

Aufgabe 1: Wörter aufräumen

Teilnahme-Id: 56043

Bearbeiter/-in dieser Aufgabe:
Sebastian Brunnert

21. September 2020

Inhaltsverzeichnis

Lösungsidee.....	1
Umsetzung.....	1
Beispiele.....	2
Quellcode.....	2

Lösungsidee

Zum Lösen dieser Aufgabe, müssen anfangs erstmals für jedes unbekannte Wort jene Möglichkeiten von einsetzbaren Wörtern gefunden werden, welche die selbe Länge besitzen und an den passenden Stellen die selben Buchstaben wie das zu ersetzende Wort besitzen. Nachdem alle Möglichkeiten gefunden wurden, muss für jedes unbekannte Wort nun die richtige Möglichkeit gefunden werden. Also müssen die möglichen Wörter so verteilt werden, dass jedes unbekannte Wort eine eindeutige Ersetzung besitzt. Bei unbekannten Wörtern, die nur eine eine Möglichkeit besitzen, ist klar, dass dies das Wort sein wird. Diese Wort-Möglichkeit darf also nicht mehr bei anderen Wörtern verwendet werden. Existieren mehr als zwei Möglichkeiten bei einem Wort muss also nur geprüft werden, ob das unbekannte Wort das einzige Wort ist, welches diese Wort-Möglichkeit zugeordnet wurde. Ist dies der Fall, so ist das gesuchte Wort diese Wort-Möglichkeit. Sind nun alle Wörter gefunden, so kann der Satz gebildet werden.

Umsetzung

Die Umsetzung dieser Aufgabe geschieht in der Programmiersprache Python (3). Zu Beginn des Programmes müssen der zu bildene Satz und alle Wort Möglichkeiten eingelesen werden. Die möglichen Wörter werden in einer Liste gesichert. Der Satz selbst wird in die verschiedenen Wörter aufgeteilt. Dabei wird auf Satzzeichen sowie Zeilenumbrüche verzichtet. Alle Wörter werden in einem Python-Dictionary gespeichert, in welchem der Key das unbekannte Wort ist und der Value eine Liste, in welcher Wort-Möglichkeiten gespeichert werden sollen. Da es möglich ist, dass unbekannte Wörter mehrfach in einem Satz auftreten, wird das Wort eindeutig gespeichert in dem am Ende n Leerzeichen zu diesem Wort hinzugefügt werden (n = Die Stelle des Wortes im Satz). Aufgrunddessen wird auch bei der algorithmischen Arbeit mit einem zu suchendem Wort die Methode .strip() verwendet, welche diese Leerzeichen wieder entfernt.

In einem for-Loop werden nun alle Wort-Möglichkeiten gesucht, welche die selbe Länge besitzen. Außerdem wird auch mittels eines for-Loop, welcher durch alle Buchstaben in dem suchenden Wort iteriert, geprüft, ob die bekannten Buchstaben auch in dem möglichen Wort an selber Stelle vorhanden sind. Dieses Wort wird anschließend als „möglich“ für dieses unbekannte Wort gekennzeichnet, indem es in die Liste der möglichen Wörter im zuvor aufgestellten Dictionary hinzugefügt wird.

Nun müssen alle Wörter passend verteilt werden. Dazu wird durch alle zu suchenden Wörter iteriert und wie in der Lösungsidee beschrieben, wird geprüft ob die Länge der Möglichkeiten für dieses Wort eins entspricht. Ist dies der Fall so wurde die Einsetzung für dieses Wort gefunden und diese Wort-Möglichkeit muss aus allen anderen Listen von Einsetzungs-Möglichkeiten von anderen Wörtern entfernt werden. Entspricht die Anzahl an Möglichkeiten allerdings nicht Eins, so wird durch die Möglichkeiten der Einsetzung wieder durchgelaufen. Für jede Möglichkeit wird geprüft, ob andere zu suchende Wörter dieses Wort-Möglichkeit besitzen. Wenn dies bei keinem unbekannten Wort der Fall ist, muss dieses mögliche Wort die Einsetzung für das unbekannte Wort sein. Das wird auch so in der Liste an Möglichkeiten des unbekannten Wortes vermerkt.

Nachdem nun durch dieses Verfahren alle Wörter eindeutig zugeteilt wurden, kann der Satz gebildet. Dazu wird jedes gesuchte Wort durch das zugehörige gefundene Wort ersetzt. Da es möglich ist, dass im gesamten Satz ähnliche Zeichenketten vorhanden sind, wird nur das erste Auftreten des zu ersetzenden Wortes ersetzt. Durch dieses Verfahren bleiben alle Satzzeichen erhalten.

Beispiele

raetsel0.txt	oh je, was für eine arbeit!
raetsel1.txt	Am Anfang wurde das Universum erschaffen. Das machte viele Leute sehr wütend und wurde allenthalben als Schritt in die falsche Richtung angesehen.
raetsel2.txt	Als Gregor Samsa eines Morgens aus unruhigen Träumen erwachte, fand er sich in seinem Bett zu einem ungeheueren Ungeziefer verwandelt.
raetsel3.txt	Informatik ist der Wissenschaft von der systematischen Darstellung, Speicherung, Verarbeitung und Übertragung von Informationen, besonders der automatischen Verarbeitung mit Digitalrechnern.

Quellcode

Der vollständige Quellcode, ist in der Datei woerterAufräumen.py zu finden.

```
# Es werden beide Zeilen aus der Datei ausgelesen
content = file.readlines()

# Der gesuchte Satz wird gespeichert
satz = content[0]

# Sowie alle Wörter die eingesetzt werden können
moeglicheWoerter = content[1].replace("\n","").split(" ")

# Es wird ein Dictionary aufgestellt, in welchem alle unbekannten Wörter einzeln gespeichert werden sollen
# Key: Das gesuchte Wort
# Value: Eine Liste, in welche mögliche Einsetzungen für das unbekannte Wort gespeichert werden
unbekannteWoerter = satz.replace(".", "").replace(",","").replace("!","").replace("\n","").split(" ")
gesuchteWoerter = {}
```

```
for i in range(0,len(unbekannteWoerter)):

    # i * ' ', deswegen, da es möglich ist, dass unbekannte Wörter doppelt in einem Satz auftreten. So kann ein
    Wort klar definiert werden

    # Aufgrunddessen wird auch beim Arbeiten mit einem zu suchendem Wort dann die Methode .strip()
    verwendet, um diese Leerzeichen zu entfernen

    gesuchteWoerter[unbekannteWoerter[i] + i * " "] = []


# In diesem for-Loop werden nun für jedes Wort die Wörter gesucht, die zumindest von der Länge und den
    bekannten Buchstaben her passen.

    for gesuchtesWort in gesuchteWoerter:

        # Dafür wird durch jedes mögliche Wort iteriert und ...

        for moeglichesWort in moeglicheWoerter:

            moeglich = False

            # Geprüft, ob die Länge übereinstimmt und ...

            if(len(gesuchtesWort.strip()) == len(moeglichesWort)):

                moeglich = True

                for i in range(0,len(gesuchtesWort.strip())):

                    # Geprüft, ob die bekannten Buchstaben übereinstimmen

                    if(gesuchtesWort[i] != "_" and gesuchtesWort[i] != moeglichesWort[i]):

                        moeglich = False


            # Wenn das Wort möglich ist, wird es zu der Liste im Dictionary hinzugefügt

            if(moeglich):

                gesuchteWoerter[gesuchtesWort].append(moeglichesWort)


# Nun müssen die Wörter passend verteilt werden, sodass zu jedem unbekannten Wort ein passendes Wort
    gefunden wurde

    for gesuchtesWort in gesuchteWoerter:

        # Wörter, bei denen nur ein Wort möglich ist, wird dieses Wort direkt zugeordnet

        if(len(gesuchteWoerter[gesuchtesWort]) == 1):

            # Dazu wird den anderen unbekannten Wörtern, bei denen dieses Wort möglich ist, das Wort entzogen

            for andereGesuchteWoerter in gesuchteWoerter:
```

```
    if(andereGesuchteWoerter != gesuchtesWort and gesuchteWoerter[gesuchtesWort][0] in
gesuchteWoerter[andereGesuchteWoerter]):

        gesuchteWoerter[andereGesuchteWoerter].remove(gesuchteWoerter[gesuchtesWort][0])

    else:

        # Andernfalls wird geprüft, ob dieses unbekannte Wort das einzige unbekannte Wort ist, welches ein
mögliches Wort als Möglichkeit besitzt

        # Dann nämlich muss dieses mögliche Wort die Einsetzungen für das unbekannte Wort sein

        for moeglichkeit in gesuchteWoerter[gesuchtesWort]:

            moeglich = True

            # Dazu wird durch die anderen möglichen Wörter iteriert und geprüft, ob diese dieses mögliche Wort
besitzen

            for andereGesuchteWoerter in gesuchteWoerter:

                if(andereGesuchteWoerter != gesuchtesWort and moeglichkeit in
gesuchteWoerter[andereGesuchteWoerter]):

                    # Ist das der Fall, so ist das Wort nicht möglich

                    moeglich = False

            # Andernfalls muss dieses mögliche Wort die Einsetzung für das unbekannte Wort sein

            if(moeglich):

                # Das wird auch so gespeichert

                gesuchteWoerter[gesuchtesWort] = [moeglichkeit]

        # Jetzt wo jedes unbekannte Wort einer eindeutigen Einsetzung zugeordnet wurde, kann der Satz gebildet
werden

        # Dazu wird durch die gesuchten Wörter iteriert

        for gesuchtesWort in gesuchteWoerter:

            # Und das gesuchte Wort wird durch das gefundene Wort ersetzt

            # Da es möglich ist, dass im Satz ähnliche Zeichenketten vorhanden sind, wird nur das erste Auftreten des
zu suchenden Wortes ersetzt

            # Außerdem werden die Leerzeichen entfernt, die zur Speicherung notwendig waren

            satz = satz.replace(gesuchtesWort.strip(), gesuchteWoerter[gesuchtesWort][0], 1)

        # Der Satz kann nun ausgegeben werden

        print(satz, end="")
```