Texas Hold'em

Problem

Given the card combination in a poker game, i.e. list of hole cards for each player, flop, turn and river, your task is to implement an algorithm to determine which player is the winner.

Basic Rules

In Texas Hold'em, there are following hand categories ordered in descending ranking, straight flush, four of a kind, full house, flush, straight, three of a kind, two pair, one pair, no pair. In each game, each player has two hole cards (private to each player) and there are 5 cards (3 flop cards, 1 turn card and 1 river card) shared among all players on the table. Each player's final hand will be the highest ranking 5 cards selected out of the 7 cards (5 cards shared on the table plus the two cards the player has) available to the player. In Texax Hold'em, when comparing each card, only the rank of the card matters and the suit of the card doesn't. Therefore, tie is also possible.

Instruction

Please find the skeleton implementation in `main.rb` and implement the `determine_winner` instance method.

For instance, in a game of three players

```
game = Game.new(nil, nil, nil)
all_hole_cards = [
    [Card.new('Heart', 12), Card.new('Heart', 13)], # player1
    [Card.new('Spade', 12), Card.new('Spade', 13)], # player2
    [Card.new('Club',   2), Card.new('Club',  13)]] # player3
flop = [Card.new('Spade', 2), Card.new('Spade', 3), Card.new('Spade', 4)]
turn = Card.new('Spade', 5)
river = Card.new('Spde', 6)

game.determine_winner(all_hole_cards, flop, turn, river)
# should return 2 (the index of player3)
```

Please be aware that this problem is rather open ended and it will take much longer than one hour to implement all the combination checks like flush, straight, straight flush, full house, pairs, etc. Therefore, we are not expecting candidates to

implement all the different checks and having some combination checks working correctly is more important than implementing more wrong combination checks.

Consequently, we recommend to start implementing simpler checks like pairs, make sure they are working properly and then adding more checks if there is more time.

Your solution will be evaluated in the descending priority of correctness, memory & time complexity, and comprehensibility and in this problem, don't worry about handling the corner case and the input will always be a legit card combination.

Submission

Upon completion, please follow the instruction described in the website (where you found the instruction to download the project) to submit your solution. You can submit as many times as you like and your last submission will be used for the final evaluation as well as marking the end of your interview.

Lastly, do not worry about submitting many times nor running a little bit over time as they will not be penalized.