

Aufgabe 2: Vollgeladen

20. November 2021

Team-ID: 00821

Team: IaC 0.9524

Bearbeiter dieser Aufgabe: Paul Buda

Lösungsidee:

Zuerst sortiert man die Hotels nach ihrer Bewertung absteigend. Danach versucht man zuerst nur mit dem bestbewerteten Hotel vom Start bis ans Ziel zu kommen. Ist das nicht möglich, so nimmt man noch das zweit-, dritt-, viert, fünft-bestbewertete Hotel usw. hinzu.

Sobald es nun möglich ist mit höchstens vier der x bestbewerteten Hotels vom Start zum Ziel zu kommen, hat man die Möglichkeit des am niedrigsten bewerteten Hotel mit der höchsten Bewertung gefunden.

Es ist zwar garantiert werden, dass das am niedrigsten bewertete Hotel die höchstmögliche Bewertung hat, jedoch kann es passieren, dass die anderen Hotels nicht die höchstmögliche Bewertung haben.

Umsetzung:

Mittels einer For-Schleifen werden zuerst die Km-Angabe und Bewertungen aller Hotels in einer Liste gespeichert, stehen am gleichen Reisekilometer mehrere Hotels stehen, so wird nur das bestbewertete in die Liste aufgenommen.

Danach werden die Hotels absteigend nach ihrer Bewertung sortiert. Im folgenden wird wieder mittels For-Schleife erst nur das bestbewertete, dann die zwei, drei, vier, fünf, usw. bestbewerteten Hotels ausgewählt. Für jede dieser Möglichkeiten werden zuerst wieder alle Hotels entfernt, welche zwischen zwei Hotels liegen, welche maximal 360km auseinander liegen, da man diese nicht anfahren muss. Danach wird geprüft, ob zwischen allen Hotels, dem Start und Ziel maximal 360km liegen, ist das der Fall, so wird diese Möglichkeit ausgegeben.

Für die Ausgabe werden die benötigten Hotels jetzt wieder nach den Reisekilometer sortiert und dann zusammen mit ihren Bewertungen ausgegeben. Zum Schluss wird noch die Bewertung des am schlechtesten bewerteten Hotel ausgegeben.

Beispiele:

Bei Hotels 1 erzeugt das Programm folgende Ausgabe:

```
Das Hotel nach 347 km hat eine Bewertung von 2.7 Sternen!  
Das Hotel nach 687 km hat eine Bewertung von 4.4 Sternen!  
Das Hotel nach 1007 km hat eine Bewertung von 2.8 Sternen!  
Das Hotel nach 1360 km hat eine Bewertung von 2.8 Sternen!  
Das am schlechtesten bewertete Hotel hat: 2.7 Sterne
```

Es wird ausgegeben, welche vier Hotels angefahren werden müssen und dass die Bewertung des am schlechtesten Bewerteten Hotel 2.7 Sterne beträgt.

Bei Hotels 3 erzeugt das Programm folgende Ausgabe:

```
Das Hotel nach 360 km hat eine Bewertung von 1.3 Sternen!  
Das Hotel nach 717 km hat eine Bewertung von 0.3 Sternen!  
Das Hotel nach 1076 km hat eine Bewertung von 3.8 Sternen!  
Das Hotel nach 1433 km hat eine Bewertung von 1.7 Sternen!  
Das am schlechtesten bewertete Hotel hat: 0.3 Sterne
```

Hier sieht man, dass wenn die Gesamtstrecke durch 5 fast 360 ($1793 / 5 = 358.6$) ergibt, die Auswahl der Hotels ziemlich eingeschränkt ist und deshalb das Hotel mit der niedrigsten Bewertung nur 0.3 Sterne hat.

Bei Hotels 5 erzeugt das Programm folgende Ausgabe:

```
Das Hotel nach 317 km hat eine Bewertung von 5.0 Sternen!  
Das Hotel nach 636 km hat eine Bewertung von 5.0 Sternen!  
Das Hotel nach 987 km hat eine Bewertung von 5.0 Sternen!  
Das Hotel nach 1271 km hat eine Bewertung von 5.0 Sternen!  
Das am schlechtesten bewertete Hotel hat: 5.0 Sterne
```

Wenn es jedoch sehr viele (1500) Hotels gibt und die Gesamtstrecke deutlich kürzer als 1800km ist, so ist die Wahrscheinlichkeit hoch, dass man wie in dem Beispiel nur sehr gut Bewertete Hotels ausgegeben bekommt.

Eigens erfundenes Beispiel:

```
8  
1400  
250 5.0  
270 4.8  
361 5.0  
608 3.5  
674 3.2  
965 4.2  
1150 3.7  
1456 2.2
```

Ausgabe des Programms:

```
Das Hotel nach 270 km hat eine Bewertung von 4.8 Sternen!  
Das Hotel nach 608 km hat eine Bewertung von 3.5 Sternen!  
Das Hotel nach 965 km hat eine Bewertung von 4.2 Sternen!  
Das Hotel nach 1150 km hat eine Bewertung von 3.7 Sternen!  
Das am schlechtesten bewertete Hotel hat: 3.5 Sterne
```

An diesem Beispiel sieht man nun ziemlich gut, dass das Programm nicht unbedingt die höchstmögliche Bewertung für die nicht am niedrigst bewerteten Hotels nimmt. So könnte man das Hotel nach 250km mit 5 Sternen, statt des Hotels nach 270km mit 4.8 Sternen nehmen. Das Programm

sortiert das Hotel nach 250km aber aus, da zwischen dem Start und dem nächsten Hotel unter 361km liegen.

Quelltext:

```
#ließt die Datei mit den Infos zu den Hotels ein
hotels = open("hotels.txt", "r")
#speichert die Gesamtanzahl an Hotels
numberHotels = int(hotels.readline())
#speichert die Gesamtreisestrecke
totalDistance = int(hotels.readline())

#ließt alle Hotels aus der Datei ein
hotelsList = [[-1, 0]]
for r in range(numberHotels):
    hotelKm, hotelRating = hotels.readline().split()

    #wenn am gleichen Reisekilometer mehrere Hotels stehen, wird nur das
bestbewertete ausgewählt
    if(hotelsList[-1][0] == int(hotelKm)):
        hotelsList[-1][1] = max(float(hotelRating), hotelsList[-1][1])
    else:
        hotelsList.append([int(hotelKm), float(hotelRating)])

hotels.close()

#sortiert die Hotels nach deren Bewertung
hotelsList.sort(key=lambda x: x[1], reverse=True)

checkHotels = []
allCheckHotels = [0, totalDistance]
#fügt nach und nach immer ein weiteres Hotel hinzu, angefangen beim
Bestbewerteten, dann nach der Bwertung absteigend
for hotel in hotelsList:
    canUsed = True
    allCheckHotels.append(hotel[0])
    allCheckHotels.sort()
    checkHotels = allCheckHotels.copy()
    minus = 0

    #entfernt Hotel wenn das Vorherige und Nächste höchstens 360 km voneinander
entfernt sind
    for r in range(len(checkHotels) - 2):
        if(checkHotels[r + 2 - minus] - checkHotels[r - minus] <= 360):
            del checkHotels[r + 1 - minus]
            minus += 1

    #überprüft, ob alle Hotels höchstens 360km voneinander entfernt sind
    for r in range(len(checkHotels) - 1):
        canUsed = False if(checkHotels[r + 1] - checkHotels[r] > 360) else canUsed

    #wenn die bestmögliche Strecke gefunde wurde, wird diese ausgegeben
    if(canUsed and len(checkHotels) <= 6):
```

```

#ließt die Hotels welche benötigt werden mit Km Angabe und Bewertung aus
useHotels = [hotel for hotel in hotelsList if (hotel[0] in checkHotels[1:-
1]))]

#sortiert die Hotels nach km
useHotels.sort(key=lambda x: x[0])

#gibt die Hotels nacheinander aus
[print("Das Hotel nach " + str(hotel[0]) + " km hat eine Bewertung von " +
str(hotel[1]) + " Sternen!") for hotel in useHotels]
break

#gibt die Sterne, des am schlechtesten bewertete Hotel aus
print("Das am schlechtesten bewertete Hotel hat: " + str(min(hotel[1] for hotel in
useHotels)) + " Sterne")

```