, Colin Seok, and the strategy of problem 3 and 4. Please include the code of this strategy in your report as well as the screen shot of the game result.

6. Among the strategies of problem 3~5, pick one to submit for the full league with your classmates. Explain your choice.

Please submit

1. Source code of Problem 6 as `Bingo_<your student id>.h`
2. Report for the problems in doc, hwp, pdf or other readable format.
3. Please submit the format of "zip" file when you send your project file to TA
   - "Your student id number_Project3"

to the TA via jyoonkim@korea.ac.kr . Even though you cannot complete problem 4 and 5, you are still able to join the league by submitting your code of Problem 3.

If you have any question for this project assignment, please contact me or TA.