Zum Winde verweht

J2, Team-ID: 00353, Team-Name: U+1F947, Leonhard Masche, 24.09.2021

Inhaltsverzeichnis

Lösungsidee

Als erstes werden die Daten aus der ausgewählten Test-Datei in eine Matrix/zweidimensionales Array eingelesen. Mit einem for-loop wird nun über die einzelnen Tage iteriert. Jedes Clique-Mitglied kann maximal eine benötigte Veränderung verursachen, da, wenn das Mitglied einen oder mehrere Tage besser bewertet, man die Bewertung des jetzigen Tages auf `0`, also die bestmögliche, ändern kann. In einer seperaten Liste wird festgehalten, wie viele 'benötigte Änderungen' es an jedem Tag gibt. Eine 'benötigte Änderung' ist also der Fall, wenn die Bewertung des jetzigen Tages schlechter als die beste Bewertung dieses Mitglieds ist.

Gibt es einen Termin mit `0` erforderlichen Änderungen, wird dieser als 'allseits beliebter Termin' ausgegeben. Gibt es diesen Termin nicht, wird das Minimum `m` an erforderlichen Änderungen ermittelt, und der früheste Tag mit `m` erforderlichen Änderungen ausgegeben.

Umsetzung

Das Programm ist in der Sprache Python umgesetzt. Der Aufgabenordner enthält neben dieser Dokumentation eine ausführbare Python-Datei. Diese Datei ist mit einer Python-Umgebung ab der Version `3.6` ausführbar.

Wird das Programm gestartet, wird zuerst eine Eingabe in Form einer einstelligen Zahl erwartet, um ein bestimmtes Beispiel auszuwählen. *(Das heißt: `0` für Beispiel `praeferenzen0.txt`)*

Nun wird die Logik des Programms angewandt und die Ausgabe erscheint in der Kommandozeile.

Beispiele

Hier wird das Programm auf die sechs Beispiele aus dem Git-Repo, und ein weiteres (`praeferenzen6.txt`) angewendet:

`praeferenzen0.txt`

```
6 7
0 0 0 0 0 0 0
1 0 0 1 1 0 0
2 2 2 1 2 2 2
2 1 1 1 2 1 2
```

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

Ausgabe zu praeferenzen0.txt

```
Kein "allseits beliebter Termin" gefunden. 2 Änderung(en) an Tag 6 benötigt.
```

---

praeferenzen1.txt

```
5 5
0 0 0 0 0
1 1 2 2 1
2 1 1 1 1
2 2 0 1 1
2 0 0 1 1
```

Ausgabe zu praeferenzen1.txt

```
Kein "allseits beliebter Termin" gefunden. 1 Änderung(en) an Tag 2 benötigt.
```

---

praeferenzen2.txt

```
8 10
0 1 0 0 0 0 0 0 0 0
2 2 2 0 1 1 2 0 0 1
1 1 0 0 1 1 1 0 0 1
2 2 2 1 2 1 1 2 1 1
1 1 1 1 2 2 1 1 2 1
1 1 2 0 1 1 1 1 1 2
2 1 2 0 2 2 2 2 0 2
1 2 1 0 2 1 2 1 2 2
```

Ausgabe zu praeferenzen2.txt

```
"Allseits beliebter Termin" gefunden: Tag 4.
```

---

praeferenzen3.txt

```
14 20
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 0 2 1 1 2 1 2 2 1 2 0 2 1 2 2 1
2 2 1 0 1 1 1 0 2 1 0 1 1 1 2 0 0 2 2 1
1 1 0 1 2 2 2 2 1 1 1 2 1 1 1 1 1 0 1 2
2 2 1 1 2 0 0 2 0 0 1 2 2 1 2 1 0 1 1 2
1 2 1 1 2 1 2 2 1 2 1 2 2 1 1 1 1 0 2 0
0 2 2 1 1 1 1 1 0 1 1 1 1 2 2 1 1 2 1 2
1 1 2 2 1 1 1 2 0 1 0 1 1 1 2 1 1 0 0 1
0 0 2 1 2 0 1 2 2 1 1 2 1 1 2 0 1 2 0 1
1 1 1 1 0 1 1 1 1 1 1 1 1 2 1 1 1 2 2 2
2 1 1 1 2 1 1 1 0 2 1 2 0 1 0 1 0 0 2 0
2 1 1 2 1 1 0 1 2 1 1 1 2 1 1 0 1 0 2 0
0 1 0 1 2 1 1 2 2 1 0 0 1 1 1 1 1 0 1 1
0 1 2 1 1 1 1 2 1 2 1 2 2 1 1 1 2 1 0 2
```

Ausgabe zu praeferenzen3.txt

```
Kein "allseits beliebter Termin" gefunden. 7 Änderung(en) an Tag 18 benötigt.
```

---

praeferenzen4.txt

```
26 40
0 0 0 0 0 ... 1 0 0 0 0
0 1 1 0 1 ... 1 0 2 2 1
0 1 1 0 1 ... 2 0 2 1 1
1 1 1 2 2 ... 2 1 2 2 1
1 1 1 2 1 ... 1 1 1 2 0
  ⋮   ⋮   ⋮   ⋮   ⋮   ⋮
2 1 2 1 2 ... 2 0 1 2 2
2 1 1 1 0 ... 1 2 2 2 1
2 0 0 1 1 ... 2 1 1 1 2
1 1 1 2 1 ... 2 1 1 1 1
2 0 2 2 1 ... 1 1 2 1 1
```

Ausgabe zu praeferenzen4.txt

```
Kein "allseits beliebter Termin" gefunden. 14 Änderung(en) an Tag 22 benötigt.
```

---

praeferenzen5.txt

```
50 80
0 0 0 1 0 ... 0 0 0 0 0
1 0 0 0 1 ... 2 0 1 0 2
1 2 1 1 0 ... 1 1 0 1 0
2 2 2 2 0 ... 2 2 1 2 0
2 2 1 1 1 ... 1 1 1 2 1
  ⋮   ⋮   ⋮   ⋮   ⋮   ⋮
1 2 2 0 0 ... 1 1 1 1 1
1 0 1 2 2 ... 1 2 1 2 0
2 2 0 1 1 ... 1 1 2 1 2
2 1 1 1 1 ... 0 1 1 0 1
1 2 1 1 2 ... 2 1 1 1 2
```

Ausgabe zu praeferenzen5.txt

```
Kein "allseits beliebter Termin" gefunden. 34 Änderung(en) an Tag 31 benötigt.
```

---

praeferenzen6.txt

```
8 7
2 2 2 1 2 0 2
2 2 2 1 2 0 2
2 2 2 1 1 0 2
2 2 2 0 2 0 2
2 2 2 1 2 0 2
2 2 2 1 2 0 2
2 2 2 1 2 0 2
2 2 2 1 2 0 2
Hinzugefügtes Beispiel
```

Ausgabe zu praeferenzen6.txt

```
"Allseits beliebter Termin" gefunden: Tag 6.
```

Quellcode

```python
# pylama:ignore=E501
from os import path

# absoluter Pfad des ausgewählten Beispiels
path = path.join(
    path.dirname(path.abspath(__file__)),
    f'beispieldaten/praeferenzen{input("Nummer des Beispiels eingeben: ")}.txt')
```

```
with open(path, 'r') as f:
    lines = f.read().split('\n')


# Datenklasse aus einem anderen Projekt
class ilist(list):
    def __init__(self, start=None, empty=None):
        if start is None:
            start = []
        list.__init__(self, start)
        self.empty = empty

    def _ensure_length(self, n):
        maxindex = n
        if isinstance(maxindex, slice):
            maxindex = maxindex.indices(len(self))[1]
        while len(self) <= maxindex:
            self.append(self.empty)

    def __getitem__(self, n):
        self._ensure_length(n)
        return super(ilist, self).__getitem__(n)

    def __setitem__(self, n, val):
        self._ensure_length(n)
        return super(ilist, self).__setitem__(n, val)


# die Datei einlesen
n, m = tuple(lines[0].split(' '))
n, m = int(n), int(m)
matrix = []
for line in lines[1:n+1]:
    matrix.append([int(x) for x in line.split(' ')])


# durch die Tage iterieren
neededchanges = ilist(empty=0)
for i in range(m):
    for prefs in matrix:
        # Anzahl der Änderungen wird inkrementiert,
        # wenn die Bewertung dieses Tages schlechter
        # als die beste Bewertung dieses Mitglieds ist
        if prefs[i] > min(prefs):
            neededchanges[i] += 1

# Ausgabe-Logik
changes = min(neededchanges)
if changes == 0:
    print(
        f'"Allseits beliebter Termin" gefunden: Tag {neededchanges.index(0)+1}.')
else:
    print(
```

```
        f'Kein "allseits beliebter Termin" gefunden. {changes} Änderung(en) an Tag
{neededchanges.index(changes)+1} benötigt.')
```