

Aufgabe 2: Treffsicherheit

Team-ID: 00833

Team-Name: Gewinner bwinf

Bearbeiter dieser Aufgabe:
Jonas Jacob Biermann

22. November 2021

Inhaltsverzeichnis

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

1 Lösungsidee

Gegeben ist eine 2 Dimensionale Tabelle (Beispieldateien). Die erste Zeile der Beispieldateien liefert die Anzahl der Personen, die in der Freundesgruppe sind und die Anzahl der Termine die zur Verfügung stehen. Die Tabelle ist wie folgt aufgebaut. Es gibt so viele Zeilen wie es Personen gibt und so viele Spalten wie es mögliche Termine gibt. Jedem Termin wird ein Status zugeordnet. Dieser beschreibt, ob die jeweilige Person an dem bestimmten Tag kann. Dies wird gemessen in 0, 1, 2, wobei 0 das beste und 2 das schlechteste ist. Jetzt gilt es herauszufinden wie viele Änderungen es braucht um einen sogenannten allseits beliebten Termin zu erreichen. Das ist ein Termin, der so beliebt ist, dass keinem einzigen Mitglied ein anderer Termin besser gefällt. Um dies zu erreichen ist die Idee mithilfe der 1. Zeile, die Zeile, die die Anzahl der Personen und die Anzahl der Termine liefert eine for Schleife zu nutzen, die erst einen Array erstellt, in dem die Werte der Tabelle (0-2) abgespeichert werden können. Dementsprechend ist ein Wert ein subarray vom subarray des ursprünglich erstellten Arrays. Also ist im erstellten array die erste Dimension alle Werte, die in der Tabelle waren, die 2. Dimension, diese Werte nach Zeilen aufgeteilt und in der 3. Dimension sind die Werte aus den einzelnen Zeilen aufgeteilt. Nun wird dieser Array genutzt um einen Array zu erstellen, der die Anzahl der nötigen Änderungen für jeden Termin festhält. Um dies zu erreichen werden 3 for-Schleifen genutzt, die es ermöglichen den Array für jeden zu vergleichenden Termin zu untersuchen. Hier wird nun für jeden Termin der Status vom ersten Termin in der Zeile mit dem Status der folgenden Termine verglichen. So wird jede Zeile durchgegangen was am Ende einen Array übrig lässt, der die Anzahl der benötigten Änderung für jede Zeile hat. Dieser Array wird anschließend nach dem kleinsten Element durchsucht. Nachdem das kleinste Element bestimmt wurde wird die Position des Elements in der erstellten Liste bestimmt und mit 1 addiert, da in einer Liste die Zählung immer mit 0 beginnt. Zum Schluss wird die minimal benötigte Anzahl an Terminänderungen und der perfekte Termin ausgegeben.

2 Umsetzung

```
1  eingabe = open('praeferenzen0.txt', 'r')
2  zeile_1 = eingabe.readline()
3  zeile_1 = zeile_1.split()
4
5  personen_Anzahl = int(zeile_1[0])
6  termin_Anzahl = int(zeile_1[1])
7
```

Der oben eingefügte Screenshot beschreibt wie die Datei geöffnet und die erste Zeile eingelesen wird. Das einlesen wird mithilfe der Funktion `.readline()` durchgeführt. Dadurch wird die 1. Zeile der Datei eingelesen und als `zeile_1` definiert. Mithilfe der `.split()` Funktion wird der eingelesene Input separiert. Anschließend werden 2 Variablen definiert. Die `personen_Anzahl` Variable und die `termin_Anzahl` Variable. Das erste Element der Liste wird der Anzahl der Personen und das 2. Element der Liste wird der Anzahl der Termine zugeordnet. Mithilfe von `int()` werden die Variablen zum Typ integer. Dies ist bei den for-Schleifen im Verlaufe des Programmes noch wichtig.

```
8  x = 0
9  ges_Status = []
10
11 for i, zeile in enumerate(eingabe):
12     zeilen_Status = []
13     for j in range(termin_Anzahl):
14         zeilen_Status.append([int(zeile.rstrip().split()[x])])
15         x += 1
16     x = 0
17     ges_Status.append(zeilen_Status)
```

Hier wird zum ersten mal eine for-Schleife ins Spiel gebracht. Zu Anfang wird die später genutzte Variable `x` auf den Ausgangswert 0 gesetzt und eine leere Liste (`ges_Status`) erstellt. In der for-Schleife wird mithilfe der `enumerate()` Funktion gezählt, wie viele Durchgänge schon abgelaufen sind. Bei jedem Durchgang der ersten for-Schleife wird also jetzt eine leere Liste namens `zeilen_Status` erstellt. Anschließend läuft eine zweite for-Schleife ab. Hier wird die vorher definierte Variable `termin_Anzahl` genutzt. Jetzt wird für die Anzahl der Termine die soeben definierte Liste `zeilen_Status` mit dem jeweiligen Wert in der Zeile erweitert. Hierbei wird mit `int()` sichergestellt, dass am Ende eine Liste aus Integern entsteht. Die `.rstrip()` Funktion ermöglicht es nachgestellte Leerzeichen und ähnliches zu löschen. Mit `.split()` werden, wie oben schon erklärt, die Elemente separiert. Da nun jedes Element der Zeile in dem Array hinzugefügt werden soll sorgt die Variable `x` dafür, dass die Zeile einmal abgearbeitet wird. Nachdem eine Zeile abgearbeitet wurde wird `x` wieder auf 0 gesetzt um den Prozess in der for-Schleife erneut zu ermöglichen. Außerdem wird die am Anfang definierte Liste `ges_Status` mit der Liste `zeilen_Status` erweitert. Dadurch erhält man den in der Lösungsidee beschriebenen 3-dimensionalen Array.

```
18
19 anzahl_aenderungen = 0
20 aenderungen_liste = []
21
22 for k in range(termin_Anzahl):
23     for l in range(personen_Anzahl):
24         for m in range(termin_Anzahl):
25             if ges_Status[1][x]>ges_Status[1][m]:
26                 anzahl_aenderungen += 1
27                 break
28             else: pass
29         aenderungen_liste.append(anzahl_aenderungen)
30         x += 1
31     anzahl_aenderungen = 0
```

Die hier gezeigte for-Schleife ist die Schleife, die die Liste mit den nötigen Änderungen erzeugt. Die Zeilen 19 und 20 sind selbst erklärend. Da die for-Schleife für jeden Termin einmal durchgeführt werden muss läuft die 1. for-Schleife in der Reichweite der zuvor definierten Variabel termin_Anzahl ab. Die nächsten beiden for-Schleifen sind wichtig um durch den Array durch zu navigieren. Da l und m im Prinzip Zähler für die Anzahl der Wiederholungen sind kann mithilfe dieser durch den Array navigiert werden. Da jede Zeile einzeln durchgegangen werden muss und in einer Zeile mehrere Spalten sind kommt erst die Schleife mit der Reichweite personen_Anzahl und dann die Schleife mit der Reichweite termin_Anzahl. In der 3. for Schleife befindet sich eine if-Abfrage. Diese if-Abfrage vergleicht den Status des ersten Termins in der Zeile mit dem Status des aktuell untersuchten Termins (m). Sollte die Bedingung der if-Abfrage erfüllt werden wird die Anzahl der Änderungen mit 1 addiert und die Schleife wird gestoppt. Anschließend läuft die 3. Schleife erneut ab. Sollte die Bedingung nicht erfüllt sein findet ein erneuter Durchlauf von Schleife 3 statt. Wenn die Schleife 3 für eine Zeile durchgelaufen ist wird die am Anfang definierte Liste mit der Anzahl der Änderungen erweitert und x wird um ein erhöht. Außerdem muss die Variable anzahl_aenderungen wieder auf 0 gesetzt werden, damit keine falschen Ergebnisse entstehen. Dies ist das

```
32
33 min_aenderungen = min(aenderungen_liste)
34 pos_termin = aenderungen_liste.index(min_aenderungen)+1
35 print('Der beste Termin ist der', pos_termin, '. Tag, es müssen', min_aenderungen, 'vorgenommen werden.')
```

Ende der Datei. Mithilfe von min() wird das kleinste Element in der Änderungsliste gefunden, was dann die minimalen nötigen Änderungen für einen perfekten Termin sind. mithilfe .index() kann nach dem Index eines Elements (in diesem fall min_aenderungen) gesucht werden. Da ein die Zählung bei einem Index immer mit 0 beginnt muss das Ergebnis der index-Funktion noch um 1 erhöht werden. Zu guter letzt wird sowohl der beste Termin als auch die minimale, nötige Anzahl an Änderungen die vorgenommen werden müssen ausgegeben.

3 Beispiele

```
Der beste Termin ist der 6 . Tag, es müssen 2 vorgenommen werden.  
Der beste Termin ist der 2 . Tag, es müssen 1 vorgenommen werden.  
Der beste Termin ist der 4 . Tag, es müssen 0 vorgenommen werden.  
Der beste Termin ist der 18 . Tag, es müssen 7 vorgenommen werden.  
Der beste Termin ist der 22 . Tag, es müssen 14 vorgenommen werden.  
Der beste Termin ist der 31 . Tag, es müssen 34 vorgenommen werden.
```

Dies sind die Ausgaben des Programms für die gegebenen Beispieldateien, praeferenzen0-praeferenzen5.

4 Quellcode

Der gesamte Quellcode sieht wie folgt aus:

```
1  eingabe = open('praeferenzen0.txt', 'r')  
2  zeile_1 = eingabe.readline()  
3  zeile_1 = zeile_1.split()  
4  
5  personen_Anzahl = int(zeile_1[0])  
6  termin_Anzahl = int(zeile_1[1])  
7  
8  x = 0  
9  ges_Status = []  
10  
11  for i, zeile in enumerate(eingabe):  
12      zeilen_Status = []  
13      for j in range(termin_Anzahl):  
14          zeilen_Status.append([int(zeile.rstrip().split()[x])])  
15          x += 1  
16      x = 0  
17      ges_Status.append(zeilen_Status)  
18  
19  anzahl_aenderungen = 0  
20  aenderungen_liste = []  
21  
22  for k in range(termin_Anzahl):  
23      for l in range(personen_Anzahl):  
24          for m in range(termin_Anzahl):  
25              if ges_Status[l][x] > ges_Status[l][m]:  
26                  anzahl_aenderungen += 1  
27                  break  
28              else: pass  
29      aenderungen_liste.append(anzahl_aenderungen)  
30      x += 1  
31      anzahl_aenderungen = 0  
32  
33  min_aenderungen = min(aenderungen_liste)  
34  pos_termin = aenderungen_liste.index(min_aenderungen)+1  
35  print('Der beste Termin ist der', pos_termin, '. Tag, es müssen', min_aenderungen, 'vorgenommen werden.')
```