B351 Project

Team: Yitian Zhang, Tian Jin, Bokai Zhuang

Name of project: Impossible or not (card game)

First of all, the RULE of this game is between two players who have two piles of cards respectively. The two piles of cards are separated into odd and even group, which holds [1,3,5,7,9] for odd group and [2,4,6,8] for even group. The way players should play is that for each round, each player should draw a card from either odd group or odd group while truly telling the opponent what group the card is drew from. After that, there should be a third party to check which player wins by comparing the cards each player drew and the winner will earn one point for each win. After all, we can tell who will be the winner by telling who has earned most points until the cards are all drew to the end. Of course, there might be tie since each player will earn one point if there is a tie.

Few NOUNS you should know about our algorithm:

* Sensibility: sensibility is how much information a card can fetch. For example, the card “5” has more sensibility than other cards. If you know opponent drew a card from odd group, and you lost with 5, then the information you know is the card opponent drew could be 7 or 9. If won with 5, then the cards could be 1or 3. Therefore we say 5 have more sensibility.
* Winning rate: the possibility to beat opponent with the card you drew. Such as winning rate for 5 here is 2/5 if opponent draws a card from even group.
* Score: score is the product of the winning rate and sensibility. The decision machine makes to decide which card should be drew is up to the scores. The highest score has the priority to be drawn first.

From a technical view to our project, the followings are what we have used in our project.

* We have created a algorithm by our own, which is score = (winning rate) \* (sensibility)
* Searching method: BFS searching

The reason why we have used this searching method is that we can store the information that what cards opponent has left so that we can predict what card they might draw for the next rounds.

For example:

1. The AI drew card 5 on the first round, and AI lost while knowing that the opponent drew a card from odd group so that we can draw a tree based on this information:

Parent: [1,2,3,4,5,6,7,8,9]

5 lost / \

Child nodes: [1,2,3,4,5,6,7,8] [1,2,3,4,5,6,8,9] tmp: [7,9]

2. The AI drew a card 4 on the second round, and there is a tie so that we can conclude that the opponent also drew a card 4 on the second round. Therefore, the tree will look like:

Parent: [1,2,3,4,5,6,7,8,9]

5 lost / \

Child nodes: [1,2,3,4,5,6,7,8] [1,2,3,4,5,6,8,9] tmp: [7,9]

4 tie / \

[1,2,3,5,6,7,8] [1,2,3,5,6,8,9]

This is the way how we use tree to store information to guess what cards the opponent might still have on hands.