

Aufgabe 5: Wichteln

Team-ID: 01011

Team: Jim Maar

Bearbeiter/-innen dieser Aufgabe:
Jim Maar

22. November 2020

Inhaltsverzeichnis

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

Lösungsidee

Ein Gegenstand, den eine Person als Erstwunsch hat, hat einen Wunschgrad von 1. Entsprechend haben Zweitwünsche den Wunschgrad 2 und Drittwünsche den Wunschgrad 3. Ein Gegenstand kann auch keinen oder mehrere Wunschgrade haben. Ein Gegenstand, den eine Person erhält, die diesen als Erstwunsch hat, hat einen Erfüllungsgrad von 1. Das Gleiche gilt wieder für Zweit- und Drittwünsche mit den entsprechenden Erfüllungsgraden 2 und 3. Ein Gegenstand kann entweder einen oder keinen Erfüllungsgrad haben. Ein Wunsch besteht aus einem Gegenstand mit einem Wunschgrad.

Ein Beispiel: Gegenstand 1 hat die Wunschgrade 2 und 3. Spieler 4 wünscht sich Gegenstand 1 mit Wunschgrad 2 und Spieler 5 wünscht sich Gegenstand 1 mit Wunschgrad 3. Spieler 4 erhält Gegenstand 1. Dadurch hat Gegenstand 1 den Erfüllungsgrad 2.

Jeder Gegenstand mit Wunschgrad 1 muss auch den Erfüllungsgrad 1 haben, damit die Anzahl an erfüllten Erstwünschen maximal ist. Alle weiteren Wünsche von so einem Gegenstand mit Wunschgrad 2 oder 3 werden gelöscht. Wenn sich eine Person nur einen Gegenstand mit einem Wunschgrad w wünscht, weil die anderen beiden Wünsche gelöscht worden sind, werden alle anderen Wünsche von demselben Gegenstand mit einem Wunschgrad $\geq w$ gelöscht. Sie können gelöscht werden, da wenn sie anstatt dem Wunsch der Person erfüllt würden, die Verteilung nicht besser wäre. Wenn diese Löschungen fertig sind, kann es sein, dass die Anzahl an Wünschen stark gesunken ist.

Die Anzahl an Personen, die sich einen Gegenstand mit einem bestimmten Wunschgrad wünschen, wird Personenzahl genannt. Im folgenden Abschnitt wird mit Personenzahl immer die Personenzahl mit Wunschgrad 1 gemeint.

Nun wird die beste Verteilung durch Brute-Force, also durch die Erstellung und Bewertung aller möglichen sinnvollen Verteilungen gefunden. Alle Personen, die sich einen Gegenstand mit einer Personenzahl von 1 wünschen, können diesen erhalten. Gegenstände können nur von genau einer Person erhalten werden. Für Gegenstände mit einer Personenzahl > 1 werden weitere Verteilungen erstellt. Die weiteren Verteilungen enthalten alle möglichen Kombinationen von erfüllten Erstwünschen.

Die weiteren Verteilungen werden folgendermaßen erstellt: Die mehrmals gewünschten Gegenstände werden durchnummeriert. Es gibt zu Beginn eine Verteilung. Für Gegenstand 1 wird diese Verteilung Personenzahl -1 mal kopiert. In jeder Verteilung erhält eine andere Person den Gegenstand. Für Gegenstand 2 wird das Gleiche mit jeder bereits vorhandenen Verteilung gemacht. Dieser Vorgang läuft für jeden weiteren mehrmals gewünschten Gegenstand ab. Die anderen Wünsche von Personen die, einen Gegenstand erhalten haben werden gelöscht.

In den neuen Verteilungen kann nun jeder Gegenstand mit Wunschgrad 2 den Erfüllungsgrad 2 haben. Diese Gegenstände mit Wunschgrad 3 werden wieder gelöscht.

Die Verteilungen können sich in der Menge an erfüllbaren Zweitwünschen unterscheiden. Nur die mit der höchsten Anzahl an erfüllbaren Zweitwünschen können die beste Verteilung sein. Mit denen wird derselbe Prozess wiederholt, diesmal unterscheiden sich die weiteren Verteilungen aber in den erfüllten Zweitwünschen. Dafür werden die Personenzahlen mit Wunschgrad 2 genutzt. Jetzt haben wir die Verteilungen mit den meisten erfüllten Erst- und Zweitwünschen. Die Verteilung, die jetzt die meisten erfüllbaren Drittwünsche enthält, ist die beste Verteilung.

Damit die Verteilung Vollständig ist erhalten die Personen noch keinen Gegenstand erhalten haben die Gegenstände ohne Erfüllungsgrad. Dies hat keinen Einfluss auf die Bewertung der Verteilung.

Umsetzung

Die Lösungsidee wird in Python implementiert. Die Wünsche werden auf 2 verschiedene Arten gespeichert. Einmal werden die Wünsche jeder Person in einer Liste gespeichert und einmal werden alle Personen, die sich einen Gegenstand mit einem bestimmten Wunschgrad wünschen, für jeden Gegenstand in einem Dictionary gespeichert. Für jeden Wunschgrad gibt es ein Dictionary. Diese sind dann noch mal alle in einer Liste. Die Liste "restGegenstände" speichert alle Gegenstände ohne Erfüllungsgrad.

Der Algorithmus besteht größtenteils aus 2 Funktionen. Die erste "löschen" löscht alle Wünsche, die wie oben beschrieben nutzlos sind. Die zweite Funktion "Verteilungenerstellung" erstellt alle Verteilungen mit unterschiedlichen erfüllten Wünschen eines Wunschgrades.

Das Modul 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

Wir rufen das Programm mit verschiedenen bwinf-Eingabedaten auf. Diese werden aus den in den Materialien zu findenden URLs importiert. Das Lösung des siebten Beispiel wird aus Platzgründen hier nicht gezeigt.

```
$ ./Wichteln.py https://bwinf.de/fileadmin/bundeswettbewerb/39/wichteln1.txt
[6, 2, 1, 3, 5, 8, 7, 10, 9, 4]
$ ./Wichteln.py https://bwinf.de/fileadmin/bundeswettbewerb/39/wichteln2.txt
[4, 5, 6, 1, 2, 3, 7, 8, 9, 10]
$ ./Wichteln.py https://bwinf.de/fileadmin/bundeswettbewerb/39/wichteln3.txt
[2, 20, 29, 8, 1, 3, 5, 12, 4, 28, 6, 9, 14, 23, 26, 30, 11, 7, 16, 10, 19, 13, 27, 15, 17, 18, 22, 21, 24, 25]
$ ./Wichteln.py https://bwinf.de/fileadmin/bundeswettbewerb/39/wichteln4.txt
[28, 21, 22, 14, 6, 5, 3, 16, 4, 9, 10, 11, 7, 20, 2, 23, 1, 27, 17, 13, 19, 25, 15, 12, 30, 26, 18, 8, 24, 29]
$ ./Wichteln.py https://bwinf.de/fileadmin/bundeswettbewerb/39/wichteln5.txt
[4, 6, 7, 18, 15, 27, 5, 9, 16, 20, 14, 13, 19, 21, 10, 8, 26, 2, 22, 3, 1, 24, 23, 11, 28, 29, 30, 12, 17, 25]
$ ./Wichteln.py https://bwinf.de/fileadmin/bundeswettbewerb/39/wichteln6.txt
[1, 27, 53, 2, 30, 21, 37, 45, 55, 51, 9, 33, 86, 26, 4, 42, 28, 39, 15, 35, 6, 23, 7, 19, 84, 34, 12, 36, 78, 25, 54, 43, 44, 50, 74, 14, 52, 48, 56, 31, 32, 83, 29, 22, 5, 85, 38, 57, 16, 59, 40, 65, 60, 61, 62, 64, 87, 46, 13, 81, 66, 75, 68, 67, 10, 80, 69, 47, 70, 71, 18, 90, 3, 72, 24, 73, 58, 76, 17, 77, 8, 63, 11, 49, 79, 41, 20, 82, 88, 89]
```

Quellcode

```
def löschen(Wünsche, Person):
    # Gegenstand ist der Gegenstand
    Gegenstand = sum(Wünsche)
    # Wunschgrad ist der Wunschgrad von dem Gegenstand
    Wunschgrad = Wünsche.index(Gegenstand)
    # Indizes beinhaltet alle Personen, die sich den Gegenstand wünschen
    Indizes = [Personen[Wunschgrad][Gegenstand]]
    # Personen wird aktualisiert, der Wunsch wird nur von einer Person gewünscht
    Personen[Wunschgrad][Gegenstand] = [Person]
    if Wunschgrad == 1 and Gegenstand in Personen[2]:
        # Bei einem Wunschgrad von 2 werden auch Wünsche mit Wunschgrad 3 gelöscht
        Indizes += [Personen[2][Gegenstand]]
        del Personen[2][Gegenstand]
    for index in range(0, len(Indizes)):
        for Person2 in Indizes[index]:
            Wünsche2 = Wünscheliste[Person2]
```

```

if Person != Person2:
    # Der Wunsch wird von anderen Personen entfernt
    Wünsche2[Wunschgrad + index] = 0
    # Die Funktion wird noch mal aufgerufen,
    # wenn die Person nur noch einen Wunsch hat
    if Wünsche2.count(0) == 2:
        löschen(Wünsche2, Person2)

```

```

def indextoWunsch(index, Wunschgrad):
    # Wunschgrad ist der Wunschgrad, für den grade Varianten erstellt werden
    # Wünsche sind die Wünsche einer Person
    Wünsche = Wünscheliste[index]
    # Alle Wünsche mit einem höheren Wunschgrad werden zurückgegeben
    if Wünsche[Wunschgrad + 1] != 0:
        yield [Wunschgrad + 1, Wünsche[Wunschgrad + 1]]
    if Wunschgrad == 0 and Wünsche[Wunschgrad + 2] != 0:
        yield [Wunschgrad + 2, Wünsche[Wunschgrad + 2]]

```

```

def Verteilungenerstellung(locPersonen, Wunschgrad):
    # locPersonen2 beinhaltet alle gewünschten Gegenstände und die Personen
    locPersonen2 = {k: v for (k, v) in locPersonen[Wunschgrad].items() if len(v) >= 1}
    # Personensammlung wird alle Verteilungen beinhalten
    Personensammlung = []
    if len(locPersonen2) > 0:
        # Gegenstandliste beinhaltet alle gewünschten Gegenstände als Liste
        Gegenstandliste = [k for (k, v) in locPersonen2.items()]

```

```

def Verteilungenerstellung2(Gegenstandindex, locPersonen):
    nonlocal Personensammlung
    # Gegenstand ist ein gewünschter Gegenstand
    Gegenstand = Gegenstandliste[Gegenstandindex]
    # Indizes sind Personen, die sich diesen Gegenstand Wünschen
    Indizes = locPersonen2[Gegenstand]
    for index in Indizes:
        # Für jede dieser Personen wird die Verteilung kopiert
        locPersonen3 = [
            {k: v.copy() for (k, v) in i.items()} for i in locPersonen
        ]
        # Der Gegenstand wird in dieser Verteilung nur von der Person gewünscht
        locPersonen3[Wunschgrad][Gegenstand] = [index]
        # Die anderen Wünsche der Person werden aus der Verteilung gelöscht
        for Wunschgrad2, Gegenstand2 in indextoWunsch(index, Wunschgrad):
            locPersonen3[Wunschgrad2][Gegenstand2].remove(index)
            if len(locPersonen3[Wunschgrad2][Gegenstand2]) == 0:
                del locPersonen3[Wunschgrad2][Gegenstand2]

```

```

    if Gegenstandindex + 1 < len(Gegenstandliste):
        # Aus dieser Verteilung werden weitere Verteilungen erstellt,
        # in denen andere Gegenstände erhalten werden
        Verteilungenerstellung2(Gegenstandindex + 1, locPersonen3)
    else:
        # Wenn es keine weiteren Gegenstände mehr gibt,
        # wird die Verteilung zu Personensammlung hinzugefügt
        Personensammlung += [locPersonen3]

    # Verteilungenerstellung2 erstellt alle weiteren Verteilung
    Verteilungenerstellung2(0, locPersonen)
else:
    # Wenn keine weiteren Verteilungen erstellt werden,
    # wird die eingegebene Verteilung zurückgegeben
    Personensammlung += [locPersonen]
return Personensammlung

```

```
Erstwünsche = [i[0] for i in Wünscheliste]
```

```
for Wunschgrad in range(1, 3):
```

```
    for Person, Wünsche in enumerate(Wünscheliste):
```

```
        # Wünsche mit Wunschgrad 2 und 3 von Gegenständen mit Wunschgrad 1 werden gelöscht
```

```
        if Wünsche[Wunschgrad] in Erstwünsche:
```

```
            Wünscheliste[Person][Wunschgrad] = 0
```

```
# Personen enthält zu jedem Gegenstand die Personen, die ihn Wünschen sortiert nach Wunschgrad
```

```
Personen = [{}, {}, {}]
```

```
for Wunschgrad in range(0, 3):
```

```
    for Person, Wünsche in enumerate(Wünscheliste):
```

```
        # Alle Personen, die nicht gelöscht wurden, werden in Personen gespeichert
```

```
        if Wünsche[Wunschgrad] != 0:
```

```
            if Wünsche[Wunschgrad] not in Personen[Wunschgrad]:
```

```
                Personen[Wunschgrad][Wünsche[Wunschgrad]] = [Person]
```

```
            else:
```

```
                Personen[Wunschgrad][Wünsche[Wunschgrad]] += [Person]
```

```
# Für jede Person, die einen einzigen Gegenstand wünscht, wird die Löschfunktion aktiviert
```

```
for Person, Wünsche in enumerate(Wünscheliste):
```

```
    if Wünsche.count(0) == 2:
```

```
        # Die Löschfunktion löscht Wünsche von Gegenständen die unnötig sind,
```

```
        # weil eine Person eh nur diesen Gegenstand erhalten kann
```

```
        löschen(Wünsche, Person)
```

```
# Personensammlung beinhaltet alle möglichen Variationen von erfüllten ErstWünschen
```

```
Personensammlung = Verteilungenerstellung(Personen, 0)
```

```
for Personen in Personensammlung:
```

```
# Gegenstände mit Wunschgrad 3 werden aus einer Verteilung rausgenommen,
# wenn sie mit Wunschgrad 2 erfüllt werden können
Personen[2] = {k: v for (k, v) in Personen[2].items() if k not in Personen[1]}

# Personensammlung2 enthält alle möglichen sinnvollen Variationen von erfüllten ZweitWünschen
Personensammlung2 = []
# maxlen ist die größte Anzahl an erfüllbaren ZweitWünschen,
# bei Erfüllung der größtmöglichen Anzahl an ErstWünschen
maxlen = max(len(i[1]) for i in Personensammlung)
for Personen in Personensammlung:
    if len(Personen[1]) == maxlen:
        # Wenn die Anzahl an erfüllbaren ZweitWünschen einer Verteilung gleich maxlen ist,
        # werden alle Variationen von erfüllten ZweitWünschen zu dieser Verteilung erstellt
        Personensammlung2 += Verteilungenerstellung(Personen, 1)

# restGegenstände enthält Gegenstände, die niemand erhalten hat
# Sie beeinflussen nicht die Bewertung
restGegenstände = list(range(1, n + 1))
# maxlen ist die größte Anzahl an erfüllbaren DrittWünschen,
# bei Erfüllung der größtmöglichen Anzahl an Erst- und ZweitWünschen
maxlen = max(len(i[2]) for i in Personensammlung2)
for Personen in Personensammlung2:
    # Die Verteilung mit den meisten erfüllbaren DrittWünschen,
    # wird in Listenform gebracht
    if len(Personen[2]) == maxlen:
        for Wunschgrad in Personen:
            for Gegenstand in Wunschgrad:
                verteilung[Wunschgrad[Gegenstand][0]] = Gegenstand
                # Erfüllte Wünsche werden aus restGegenstände entfernt
                restGegenstände[Gegenstand - 1] = 0
            break
# Die restGegenstände werden in das Endergebnis eingesetzt
restGegenstände = [i for i in restGegenstände if i != 0]
verteilung = [i if i != 0 else restGegenstände.pop(0) for i in verteilung]
# Das Ergebnis wird ausgegeben
print(verteilung)
```