

Junioraufgabe 2: Treffsicherheit

Team-ID: 00694

Team-Name: Der Teufel schreibt Java

Bearbeiter/-innen dieser Aufgabe:
Frederico Aberle

21. November 2021

Inhaltsverzeichnis

1	Lösungsidee	1
2	Umsetzung	1
3	Beispiele	2
4	Quellcode	2

1 Lösungsidee

Wenn wir für jede Person den geringsten Value herausfinden würden, d.h. den für diese Person beste Antwort von den Antworten 'sehr gut', 'mäßig' und 'überhaupt nicht', dann könnten wir daraus für jeden Termin bestimmen, wie viele Änderungen in der Präferenztafel gemacht werden müsste, damit der Termin 'allseits beliebt' wäre. Bei bekanntem geringsten Value können wir je Termin die Antwort des Mitglieds für diesen Termin mit dem geringsten Value vergleichen. Stimmen diese miteinander überein muss nichts an der Präferenztafel geändert werden, ansonsten schon. Damit können wir zählen wie viele Änderungen dieser Termin benötigt. Am Ende suchen wir noch den Termin mit den geringsten Änderungen. Das ist dann unsere Lösung.

2 Umsetzung

Die Präferenzen speichern wir in einer $n \times m$ Matrix p , wobei n für die Anzahl der Mitglieder steht und m für die Anzahl der Termine. In der Liste `minValues` speichern wir für jedes Mitglied die 'beste' Antwort von allen Terminen der Person. Eine 0 entspricht dabei der Antwort 'sehr gut', eine 1 'mäßig' und eine 2 'überhaupt nicht'. In der Liste `changes` speichern wir die Anzahl der Änderungen, die in der Präferenztafel gemacht werden müssen, damit der Termin 'allseits beliebt' ist. Um die Änderungen zu bestimmen, iterieren wir in einer for-Loop durch die Termine und in einer zweiten Nested for-Loop durch die Mitglieder. Dann können wir die Präferenz $p[i][j]$ mit $1 \leq i \leq n$ und $1 \leq j \leq m$ mit `minValue[i]` vergleichen. Falls diese nicht übereinstimmen sollten, wissen wir, dass für diesen Termin eine Änderung für die jeweilige Person gemacht werden muss. Diese Änderung halten wir in der Liste `changes` fest, indem wir `changes[j]` um 1 erhöhen. Das Resultat ist nach Beenden der beiden Schleifen das Minimum von `changes`, da es den Termin mit den wenigsten Änderungen in der Präferenztafel darstellt.

3 Beispiele

Es werden für die Testbeispiele folgende Ausgaben erzeugt:

praeferenzen0.txt

Es müssen mindestens 2 Einträge verändert werden.

Der Termin dafür lautet:

[0, 0, 2, 1, 0, 1]

praeferenzen1.txt

Es müssen mindestens 1 Einträge verändert werden.

Der Termin dafür lautet:

[0, 1, 1, 2, 0]

praeferenzen2.txt

Es müssen mindestens 0 Einträge verändert werden.

Der Termin dafür lautet:

[0, 0, 0, 1, 1, 0, 0, 0]

praeferenzen3.txt

Es müssen mindestens 7 Einträge verändert werden.

Der Termin dafür lautet:

[0, 2, 2, 0, 1, 0, 2, 0, 2, 2, 0, 0, 0, 1]

praeferenzen4.txt

Es müssen mindestens 14 Einträge verändert werden.

Der Termin dafür lautet:

[0, 1, 2, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 2, 1, 1, 0, 2, 2, 0, 2, 0, 1, 0]

praeferenzen5.txt

Es müssen mindestens 34 Einträge verändert werden.

Der Termin dafür lautet:

[0, 1, 2, 2, 2, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 2, 1, 2, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 0, 2, 0, 1, 0, 1, 2, 0, 2, 1]

4 Quellcode

```

1 file = open("praeferenzen5.txt")
  firstLine = file.readline().strip().split()
3
  n = int(firstLine[0]) # n Mitglieder
5 m = int(firstLine[1]) # m Termine

7 p = file.read().splitlines() # nxm Praeferenzen
  for i in range(len(p)):
9     p[i] = p[i].split()
    p[i] = [int(x) for x in p[i]]
11
  # Berechnet fuer jedes Mitglied den Value fuer den beliebtesten Termin
13 minValues = [0 for i in range(n)]
  for i in range(len(p)):
15     minValues[i] = min(p[i])

17 # Zaehlt wie viele Aenderungen pro Termin gemacht werden
  # muessen, sodass Termin fuer jedes Mitglied am besten gefaellt
19 changes = [0 for i in range(m)]
  for termin in range(m):
21     for mitglied in range(n):
        if p[mitglied][termin] > minValues[mitglied]:
23         changes[termin] += 1
  result = min(changes)
25
  # Gibt den resultierenden Termin aus
27 inTerminIndex = changes.index(result)
  resultTermin = []
29 for i in range(n):
    resultTermin.append(p[i][inTerminIndex])

```