Aufgabe 2: Spießgesellen

Teilnahme-ID: 57048

Bearbeiter/-in dieser Aufgabe:
Frederico Aberle

19. April 2021

Inhaltsverzeichnis

1 Lösungsidee

1.1 Teilaufgabe a)

Aus Aussage (1) und (2) ist bekannt, dass dort Schüssel 5 und Banane beides Mal vorkommen. Das trifft auf keine andere Schüssel bzw. Obstsorte zu. Folglich ist Banane in Schüssel 5. Analog ist Erdbeere in Schüssel 2, Weintraube in Schüssel 3 und Pflaume in Schüssel 6. Apfel und Brombeere sowie Schüssel 1 und 4 sind beides Mal in Aussage in Aussage (1) und (3) enthalten. Es lässt sich aber nicht eindeutig zuordnen welche von den beiden Obstsorten zu welchen von den beiden Schüsseln gehört. Da die beiden Obstsorten aber beide zu Donalds Lieblingssorten gehören ist es egal. Donald würde bei einen von den beiden Schüsseln immer eine seiner Lieblingssorte greifen. Folglich muss sich Donald aus den Schüsseln 1, 3 und 4 bedienen, um eine seiner Lieblingssorten zu greifen.

1.2 Teilaufgabe b)

Die Idee ist es die Obstsorten nach und nach den Schüsseln zuzuordnen. Dabei kann eine Obstsorte mehreren Schüsseln gleichzeitig zugeordnet werden, falls eine Zuordnung nicht eindeutig ist. Die Schüssel erfährt also eine Reihe von potenziellen Kandidaten unter denen sich die 'tatsächliche' Obstsorte befindet.

Wir haben demnach n Schüssel auf der Party, die von 1 bis n nummeriert sind. Um nun eine Obstsorte mehreren Schüsseln zuordnen zu können, wird verglichen, ob die Beobachtungen dieser Obstsorte mit den Beobachtungen der jeweiligen Schüssel übereinstimmen. Das heißt, wenn die Obstsorte 'Apfel' in der

Beobachtung Nummer 1 und 2 vorkommt und zum Beispiel die Schüssel Nummer 8 in der Beobachtung Nummer 1 und 2 vorkommt, bedeutet das, dass Apfel potenziell in der Schüssel Nummer 8 liegt. Apfel und Schüssel Nummer 8 sind ein 'Matching'.

Das wird mit allen Obstsorten aus allen Beobachtungen durchgeführt, um keine Obstsorte wegzulassen. Es ist auch nötig alle Obstsorten aus den Beobachtungen den Schüsseln zuzuordnen und nicht nur die Obstsorten aus Donalds Wunschsorten, wie man meinen könnte, da für eine Schüssel unter den potenziellen Kandidaten sich auch keine Wunschsorte befinden kann. In diesem Fall kann man nicht mehr sicher davon ausgehen, dass sich nur potenzielle Wunschsorten in der Schüssel befinden und es kann sein, dass beim Ziehen aus der Schüssel eine nicht gewünschte Obstsorte rauskommt. Weiterhin ist es erforderlich alle Obstsorten aus den Beobachtungen den Schüsseln zuzuordnen, da man daraus schließen kann, welche die übrig gebliebenen Obstsorten sind, also die Obstsorten, die nicht in den Beobachtungen vorkommen. Diese Obstsorten werden dann den Schüsseln zugeordnet, die auch nicht in den Beobachtungen vorkommen. Und unter diesen übrig gebliebenen Obstsorten beziehungsweise Schüsseln kann sich auch eine Obstsorte von Donalds Wunschsorten befinden, die für die Lösung relevant ist.

Zum Schluss muss man noch für jede Schüssel sichergehen, dass diejenigen Schüsseln mit Donalds Wunschsorten auch nur potenzielle Obstsorten aus Donalds Wunschsorten enthalten. Nur so kann man für die jeweilige Schüssel sicher sein, dass man eine von Donalds Wunschsorte greift. Falls eine Schüssel beides, also Wunschsorte und keine Wunschsorte enthält, kann Donald die Menge der Schüsseln, in denen Wunschsorten zu finden sind, nicht eindeutig bestimmen.

2 Umsetzung

Zunächst erstellen wir das Dictionary a für Abkürzungen. So behält man besser den Überblick bei Zwischenergebnissen. Der erste Buchstabe jedes Elements aus der Liste von Donalds Wunschsorten und aus der Liste von den Beobachtungen der Obstsorten soll als key im Dictionary gespeichert werden. Value ist die Obstsorte selbst.

Danach erstellen wir das Dictionary result in dem für jede Schlüssel (key) von 1 bis n die potenziellen Obstsorten (value) gespeichert werden.

Das gleich aufgebaute Dictionary observation speichert die Beobachtung der Schüsseln in Form einer Liste (value) für jede Schüssel (key) von 1 bis n. Zur Speicherung der Beobachtung wird für jedes Element, also jede Schüssel, in observation geprüft, ob sie in den N Beobachtungen vorkommt. Falls die Schüssel in der i-ten Beobachtung vorkommt, wird an der Stelle i in der Liste eine '1' hinzugefügt. Falls die Schüssel in der i-ten Beobachtung nicht vorkommt, wird eine '0' hinzugefügt. Zum Beispiel wird für die 8. Schüssel der Wert [1, 1, 0, 0] gespeichert. Das heißt, dass in der ersten und zweiten Beobachtung aus der Schüssel Nummer 8 gegriffen wurde.

Da wir nun alle Beobachtungen der Schüsseln gespeichert haben, müssen wir diese nur noch mit den Beobachtungen der Obstsorten vergleichen. Dabei iterieren wir durch die Beobachtungen der Obstspieße (die alle in der Liste fruits gespeichert sind) und durch jedes Element der Beobachtungen der Obstspieße. So haben wir jede Obstsorte abgedeckt. Danach prüft man, ob die Obstsorte nicht schon zuvor gecheckt wurde, um unnötige Iterationen und unerwünschte Änderungen an unserem Ergebnis zu vermeiden. Dazu legt man vor Eintritt in die for-Schleifen eine Liste mit Boolean-Werten an. Falls die Obstsorte noch nicht vorgekommen ist, soll im nächsten Schritt nochmal durch die Beobachtungen der Obstspieße iteriert werden und geprüft werden, ob die ausgewählte Obstsorte in den Beobachtungen der Obstspieße enthalten ist. Danach soll wie beim Dictionary observation eine Liste angelegt werden, die angibt in welchen Beobachtungen die Obstsorte zu finden ist. '1' an der Stelle i bedeutet, dass die Obstsorte in der i-ten Beobachtungen zu finden ist, andernfalls soll '0' der Liste hinzugefügt werden. Zuletzt wird gesucht, wo die Liste mit den Beobachtungen der Obstsorte gleich der Liste mit den Beobachtungen der Schüssel ist. Dies erreicht man, indem man durch das Dictionary observation iteriert und prüft, ob der value von observation an der Stelle i gleich der Liste mit den Beobachtungen der Obstsorte ist. Falls ja, wird die Obstsorte an der Stelle i im Dictionary result hinzugefügt und der Boolean-Wert der Obstsorte auf True gesetzt.

Nachdem alle Obstsorten aus den Beobachten zugeordnet wurden, fehlen noch die übrig gebliebenen. Der Vorteil ist, dass man einfach durch die Boolean-Liste iterieren kann, um zu schauen an welchem Index der Wert noch auf False gesetzt ist. Alle diese Indexe werden in den Buchstaben konvertiert und in der Liste left gespeichert. Zum Schluss wird durch result iteriert, um nach leeren Listen zu schauen. Die leeren Listen werden mit der Liste left erweitert.

Übrig bleibt also das Dictionary result, das jeder Schüssel mehrere potenzielle Obstsorten zuordnet. Falls nun eine Schüssel eine Wunschsorte und keine Wunschsorte enthält, kann Donald die Menge der

Schüsseln, in denen Wunschsorten zu finden sind, nicht eindeutig bestimmen. Falls die Schüssel nur Wunschsorten enthält soll die Schüssel zur Menge hinzugefügt werden und anschließend nach der Iteration die Menge ausgegeben werden.

3 Laufzeit

Man iteriert durch die Beobachtungen, um jede einzelne Obstsorte auf sein Vorkommen in den Beobachtungen zu überprüfen. Dabei werden aber bereits geprüfte Obstsorten übersprungen. Deswegen werden auch maximal die Anzahl an Obstsorten geprüft. Für jedes dieser Obstsorten iteriert man nochmal durch die Anzahl der Beobachtungen und prüft, ob die Obstsorte in der jeweiligen Beobachtung enthalten ist. Hier ist also zusätzlich die Länge der Beobachtungen entscheidend. Für den Worst Case ergibt sich also folgende Laufzeit:

$$\mathcal{O}(N^2 \cdot m \cdot l) \tag{1}$$

Wobei N für die Anzahl der Beobachtungen, m für die Anzahl der Obstsorten und l für die maximale Länge der Beobachtungen steht.

4 Beispiele

4.1 spiesse1.txt

Donald soll sich aus den Schüsseln 1, 2, 4, 5, 7 bedienen, um eine seiner Lieblingssorten zu ziehen.

4.2 spiesse2.txt

Donald soll sich aus den Schüsseln 1, 5, 6, 7, 10, 11 bedienen, um eine seiner Lieblingssorten zu ziehen.

4.3 spiesse3.txt

Das Ergebnis ist nicht eindeutig, da unter den potenziellen Schüsseln auch die Sorten Grapefruit vorkommen.

4.4 spiesse4.txt

Donald soll sich aus den Schüsseln 2, 6, 7, 8, 9, 12, 13, 14 bedienen, um eine seiner Lieblingssorten zu ziehen.

4.5 spiesse5.txt

Donald soll sich aus den Schüsseln 1, 2, 3, 4, 5, 6, 9, 10, 12, 14, 16, 19, 20 bedienen, um eine seiner Lieblingssorten zu ziehen.

4.6 spiesse6.txt

Donald soll sich aus den Schüsseln 4, 6, 7, 10, 11, 15, 18, 20 bedienen, um eine seiner Lieblingssorten zu ziehen.

4.7 spiesse7.txt

Das Ergebnis ist nicht eindeutig, da unter den potenziellen Schüsseln auch die Sorten Litschi, Banane vorkommen.

5 Quellcode

```
"""Eingabe"""
file = open('spiesse7.txt')
numFruits = int(file.readline())  # Anzahl der verfuegbaren Obstsorten
# print(numFruits)
fav = file.readline().split()  # Donalds Wunschsorten ('Favoriten')
# print(fav)
N = int(file.readline())  # Anzahl N der beobachteten Obstspiesse
# print(N)
# in je zwei Zeilen hintereinander eine Beobachtung; in der ersten Zeile die Menge der Schuesselnummern
# der zweiten Zeile angegebenen Obstsorten stammen ('bowls and fruits')
bowls = []
fruits = []
for i in range(N):
    bowls.append(file.readline().split())
    fruits.append(file.readline().split())
# print(bowls)
# print(fruits)

"""Variablen"""
# erstellt dictionary mit Anfangsbuchstabe als key und Obstsorte als value ('a' fuer 'abbreviations')
# aendert Obstsorten von fav und baf zu den Anfangsbuchstaben der Obstsorten
a = {}

for i in range(len(fav)):
    a[fav[i][0]] = fav[i]
    fav[i] = fav[i][0]

for i in range(len(fruits)):
    for j in range(len(fruits[i])):
        a[fruits[i][j][0]] = fruits[i][j]
        fruits[i][j] = fruits[i][j][0]

a = dict(sorted(a.items()))
# print(a)

# dictionary, das jeder Schuessel eine Obstsorte zuordnen soll
result = {}

for i in range(1, numFruits+1):
    result[str(i)] = []
# print(result)

# gleich aufgebauter dictionary wie 'result'
# values geben an, ob der Key (Schuessel) in den Beobachtungen ('observation') vorkommt
# 1 an der Stelle i bedeutet, dass die Schuessel in der i-ten Beobachtung vorkommt
# 0 an der Stelle i bedeutet, dass die Schuessel in der i-ten Beobachtung nicht vorkommt
observation = {}

for i in range(1, numFruits+1):
    observation[str(i)] = []

for x in observation:
    for y in bowls:
        if x in y:
            observation[x].append(1)
        else:
            observation[x].append(0)
# print(observation)

# Liste mit dem boolean-Wert, ob die Obstsorte schon einer Schuessel zugeordnet wurde
checked = [False] * numFruits

"""Programm"""
for i in fruits:
    # iteriert durch die Beobachtungen der Obstsorten
    for j in i:
        # iteriert durch die Obstsorten
        if not checked[ord(j) - ord('A')]:
            # prueft, ob eine Obstsorte schon einer Schuessel zugeordnet wurde
            temp = []
            for k in fruits:
```

```
                # erstellt eine Liste mit der Beobachtung der Obstsorte
                if j in k:
                    temp.append(1)
                else:
                    temp.append(0)
            for l in observation:
                # vergleicht die Liste mit der Beobachtung der Obstsorte mit der Beobachtung der
                # Schuessel und ordnet ggf. zu
                if observation[l] == temp:
                    result[l].append(j)
                    checked[ord(j) - ord('A')] = True

# nicht zugeordnete Schuesseln werden mit nicht gepruefte Obstsorten verglichen und ggf. zugeordnet
left = []

for i in range(len(checked)):
    if not checked[i]:
        left.append(chr(ord('A') + i))

for x in result:
    if result[x] == []:
        result[x].extend(left)

"""Ausgabe"""
print(fav)
# print(bowls)
# print(fruits)
# print(result)

r = []  # speichert alle richtigen Obstsorten
w = []  # speichert alle falschen Obstsorten, die in einer Schuessel mit moeglichen richtigen
# Obstsorten enthalten sind
for i in result:
    x = result[i]
    b1 = False
    b2 = False
    temp1 = []
    temp2 = []
    for y in x:
        if y in fav:
            b1 = True
            if i not in temp1:
                temp1.append(i)
        else:
            b2 = True
            if y not in w:
                temp2.append(y)
    r.extend(temp1)
    if b1 and b2:
        w.extend(temp2)

for i in range(len(w)):
    w[i] = a[w[i]]

if w == []:
    print('Donald soll sich aus den Schuesseln ', end='')
    print(', '.join(r), end='')
    print(' bedienen, um eine seiner Lieblingssorten zu ziehen.')
else:
    print('Das Ergebnis ist nicht eindeutig, da unter den potenziellen Schuesseln auch die Sorten ', end
    print(', '.join(w), end='')
    print(' vorkommen.')
```