

Teorija iger: Texas hold'em poker

Žiga Štosir

Junij 2022


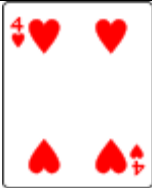






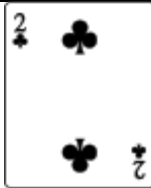




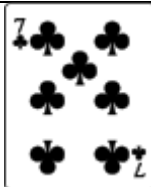




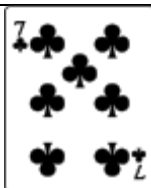




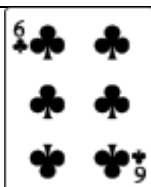


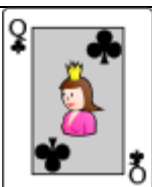
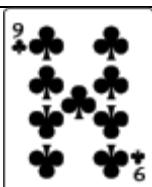
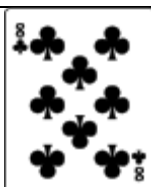
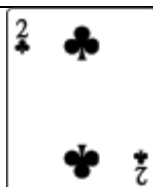



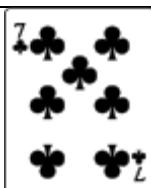




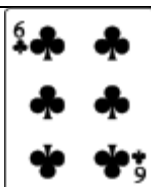

1. Uvod

Poker je ena izmed najpopularnejših iger s kartami, ki vključuje stave in pri kateri igra vsak igralec individualno. Zmagovalec je določen na podlagi kombinacije igralčevih kart, kjer nekatere ostanejo skrite do konca igre.

Obstajajo različne verzije pokra, ki se razlikujejo v skupnem številu kart, številu razdeljenih kart ter številu kart na mizi, ki ostanejo skrite. Igre pa se razlikujejo tudi v limitu stav oziroma maksimalni možni stavi. Pri večini poker različic se prvi krog stav prične tako, da eden ali več izmed igralcev vloži začetno stavo, nato se akcije igralcev vrstijo v smeri urnega kazalca, pri čemer igralec izenači do sedaj najvišjo stavo, odstopi ali pa stavo poviša. V primeru odstopa igralec izgubi ves svoj predhodni vložek v tej igri.

V projektni nalogi sem se osredotočil na eno izmed najpopularnejših verzij pokra, imenovano Texas hold'em. Vsakemu igralcu se razdelita dve karti z licem navzdol, nato pa se razdeli še pet skupinskih kart z licem navzgor v treh fazah. Najprej se razdelijo tri skupinske karte (angl. "the flop"), nato jim sledi posamična krta (angl. "the turn") in na koncu še zadnja skupinska karta (angl. "the river"). Vsak igralec si želi narediti čim boljšo roko iz petih kart iz vseh možnih kombinacij sedmih kart, svojih dveh in petih skupinskih. Pred in po vsaki izmed teh faz lahko v od prej določenem zaporedju igralci stavijo ali pa odstopijo. Igralec, ki ima na koncu vseh

krogov stav najboljšo roko in ni odstopil, je zmagovalec te igre. Možne kombinacije rok in kako si sledijo po vrednosti, je prikazano v spodnji tabeli.

Najvišja karta	Brez kombinacije					
Par	Dve isti karti					
Dva para	Dva para istih kart					
Tris	Tri iste kart					
Lestvica	Pet kart, ki si sledijo v vrstnem redu					
Flush	Vseh pet kart iste barve					
Full house	Tri iste karte in par dveh istih kart					
Štiri iste	Štiri iste vrednosti kart					











Barvna lestvica	Pet kart, ki si zaporedno sledijo in so iste barve					
Royal Flush						

Tabela 1 Prikaz možnih vrednosti rok v naraščajočem si zaporedju.

2. Opis naloge

Cilj projektne naloge je bil izdelati simulacijo igre Texas hold'em pokra za več igralcev. Pri implementaciji sem imel veliko težav s samo strategijo igralcev, ki mi je na kunci ni uspelo implementirati. Zato sem se osredotočil na simulacijo iger pri katerih noben izmed igralcev ne odstopi in eksperimentalno določil kakšna je verjetnost, da posamezna začetna roka vsakega igralca na koncu tudi zmaga. Implementiral pa sem tudi program, ki izvede Monte Carlo simulacijo in primerja uspešnost dveh začetnih rok.

Za implementacijo sem uporabil programski jezik Python. Rešitev je implementirana po posameznih funkcijah (mešanje kart, deljenje kart, itd.), ki skupaj v pravilnem vrstnem redu simulirale potek igre.

3. Implementacija

3.1 Predstavitev kart

Vrednost kart in posamezne barve sem shranil v vsak svoj seznam, ki sem ju nato združil v igralni kupček in ga premešal. Predstavitev je v kodi izgledala takole:

```
suits = ['S', 'H', 'C', 'D']
cardranks = {'2':2, '3':3, '4':4, '5':5, '6':6, '7':7, '8':8,
             '9': 9, 'T': 10, 'J':11, 'Q':12, 'K': 13, 'A': 14}

def deck(shuffled=False):
    d = [''.join((rank, suit)) for suit in suits for rank in cardranks]
    if shuffled:
        random.shuffle(d)
    return d
```

1. Prikaz predstavitve kart v programu

V tej predstavitvi si lahko pogledamo dva primera končnih rok, ki sta samo seznama petih elementov. Prvi predstavlja "flush", sestavljen iz kart, ki so vsi iste barve, v tem primeru pikov. Drugi pa "full house", ki je sestavljen iz dveh asov in treh kraljev.

```
#Flush (sami piki(spades))
flush = ['AS', 'JS', '9S', '6S', '2S']

#Full House (dve asi in trije kralji)
full_house = ['AS', 'AD', 'KC', 'KD', 'KH']
```

2 Prikaz dveh končnih rok.

3.2 Primerjava rok

Najzahtevnejši del je bila implementacija primerjanja rok med sabo. Za ta namen sem v pythonu implementiral "class" imenovan "PokerHand", kar nam pride prav ko si kasneje želimo simulirati igro. Izvedeti si želimo kakšno vrednost predstavlja končna roka, ki jo imamo. Ko roko ponovno shranimo oziroma jo imamo predstavljeno kot "class", lahko določimo vrednost na nekaj izmed teh načinov:

- če so vse karte iste barve, imamo flush, to lahko predstavimo tako, da pogledamo dolžino seznama barv. Če je ta enaka ena, to pomeni da imamo v roki samo eno vrsto barve in tako tudi flush,

- Če imamo štiri iste vrednosti ali pa full house, imamo v roki največ dva različna ranga kart. Kar lahko predstavimo podobno, da je dolžina seznama rangov vseh kart enaka dva.
- Če imamo lestvico, to pomeni da imamo pet različnih kart, torej je dolžina seznama rangov kart enaka pet. Razlika med največjim in najmanjšim rangom pa je enaka štiri. Itd ...

```
def flush(self):
    return len(set(self._suits)) == 1

def straight(self):
    return len(set(self._ranks)) == 5 and \
           max(self._ranks) - min(self._ranks) == 4 or \
           (len(set(self._ranks)) == 5 and \
            max(self._ranks) == 14 and \
            sum(self._ranks) == 28)

def straight_flush(self):
    return self.flush() and self.straight()

def four_of_a_kind(self):
    return len(set(self._ranks)) == 2 and self.ordered[0][1] == 4
```

3 Opisani primeri v kodi

Tukaj pa si pomagamo še s funkcijo "ordered", ki nam vrne seznam parov tipa: (rang karte, število kart tega ranga), ki je urejen najprej po številu kart posameznega ranga nato pa še po rangi karte. To nam pomaga pri določanju vrednosti roka, pomaga pa tudi v primeru ko imamo dve roki z enako kombinacijo, na primer ko je v vsaki izmed rok po en par.

```
'AS', 'AD', 'KC', 'KD', 'KH' - #Ordered: = [(14, 2), (13, 3)]
```

4Primer kako izgleda "ordered" full house iz slike dva

Na koncu vsaki roki na podlagi tega določimo številsko vrednost, ki jo nato uporabimo za primerjavo rok med sabo.

3.3 Simulacija igre

Za simulacijo same igre, ki je sestavljena iz miza s poljubnim številom igralcev, potrebujemo samo še nekaj stvari. Začnemo z zmešanim kupčkom kart in vsakemu izmed igralcev razdelimo po dve karti, Nato pa razdelimo še pet skupinskih kart, flop, turn in pa river.

Med samo simulacijo pa lahko hranimo tudi, kateri izmed igralcev oziroma katera roka je v trenutnem stanju igre najboljša. Torej kateri igralec ima najboljšo roko po prvem deljenju kart, po flopu, itd.

3.3.1 Simulacija večjega števila iger

Ko sem simuliral večje število iger za različno število igralcev sem opazoval, kolikokrat bo nek tip izbrane roke po začetnem deljenju kart na koncu tudi uspešen. Opazoval sem roke, v katerih je le najvišja karta asa. Roke pri katerih sta obe začetni karti iste barve in pa roke, ki vsebujejo par po prvem deljenju. Za te tri primere bom predstavil simulacijo 5000 iger za dva tri in šest igralcev.

```
Simulating for 2 players
```

```
1062 of 5000 Games where Ace High during Deal is a winner,  
in which 3911 games had an ace high hand.
```

```
1419 of 5000 Games where Flush during Deal is a winner,  
in which 4386 games had a flush hand.
```

```
504 of 5000 Games where Pairs during Deal is a winner,  
in which 1937 games had a pair hand.
```

5 Simulacija za 2 igralca.

```
Simulating for 3 players
```

```
1036 of 5000 Games where Ace High during Deal is a winner,  
in which 3929 games had an ace high hand.
```

```
1524 of 5000 Games where Flush during Deal is a winner,  
in which 4440 games had a flush hand.
```

```
503 of 5000 Games where Pairs during Deal is a winner,  
in which 1919 games had a pair hand.
```

6 Simulacija za tri igralce.

```
Simulating for 6 players
```

```
1005 of 5000 Games where Ace High during Deal is a winner,  
in which 3872 games had an ace high hand.
```


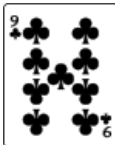


```
1453 of 5000 Games where Flush during Deal is a winner,  
in which 4412 games had a flush hand.
```

```
520 of 5000 Games where Pairs during Deal is a winner,  
in which 1867 games had a pair hand.
```

7 Simulacija za šest igralcev.

3.4 Primerjava uspešnosti posameznih začetnih rok

Implementiral pa sem tudi skripto, ki izvede simulacijo tipa Monte Carlo igre dveh rok, prebranih iz argumentov, ki jih podamo skripti. Simulacijo lahko izvedemo s poljubnima začetnima kartama in pa naključnimi skupinskimi kartami, lahko pa tudi kakšno izmed skupinskih kart določimo sami.

Primer simulacije tisoč iger za roki   in   s poljubnimi skupnimi kartami.

```
Total Hands: 1000  
Hand1: 630 Hand2: 367 Ties: 3  
Hand1: 63.00% Hand2: 36.70% Ties: 0.30%
```

8 Rezultat zgoraj opisane simulacije.

4. Literatura

- PLETIKOS, Nik, 2013, Enostavni modeli pokra in njihova analiza s teorijo iger : delo diplomskega seminarja [na spletu]. Univerza v Ljubljani, Fakulteta za matematiko in fiziko . [Dostopano 11 maj 2022]. Pridobljeno <https://repozitorij.uni-lj.si/IzpisGradiva.php?lang=slv&id=96946>
- Wikipedia. (2022). Texas hold'em. Pridobljeno s https://en.wikipedia.org/wiki/Texas_hold_%27em