42. BWInf - Junioraufgabe 1

Inhaltsverzeichnis

Lösungsidee

Die Anzahl der Geschenke pro Tüte darf sich nur um 1 unterscheiden (Anforderung *A*). Dasselbe gilt für jeden Geschenktyp (Anforderung *B*). Eine Möglichkeit, das Ziel zu erreichen, ist folgendes:

1. Für jeden Geschenktyp werden die Geschenke nacheinander verteilt.
2. Ein Tütenindex zählt mit, welche Tüte als nächstes ein Geschenk bekommt. Bei jedem Geschenk wird der Index inkrementiert und gegebenenfalls auf 0 zurückgesetzt (nach der letzten Tüte).
3. Beim Wechsel zum nächsten Geschenktyp wird der Index beibehalten.

Um nicht nach jeder Tüte überprüfen zu müssen, ob sie die letzte war, bietet es sich an, Modulo zu verwenden. Hierzu ein Beispiel:

- Es gibt 3 Tüten.
- Gerade wurde ein Geschenk in die dritte Tüte gelegt
  - => alle 3 Tüten haben gleichviel Inhalt
- Um den Index der nächsten Tüte zu berechnen, wird folgendes getan: (2+1) % 3 = 0
  - => der nächste Tütenindex ist 0, also wird das nächste Geschenk in die erste Tüte gelegt

Die beiden Anforderungen werden wegen Folgendem erfüllt:

- Der Tütenindex läuft durch und wird auch bei Geschenktypwechsel nicht zurückgesetzt.
  - => Anforderung *A*
- Für jeden Geschenktyp werden die Geschenke der Reihe nach in die Tüten verteilt.
  - => Anforderung *B*

Umsetzung

Das Programm ist mit Python geschrieben.

alle_beispiele_loesen.py

- Startbares Skript
- Findet die Lösung für alle vorgegeben Beispieldateien (im Verzeichnis Input).

- Dazu für jede Datei
    1. Kreiert ein Objekt der Klasse `Parameter` (siehe unten), basierend auf dem Namen der Beispieldatei.
    2. Ruft `loese` (siehe unten) mit dem Objekt aus Punkt 1 auf, um den Inhalt der Tüten zu erhalten.
    3. Ruft `schreibe` (siehe unten) mit dem Namen der Beispieldatei und den Tüteninhalten auf, um die Lösung in eine Output-Datei zu schreiben.

input_leser.py

- Klasse `Parameter`
    - Liest Beispieldatei im Konstruktor und speichert folgende 3 Informationen in Objektattributen:
        1. Anzahl an Tütennummern
        2. Anzahl an Geschenktypen
        3. Anzahl an Geschenken eines Geschenktyps

loeser.py

- Funktion `loese`
    - Setzt die Lösungsidee um, benutzt die Informationen, die im `Parameter`-Objekt stehen
    - Gibt das Resultat (gefüllte Tüten) als `list[list[int]]` zurück
        1. Es wäre möglich, dass es mehr als 26 Geschenktypen gibt. => Die Geschenktypen werden in Zahlen statt Buchstaben angegeben.=> `int`
        2. In einer Tüte befinden sich mehrere Geschenke. => `list[int]`
        3. Das Resultat beinhaltet mehrere Tüten. => `list[list[int]]`
    - Der Typ der Geschenke wird mit einem Index im halboffenen Intervall [0,#Geschenktypen) festgehalten.

output_schreiber.py

- Funktion `schreibe`
    - Kreiert Output-Datei basierend auf dem vorgegebenen Dateinamen.
    - Dabei wird jede Tüte in eine andere Zeile geschrieben.
    - Die Geschenke einer Tüte werden durch Komma getrennt.

Beispiele

Lösungsidee am Beispiel „wundertuete2“

Input (Beispieldatei)

```
9

4

10

9

3
```

5

Nach dem ersten Geschenktyp :

Es gibt einen Überlauf, die Tüte mit Index 0 bekommt ein Geschenk mehr, der nächste Empfänger ist die Tüte mit Index 1 (Pfeil).

```
0,0
0   <--
0
0
0
0
0
0
0
```

Nach dem zweiten Geschenktyp:

Die Geschenke wurden von Tüte 1 bis Tüte 0 vergeben. => erneuter Überlauf

```
0,0,1
0,1  <--
0,1
0,1
0,1
0,1
0,1
0,1
```

Nach dem dritten Geschenktyp:

Die Geschenke wurden von Tüte 1 bis Tüte 3 vergeben.

```
0,0,1
0,1,2
0,1,2
0,1,2
0,1   <--
```

```
0,1

0,1

0,1

0,1
```

Nach dem vierten und letzten Geschenktyp:

Zufällig hat jede Tüte gleichviele Geschenke – von jedem Geschenktyp hat jede Tüte gleichviele (oder ein Geschenktyp mehr).

```
0,0,1  <--

0,1,2

0,1,2

0,1,2

0,1,3

0,1,3

0,1,3

0,1,3

0,1,3
```

Ergebnisse

Wundertuete0

```
0,0,1,2

0,1,1

0,1,2
```

Wundertuete1

```
0,0,0,1,2,2

0,0,0,1,2,2

0,0,0,1,2,2

0,0,0,1,2,2

0,0,0,1,2,2

0,0,0,1,2,2
```

Wundertuete2

```
0,0,1

0,1,2
```

0,1,2

0,1,2

0,1,3

0,1,3

0,1,3

0,1,3

0,1,3

Wundertuete3

0,1

0,1

1,2

1,2

1,2

1,2

1,2

1,2

1,3

1,3

1,4

Wundertuete4

0,0,1,1,1,1,1,1,2,2,2,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4
,4,4,4,4,4,4,5,5,5,5,5

0,0,1,1,1,1,1,1,2,2,2,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4
,4,4,4,4,4,5,5,5,5,5,5

0,0,1,1,1,1,1,1,2,2,2,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4
,4,4,4,4,4,5,5,5,5,5,5

0,0,1,1,1,1,1,1,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4
,4,4,4,4,4,5,5,5,5,5,5

0,1,1,1,1,1,1,1,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4
,4,4,4,4,4,5,5,5,5,5,5

0,1,1,1,1,1,1,1,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4
,4,4,4,4,4,5,5,5,5,5

0,1,1,1,1,1,1,1,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4
,4,4,4,4,4,5,5,5,5,5

0,1,1,1,1,1,1,1,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4
,4,4,4,4,4,5,5,5,5,5

0,1,1,1,1,1,1,1,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4
,4,4,4,4,4,5,5,5,5,5

0,1,1,1,1,1,1,1,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4
,4,4,4,4,4,5,5,5,5,5

0,1,1,1,1,1,1,1,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4
,4,4,4,4,4,5,5,5,5,5

0,1,1,1,1,1,1,1,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4
,4,4,4,4,4,5,5,5,5,5

0,1,1,1,1,1,1,1,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4
,4,4,4,4,4,5,5,5,5,5

0,1,1,1,1,1,1,1,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4
,4,4,4,4,4,5,5,5,5,5

0,1,1,1,1,1,1,1,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4
,4,4,4,4,4,5,5,5,5,5

0,1,1,1,1,1,1,1,2,2,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4
,4,4,4,4,4,5,5,5,5,5

0,1,1,1,1,1,1,2,2,2,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4
,4,4,4,4,4,5,5,5,5,5

Wundertuete5

Auszug aus den insgesamt 97 Tüten. Komplette Liste in `junior1/output/wundertuete5.txt`

Ersten 10 Tüten

0,2,2,3,4,4,6,6,8,9,10,11,13,14,14,15,15,17,18,19,19,21,21,22

0,2,2,3,4,4,6,6,8,9,10,11,13,14,14,15,16,17,18,19,19,21,21,22

0,2,2,3,4,4,6,6,8,9,10,11,13,14,14,15,16,17,18,19,19,21,21,22

0,2,2,3,4,4,6,6,8,9,10,11,13,14,14,15,16,17,18,19,19,21,21,22

0,2,2,3,4,4,6,6,8,9,10,11,13,14,14,15,16,17,18,19,19,21,21,22

0,2,2,3,4,4,6,6,8,9,10,11,13,14,14,15,16,17,18,19,19,21,21,22

0,2,2,3,4,5,6,6,8,9,10,11,13,14,14,15,16,17,18,19,19,21,21,22

0,2,2,3,4,5,6,6,8,9,10,11,13,14,14,15,16,17,18,19,19,21,21,22

0,2,2,3,4,5,6,6,8,9,10,11,13,14,14,15,16,17,18,19,19,21,21,22

0,2,2,3,4,5,6,6,8,9,10,11,13,14,14,15,16,17,18,19,19,21,21,22

Letzten 10 Tüten

1,2,3,4,4,5,6,7,9,10,11,13,14,14,15,15,17,18,19,19,21,21,22

1,2,3,4,4,5,6,7,9,10,11,13,14,14,15,15,17,18,19,19,21,21,22

1,2,3,4,4,6,6,8,9,10,11,13,14,14,15,15,17,18,19,19,21,21,22

1,2,3,4,4,6,6,8,9,10,11,13,14,14,15,15,17,18,19,19,21,21,22

2,2,3,4,4,6,6,8,9,10,11,13,14,14,15,15,17,18,19,19,21,21,22

2,2,3,4,4,6,6,8,9,10,11,13,14,14,15,15,17,18,19,19,21,21,22

2,2,3,4,4,6,6,8,9,10,11,13,14,14,15,15,17,18,19,19,21,21,22

2,2,3,4,4,6,6,8,9,10,11,13,14,14,15,15,17,18,19,19,21,21,22

2,2,3,4,4,6,6,8,9,10,11,13,14,14,15,15,17,18,19,19,21,21,22

2,2,3,4,4,6,6,8,9,10,11,13,14,14,15,15,17,18,19,19,21,21,22

Quelltext: Auszüge

```python
def loese(param: Parameter) -> list[list[int]]:
    # initialisiere fuer jede Tuete eine leere Liste von Geschenken
    result = [[] for _ in range(param.tueten_num)]
    tueten_index = 0
    for geschenk_typ_index in range(0, param.geschenk_typ_num):
        # irrelevant, wie viele Geschenke des Geschenktyps verteilt wurden => _
        for _ in range(0, param.geschenk_num_by_typ[geschenk_typ_index]):
            result[tueten_index].append(geschenk_typ_index)
            # Modulo, damit nach der letzten Tuete mit Index 0 angefangen wird
            tueten_index = (tueten_index + 1) % param.tueten_num
    return result
```