Aufgabe 3: Tobis Turnier

Team-ID: 01011

Team: Jim Maar

Bearbeiter/-innen dieser Aufgabe:
Jim Maar

21. November 2020

Inhaltsverzeichnis

Lösungsidee

Um die Turniere simulieren zu können, müssen erst einmal einzelne Kämpfe zwischen 2 Spielern simuliert werden. Das wird gemacht, in dem eine zufällige Zahl zwischen 0 und 1 initiiert wird. Wenn diese kleiner ist als die Gewinnwahrscheinlichkeit vom ersten Spieler, gewinnt dieser. Wenn das nicht der Fall ist gewinnt der zweite Spieler. Die Gewinnwahrscheinlichkeit eines Spielers ist seine Spielstärke geteilt durch die Summe der Spielstärken beider Spieler. Die Gewinnwahrscheinlichkeiten von jedem möglichen Kampf werden am Anfang ausgerechnet und gespeichert. Eine Turniervariante wird eine hohe Anzahl an Malen simuliert. Dabei wird die Anzahl der Siege des spielstärksten Spielers mitgezählt. Am Ende wird diese durch die Anzahl der Simulationen geteilt und mal hundert genommen, um die Turniersiegwahrscheinlichkeit in Prozent zu erhalten. Da die Wahrscheinlichkeit das Ergebnis einer Simulation ist, wird sie gerundet. Die drei zu simulierenden Turniervarianten werden nun in einzelnen Abschnitten beschrieben.

Liga:
Angefangen mit dem spielstärksten Spieler kämpft jeder Spieler gegen jeden anderen Spieler. Dabei wird die Anzahl der Siege jedes Spielers gespeichert. Wenn ein Spieler mehr Siege hat als der spielstärkste Spieler oder genau gleich viele mit einer niedrigeren Nummer, hat der spielstärkste Spieler verloren, also er kann nicht mehr gewinnen. Passiert das nie, hat er gewonnen.

K.O.:
Die Spieler werden in einer Liste in eine zufällige Reihenfolge gesetzt und aufgereiht. Jeder zweite Spieler kämpft gegen den nächsten Spieler. Der Gewinner jedes Kampfes kommt in die nächste Runde. Mit diesen Spielern wird der Prozess wiederholt. Das wird solange gemacht, bis der

spielstärkste Spieler raus geflogen ist, also Verloren hat oder er der letzte übrig gebliebene Spieler ist, also gewonnen hat.

K.O. x5:
Wir gehen einen best of 5 Kampf zwischen einem Spieler 1 und einem Spieler 2 durch. P1 ist die Wahrscheinlichkeit, dass Spieler 1 gewinnt und P2, dass Spieler 2 gewinnt. Auf einem Wahrscheinlichkeitsraum, in dem alle Ergebnisse für die fünf Kämpfe stehen, sehen wir, dass Spieler 1 in einem Fall nach drei Spielen in drei Fällen nach vier Spielen und in sechs Fällen nach fünf Spielen gewinnt.Die Wahrscheinlichkeit dieser Fälle wird berechnet durch P1 hoch 3 multipliziert mit P2 hoch 0 bis 2. Die Wahrscheinlichkeit, dass Spieler 1 gewinnt, beträgt also P1^3 + 3*P1^3*P2 + 6*P1^3*P2^2. K.O. x5 kann genauso simuliert werden wie K.O. nur, dass die Gewinnwahrscheinlichkeiten mit dieser Formel berechnet werden.

Optimierungen:

Liga:
Die Spieler werden sortiert nach ihrer Spielstärke. So verliert der spielstärkste Spieler durchschnittlich früher. Dadurch, dass das Turnier beendet wird, sobald der Spielstärkste verliert, läuft das Turnier durchschnittlich schneller ab.

K.O./K.O. x5:
Anstatt ein komplettes Turnier komplett zu simulieren, wird die Gewinnwahrscheinlichkeit des spielstärksten Spielers berechnet. Die Berechnungen der Wahrscheinlichkeiten werden exponentiell mit der Anzahl der Spieler größer, weshalb die Turniere immer noch zum Teil simuliert werden.

Erst mal wird die ursprüngliche Funktion durchgeführt. Aber nur bis noch 8 Spieler im Rennen sind. Danach wird die Gewinnwahrscheinlichkeit für jeden Spieler berechnet. Das wird ähnlich gemacht wie bisher, nur verkleinert sich die Anzahl der Spieler nicht groß, dafür aber die Anzahl der Plätze. Einen Platz nennen wir die Plätze in einem Turnierbaum, in denen Spieler oder Mannschaften eingetragen werden. Ein Platz besteht aus den Spielern, die zu einer Wahrscheinlichkeit diesen Platz einnehmen können. Um die Anzahl der Plätze zu verkleinern und die Wahrscheinlichkeiten der Spieler auszurechnen, kämpfen alle Spieler eines Platzes jeweils gegen alle Spieler des Platzes rechts daneben.

Die Wahrscheinlichkeit eines Spielers auf dem Platz zu sein, auf dem er ist, wird momentane Wahrscheinlichkeit genannt. Diese beträgt zu Beginn 1. Die Wahrscheinlichkeit eines Spieler weiter zu kommen, also die nächste momentane Wahrscheinlichkeit beträgt, wenn der andere Platz nur einen Spieler enthält, das Produkt der Siegeswahrscheinlichkeit des Spielers und den momentanen Wahrscheinlichkeiten der beiden Spieler. Bei mehreren Spielern in dem anderen Platz beträgt die Wahrscheinlichkeit die Summe dieser Wahrscheinlichkeiten für jeden Spieler.

Alle Spieler aus zwei gegeneinander kämpfenden Plätzen kommen jetzt in einen Platz. Dadurch wird die Anzahl der Plätze halbiert. Eine Ausnahme bildet dabei der Platz, der den spielstärksten Spieler enthält. Die Wahrscheinlichkeiten der Spieler, gegen die der spielstärkste Spieler schon gekämpft hat, werden nicht benötigt. Deshalb kommt nur der spielstärkste Spieler in den nächsten Platz.

Dieser Vorgang wird solange wiederholt, bis nur noch der letzte Platz übrig ist. Dieser enthält den besten Spieler. Die Turniersiegwahrscheinlichkeit von diesem wird zurückgegeben. Der Vorgang startet erst bei 8 Spielern, da die Anzahl an Rechnungen exponentiell steigt und es mit 8 am schnellsten funktioniert. K.O. x5 kann immer noch genauso simuliert werden wie K.O..

Umsetzung

Die Lösungsidee wird in Python implementiert. Die Kämpfe werden mit einer random Funktion simuliert. Dafür muss die Bibliothek random importiert werden. Die Gewinnwahrscheinlichkeiten werden in einer zweidimensionalen Liste gespeichert. Diese enthält für jeden Spieler eine Liste mit den Siegeswahrscheinlichkeiten im Kampf gegen jeden anderen Spieler. K.O. und K.O. x5 benutzen beide die gleichen Funktionen. Die Gewinnwahrscheinlichkeiten werden bei K.O. x5 aber mit der anderen Formel berechnet.

Die Funktion “vorrechnen” erstellt die Gewinnwahrscheinlichkeiten. Dafür benutzt sie die Funktion “Wahrscheinlichkeitskampf”, die die Gewinnwahrscheinlichkeiten eines Kampfes zweier Spieler zurückgibt. Die Liga wird mit der Funktion “Liga” simuliert und K.O./K.O. x5 mit der Funktion ”KO”. Die K.O. Funktion benutzt 2 Funktionen “KORunde” und “KORunde2” um die Runden, in denen jeder Spieler einmal kämpft, zu simulieren. In “KORunde” kommt nur die Hälfte der Spieler in die nächste Runde, während in “KORunde2” jeder Spieler (der noch nicht gegen den spielstärksten Spieler gekämpft hat) zu einer gewissen Wahrscheinlichkeit in die nächste Runde kommt.

Das Moduls sys wird benutzt, damit das Programm mit der einzulesenden Datei oder einem Link einer in den Materialien zu findenden Beispiel Text Datei als Parameter auf der Kommandozeile aufgerufen werden kann.

Um die Links als Parameter benutzen zu können, wird das Modul urllib.request benutzt.

Beispiele

$ ./TobisTurnier.py https://bwinf.de/fileadmin/bundeswettbewerb/39/spielstaerken1.txt
Liga: 34.62%
K.O.: 40.35%
K.O. x5: 60.17%
$ ./TobisTurnier.py https://bwinf.de/fileadmin/bundeswettbewerb/39/spielstaerken2.txt
Liga: 20.95%
K.O.: 30.17%
K.O. x5: 35.91%
$ ./TobisTurnier.py https://bwinf.de/fileadmin/bundeswettbewerb/39/spielstaerken3.txt
Liga: 31.53%
K.O.: 16.69%
K.O. x5: 27.8%

$ ./TobisTurnier.py https://bwinf.de/fileadmin/bundeswettbewerb/39/spielstaerken4.txt
Liga: 11.45%
K.O.: 6.92%
K.O. X5: 7.54%

Empfehlung für Tobi

Ein Teil der Aufgabe besteht darin, Tobi eine Turniervariante zu empfehlen. Also Tobi bei der Liga hat der spielstärkste Spieler oft eine höhere Gewinnchance, als es bei K.O. und K.O. x5 der Fall wäre. Die Gewinnwahrscheinlichkeiten bei der Liga sind aber zu einem großen Teil von der Nummerierung abhängig.Es kann bei der Liga passieren, dass ein Spieler mit einer niedrigen Nummer eine sehr viel höhere Gewinnwahrscheinlichkeit hat als der spielstärkste Spieler. Da das bei K.O. und K.O. x5 nie der Fall ist, findest du diese wahrscheinlich besser. Bei K.O. x5 gewinnt der spielstärkste Spieler immer mit einer größeren Wahrscheinlichkeit als bei K.O.. Daher empfehle ich dir, dein Turnier nach der Turnierform K.O. x5 zu gestalten.

Quellcode

```
# Liga
def Liga():
    # Ligagewinne speichert die Gewinne jedes Spielers
    Ligagewinne = n * [0]
    for index, Spieler1 in enumerate(Spieler):
        for Spieler2 in Spieler[index + 1 :]:
            # Jeder Spieler kämpft gegen jeden anderen Spieler
            Ligagewinne[kampf(Spieler1, Spieler2)] += 1
        # Wenn ein Spieler mehr Siege hat als der beste Spieler
        # oder genau gleich viele mit einer niedrigeren Nummer hat der beste Spieler verloren
        # Es wird also eine 0 zurückgegeben
        if Spieler1 != besterSpieler and (
            Ligagewinne[Spieler1] > Ligagewinne[besterSpieler]
            or (
                Ligagewinne[Spieler1] == Ligagewinne[besterSpieler]
```

```
            and Spieler1 <= besterSpieler
        )
    ):
        return 0
    # Wenn das nie zugetroffen ist, hat er gewonnen. Es wird also eine 1 zurückgegeben
    return 1


# K.O.
def KORunde(Spieler):
    # nächsteSpieler beinhaltet die Spieler, die weiter kommen
    nächsteSpieler = []
    # Jeder Spieler kämpft gegen einen anderen Spieler, die Gewinner kommen in die Liste
    for i in range(0, len(Spieler), 2):
        Spieler1 = Spieler[i]
        Spieler2 = Spieler[i + 1]
        nächsteSpieler += [kampf(Spieler1, Spieler2)]
    # nächsteSpieler wird zurückgegeben
    return nächsteSpieler


def KORunde2(Plätze, Wahrscheinlichkeiten):
    # Wahrscheinlichkeitenneu beinhaltet die Wahrscheinlichkeit, jedes Spielers weiter zu kommen
    Wahrscheinlichkeitenneu = n * [0]
    # Plätzeneu beinhaltet die nächsten Plätze
    Plätzeneu = []
    # Jeder Spieler auf einem Platz kämpft gegen einen jeden Spieler aus einem anderen Platz
    for i in range(0, len(Plätze), 2):
        Platz1 = Plätze[i]
        Platz2 = Plätze[i + 1]
        # Alle Spieler aus den beiden Plätzen kommen dann in einen Platz
        # Wenn der beste Spieler kämpft, kommt nur er weiter
        if i == 0:
            Plätzeneu += [Platz1]
        else:
            Plätzeneu += [Platz1 + Platz2]
        for Spieler1 in Platz1:
            for Spieler2 in Platz2:
                # Basiswahrscheinlichkeit ist die Wahrscheinlichkeit,
                # dass die beiden Spieler gegeneinander kämpfen
                # Sie ist das Produkt der momentanen Wahrscheinlichkeiten, der beiden Spieler
                Basiswahrscheinlichkeit = (
                    Wahrscheinlichkeiten[Spieler1] * Wahrscheinlichkeiten[Spieler2]
                )
```

```
            # P1 und P2 sind Wahrscheinlichkeiten, dass der jeweilige Spieler gewinnt
            # Das Produkt der beiden ist die Wahrscheinlichkeit, dass der Spieler weiter kommt
            # Sie wird zu Wahrscheinlichkeitenneu hinzugefügt
            P1 = Gewinnwahrscheinlichkeiten[Spieler1][Spieler2]
            Wahrscheinlichkeitenneu[Spieler1] += Basiswahrscheinlichkeit * P1
            # Die Gewinnwahrscheinlichkeiten gegen den besten Spieler werden nicht gebraucht
            if i != 0:
                P2 = Gewinnwahrscheinlichkeiten[Spieler2][Spieler1]
                Wahrscheinlichkeitenneu[Spieler2] += Basiswahrscheinlichkeit * P2
    # Die neuen Plätze und Wahrscheinlichkeiten werden zurückgegeben
    return Plätzeneu, Wahrscheinlichkeitenneu


def KO():
    # KOSpieler beinhaltet alle Spieler, die noch im Spiel sind, in einer zufälligen Reihenfolge
    # Der beste Spieler ist aber immer das erste Element
    KOSpieler = list(range(0, n))
    KOSpieler.pop(besterSpieler)
    random.shuffle(KOSpieler)
    KOSpieler = [besterSpieler] + KOSpieler
    # Alle Runden werden durchgegangen, bis nur noch 8 Spieler im Spiel sind
    for _ in range(3, int(log(n, 2))):
        # Jeder Spieler kämpft ein Mal, die Verlierer fliegen raus
        KOSpieler = KORunde(KOSpieler)
        # Wenn der beste Spieler raus ist, hat er eine Turniersiegwahrscheinlichkeit von 0
        if KOSpieler[0] != besterSpieler:
            return 0
    # KOSpieler wird in Platzform gebracht
    KOPlätze = [[i] for i in KOSpieler]
    # Wahrscheinlichkeiten beinhaltet die Wahrscheinlichkeiten aller Spieler,
    # dass sie auf dem jetzigen Platz sind
    Wahrscheinlichkeiten = n * [1]
    # Die zweite KO Funktion wird durchgeführt, bis nur noch ein Platz übrig ist
    for _ in range(0, 3):
        KOPlätze, Wahrscheinlichkeiten = KORunde2(KOPlätze, Wahrscheinlichkeiten)
    # Die Turniersiegwahrscheinlichkeit des besten Spielers wird zurückgegeben
    return Wahrscheinlichkeiten[besterSpieler]

# Gewinnwahrscheinlichkeiten enthält die Gewinnwahrscheinlichkeiten
# von jedem Spieler gegen jeden anderen Spieler
Gewinnwahrscheinlichkeiten = vorrechnen(False)
# Gewinne beihaltet die Anzahl an Turniersiegen von dem besten Spieler
Gewinne = 0
# Das Turnier wird viele Male simuliert
# Liga gibt 1 für einen Sieg und 0 für eine Niederlage des besten Spielers zurück
for _ in range(0, durchgänge):
    Gewinne += Liga()
```

```
# Die Gewinne werden durch die Anzahl der Durchgänge geteilt und mal 100
# genommen Turniergewinnwahrscheinlichkeit in Prozent zu erhalten
# Da es sich um eine Simulation handelt, wird die Wahrscheinlichkeit gerundet
print(f"Liga: {round(Gewinne / durchgänge * 100, 2)}%")

# Mit KO und KO5 wird das Gleiche gemacht
# Bei KO/KO5 bekommt der beste Spieler eine Turniergewinnwahrscheinlichkeit
# Diese wird in Gewinne eingetragen
Gewinne = 0
for _ in range(0, durchgänge):
    Gewinne += KO()
print(f"K.O.: {round(Gewinne / durchgänge * 100, 2)}%")

# Bei KO5 werden die Gewinnwahrscheinlichkeiten mit der Formel berechnet
Gewinnwahrscheinlichkeiten = vorrechnen(True)
Gewinne = 0
for _ in range(0, durchgänge):
    Gewinne += KO()
print(f"K.O. x5: {round(Gewinne / durchgänge * 100, 2)}%")
```