

# Aufgabe 5: Wichteln

Teilnahme-Id: 56043

Bearbeiter/-in dieser Aufgabe:  
Sebastian Brunnert

21. September 2020

## Inhaltsverzeichnis

Lösungsidee.....	1
Umsetzung.....	1
Beispiele.....	2
Quellcode.....	3

## Lösungsidee

Wenn man Erst-Wünsche oder Zweit-Wünsche bei dieser Problematik einfach an Schüler ausgibt, ohne zu beachten, ob es Schüler gibt, denen besser diese Geschenke gegeben werden sollten, da dadurch dann z.B. mehr Zweit-Wünsche beachtet werden können, erhält man je nach Reihenfolge der Schüler oder der generellen Wunsch-Verteilung meistens nie eine perfekte Verteilung (außer z.B. bei Beispiel 2). Daher muss entweder algorithmisch herausgefunden werden, welchem Schüler man am besten ein Geschenk gibt, welches sich auch andere Schüler an selber Stelle wünschten. Eine andere Methode ist es allerdings auch, dass die Liste der Schüler immer zufällig gemischt wird und dann über viele Durchläufe die beste Verteilung herausgefunden wird. Zwar besteht immer noch ein Restrisiko, dass es sich bei dieser Verteilung nicht um die beste Verteilung handelt, doch bei ausreichend hoher Wiederholungs-Rate des letztlich zufälligen Verteilungs-Ablaufes, ist dieses Restrisiko stochastisch gesehen vernachlässigbar. Dementsprechend habe ich mich dafür entschieden, die Geschenke letztlich simpel an potenzielle Abnehmer zu verteilen, also an Schüler, die sich dieses Geschenk wünschen. Dieser Ablauf wird häufig wiederholt und die beste Verteilung wird dadurch ermittelt.

## Umsetzung

Die Umsetzung dieser Aufgabe geschieht in der Programmiersprache Python (3). Zu Beginn des Programmes werden die Wünsche der Schüler eingelesen und eine Vorlage-Schema wird erstellt: Ein Array mit Objekten, in denen zum Einen ein drei Elemente großer Array gespeichert wird mit Integern, welche die Wünsche wiedergeben, sowie zum Anderen ein Integer, der das Geschenk widerspiegelt, welches dem Schüler zugeteilt wurde. Dieses Schema soll im nachfolgenden Programm-Verlauf oftmals kopiert werden, sodass die Datei nicht immer wieder erneut eingelesen werden muss. Außerdem wurde eine Hilfsmethode aufgestellt, mit welcher geprüft werden kann, ob innerhalb eines Verteilungs-Schemas (also eine bearbeitete Kopie der Vorlage) ein Geschenk bereits an einen Schüler verteilt wurde. Nun wird eine über eine Hohe Anzahl an Wiederholungen folgende letztlich simple Logik wiederholt: Die Vorlage wird kopiert und zufällig gemischt mittels `random.shuffle()`. Anschließend wird durch alle Schüler iteriert, und wenn der Erst-Wunsch des Schülers nicht bereits an einen anderen Schüler verteilt wurde, wird diesem Schüler sein Erst-Wunsch zugeteilt. Selbiges wird mit dem Zweit- und dem Drittwunsch durchgeführt. Nachdem nun (möglichst viele) Wünsche erfüllt wurden, wird gemessen, wie oft welche Wünsche erfüllt wurden. Es werden zumindest zum jetzigen Zeitpunkt des

Programmes Schüler vernachlässigt, welchen noch kein Geschenk zugeteilt wurde, um das Programm effizienter zu gestalten. Wenn es sich bei dieser Verteilung um die zumindest aktuelle „beste“ Verteilung handelt, wird diese zwischengespeichert, um sie später mit anderen Verteilungs-Schemen zu vergleichen bzw. am Ende des Programmes auszugeben. Wurde dieses Logik nun oftmals ausgeführt, wurde mit ziemlich hoher Wahrscheinlichkeit die beste Verteilung gefunden und diese kann ausgegeben werden. Bei dieser Ausgabe werden auch Geschenke für die Schüler ermittelt, dessen Wünsche nicht berücksichtigt werden konnten.

## Beispiele

wichteln1.txt	<p>Beste Verteilung:</p> <p>Schüler 1 wurde Geschenk 7 zugeteilt.          Schüler 2 wurde Geschenk 2 zugeteilt.          Schüler 3 wurde Geschenk 6 zugeteilt.          Schüler 4 wurde Geschenk 4 zugeteilt.          Schüler 5 wurde Geschenk 10 zugeteilt.          Schüler 6 wurde Geschenk 8 zugeteilt.          Schüler 7 wurde Geschenk 3 zugeteilt.          Schüler 8 wurde Geschenk 9 zugeteilt.          Schüler 9 wurde Geschenk 1 zugeteilt.          Schüler 10 wurde Geschenk 5 zugeteilt.</p> <p>Erfüllte Erst-Wünsche: 6          Erfüllte Zweit-Wünsche: 0          Erfüllte Dritt-Wünsche: 2</p>
wichteln2.txt	<p>Beste Verteilung:</p> <p>Schüler 1 wurde Geschenk 5 zugeteilt.          Schüler 2 wurde Geschenk 6 zugeteilt.          Schüler 3 wurde Geschenk 10 zugeteilt.          Schüler 4 wurde Geschenk 4 zugeteilt.          Schüler 5 wurde Geschenk 9 zugeteilt.          Schüler 6 wurde Geschenk 8 zugeteilt.          Schüler 7 wurde Geschenk 7 zugeteilt.          Schüler 8 wurde Geschenk 3 zugeteilt.          Schüler 9 wurde Geschenk 2 zugeteilt.          Schüler 10 wurde Geschenk 1 zugeteilt.</p> <p>Erfüllte Erst-Wünsche: 3          Erfüllte Zweit-Wünsche: 0          Erfüllte Dritt-Wünsche: 0</p>
wichteln3.txt	<p>Beste Verteilung:</p> <p>Schüler 1 wurde Geschenk 3 zugeteilt.          Schüler 2 wurde Geschenk 26 zugeteilt.          Schüler 3 wurde Geschenk 12 zugeteilt.          Schüler 4 wurde Geschenk 15 zugeteilt.          [...]         Schüler 29 wurde Geschenk 17 zugeteilt.          Schüler 30 wurde Geschenk 1 zugeteilt.</p> <p>Erfüllte Erst-Wünsche: 15          Erfüllte Zweit-Wünsche: 6          Erfüllte Dritt-Wünsche: 1</p>
wichteln4.txt	<p>Beste Verteilung:</p> <p>Schüler 1 wurde Geschenk 19 zugeteilt.          Schüler 2 wurde Geschenk 27 zugeteilt.          Schüler 3 wurde Geschenk 28 zugeteilt.          Schüler 4 wurde Geschenk 6 zugeteilt.          [...]         Schüler 29 wurde Geschenk 4 zugeteilt.          Schüler 30 wurde Geschenk 29 zugeteilt.</p>

	Erfüllte Erst-Wünsche: 15 Erfüllte Zweit-Wünsche: 4 Erfüllte Dritt-Wünsche: 3
wichteln5.txt	Beste Verteilung: Schüler 1 wurde Geschenk 6 zugeteilt. Schüler 2 wurde Geschenk 10 zugeteilt. Schüler 3 wurde Geschenk 7 zugeteilt. Schüler 4 wurde Geschenk 4 zugeteilt. [...]         Schüler 29 wurde Geschenk 27 zugeteilt. Schüler 30 wurde Geschenk 5 zugeteilt. Erfüllte Erst-Wünsche: 13 Erfüllte Zweit-Wünsche: 1 Erfüllte Dritt-Wünsche: 7
wichteln6.txt	Beste Verteilung: Schüler 1 wurde Geschenk 35 zugeteilt. Schüler 2 wurde Geschenk 39 zugeteilt. Schüler 3 wurde Geschenk 12 zugeteilt. Schüler 4 wurde Geschenk 48 zugeteilt. [...]         Schüler 89 wurde Geschenk 46 zugeteilt. Schüler 90 wurde Geschenk 13 zugeteilt. Erfüllte Erst-Wünsche: 37 Erfüllte Zweit-Wünsche: 3 Erfüllte Dritt-Wünsche: 21
wichteln7.txt (20.000 Wiederholungen)	Schüler 1 wurde Geschenk 631 zugeteilt. Schüler 2 wurde Geschenk 941 zugeteilt. Schüler 3 wurde Geschenk 841 zugeteilt. Schüler 4 wurde Geschenk 150 zugeteilt [...]         Schüler 998 wurde Geschenk 319 zugeteilt. Schüler 999 wurde Geschenk 1 zugeteilt. Schüler 1000 wurde Geschenk 310 zugeteilt.. Erfüllte Erst-Wünsche: 541 Erfüllte Zweit-Wünsche: 108 Erfüllte Dritt-Wünsche: 46

## Quellcode

*Der vollständige Quellcode, ist in der Datei wichteln.py zu finden.*

# Eine Vorlage wird erstellt, in der alle Schüler gespeichert werden, samt Geschenk, welches zugeteilt wurde und einem Array mit den Wünschen des Schülers

```
schuelerArrVorlage = []
```

```
for i in range(1,len(content)):
```

```
    schuelerArrVorlage.append({
```

```
        "zugeteilt": None,
```

```
        "wuensche": [int(s) for s in content[i].split() if s.isdigit()]
```

```
}}
```

```
# Hilfsmethode um zu prüfen, ob innerhalb einer möglichen Verteilung ein Geschenk schon verteilt wurde
```

```
# schuelerArr ist das Verteilungs-Schema
```

```
def istVerteilt(schuelerArr,geschenkId):
```

```
    for schueler in schuelerArr:
```

```
        if(schueler["zugeteilt"] == geschenkId):
```

```
            return True
```

```
    return False
```

```
# In diesen beiden Variablen werden die (jedenfalls bisher) besten Verteilungen zwischengespeichert
```

```
besteVerteilung = None
```

```
besteErfuellteWuensche = None
```

```
print("Berechnen...")
```

```
# Nun wird 1000 mal eine mögliche Verteilung geniert
```

```
# Jedesmal wird geschaut, wie "gut" diese Verteilung ist
```

```
# Wenn sie die (jedenfalls bisher) beste ist, wird sie gespeichert
```

```
for z in range(0,1000):
```

```
    print(str(z) + "/1000")
```

```
# Die Vorlage wird kopiert und zufällig gemischt
```

```
schuelerArr = copy.deepcopy(schuelerArrVorlage)
```

```
random.shuffle(schuelerArr)
```

```
# Die Erst-Wünsche von gewissen Schülern werden erfüllt.
```

```
# Dabei wird nicht berücksichtigt, welchen Einfluss die Verteilung auf die Verteilung von Zweit-Wünschen  
hat
```

```
for schueler in schuelerArr:
```

```
    if(not istVerteilt(schuelerArr,schueler["wuensche"][0])):
```

```
        schueler["zugeteilt"] = schueler["wuensche"][0]
```

```
# Zweitwünsche werden verteilt

for schueler in schuelerArr:
    if(schueler["zugeteilt"] is None):
        if(not istVerteilt(schuelerArr,schueler["wuensche"][1])):
            schueler["zugeteilt"] = schueler["wuensche"][1]

# Drittwünsche werden verteilt

for schueler in schuelerArr:
    if(schueler["zugeteilt"] is None):
        if(not istVerteilt(schuelerArr,schueler["wuensche"][2])):
            schueler["zugeteilt"] = schueler["wuensche"][2]

# Es wird gemessen, wie oft Wünsche erfüllt wurden
erfuellteWuensche = [0,0,0]

for schueler in schuelerArr:
    for wunschId in range(0,3):
        if(schueler["wuensche"][wunschId] == schueler["zugeteilt"]):
            erfuellteWuensche[wunschId] += 1

# Es wird geprüft, ob diese Verteilung "besser" war als ein vorige bzw. die erste ist
# Wenn ja wird diese gespeichert
if(besteErfuellteWuensche == None):
    besteErfuellteWuensche = erfuellteWuensche
    besteVerteilung = schuelerArr
else:
    if((erfuellteWuensche[1] > besteErfuellteWuensche[1]) or (erfuellteWuensche[1] ==
besteErfuellteWuensche[1] and erfuellteWuensche[2] > besteErfuellteWuensche[2]]):
        besteErfuellteWuensche = erfuellteWuensche
        besteVerteilung = schuelerArr
```