

Aufgabe 5: Stadtführung

Teilnahme-ID: 68772

Team-ID: 00083

Bearbeiter/-in dieser Aufgabe:
Linus Schumann

19. November 2023

Inhaltsverzeichnis

1	Lösungsidee	2
1.1	Feststellung der Aufgabe	2
1.2	Lösungsansatz	2
2	Umsetzung	2
2.1	Allgemeines	2
2.2	Implementierung der Lösungsidee	2
2.2.1	Main-Funktion	2
2.2.2	kill_front-Funktion	3
2.2.3	kill_circle-Funktion	3
2.2.4	print_results-Funktion	3
3	Beispiele	3
3.1	Beispiel 1	3
3.2	Beispiel 2	3
3.3	Beispiel 3	4
3.4	Beispiel 4	4
3.5	Beispiel 5	4
3.6	Eigene Beispiele	5
3.6.1	Beispiel 6	5
3.6.2	Beispiel 7	5
4	Quellcode	6

1 Lösungsidee

1.1 Feststellung der Aufgabe

Grundsätzlich ist eine Tour T gegeben. Diese besteht aus einer Menge von Knoten V und einer Menge von Kanten E . Die Kanten sind dabei gerichtet und gewichtet. Die Gewichtung der Kanten gibt dabei die Distanz zwischen den Knoten an. Auch können nur die Kanten genutzt werden, die in E enthalten sind. Die Tour T ist dabei ein Kreis, der alle Knoten in V mindestens einmal enthält. Auch gibt es eine Teilmenge V' von V , die die Knoten enthält, die essenziell für die Tour sind.

Ein Knoten besteht immer aus einem Ort und einer Zeit. Der Ort gibt dabei an, wo sich der Knoten befindet und die Zeit ist für die chronologische Reihenfolge der Knoten in der Tour zuständig.

Nun soll eine Tour T' gefunden werden, die die folgenden Bedingungen erfüllt:

- T' ist ein Kreis, der alle Knoten in V' mindestens einmal enthält.
- Die Summe der Distanzen aller Kanten in T' ist minimal.
- Die Zeiten der Knoten in T' sind chronologisch aufsteigend.
- Es dürfen nur Kanten aus E genutzt werden.

1.2 Lösungsansatz

Da nur die Kanten aus E genutzt werden dürfen, ist es nicht möglich, dass die Tour nicht chronologisch aufsteigend ist. Es können außerdem nur Kreise aus T entfernt werden und der Anfang bzw. das Ende der Tour kann verändert werden.

Mein Lösungsansatz ist es nun, alle Möglichkeiten zu finden, den Startpunkt bzw. den Endpunkt zu verändern. Danach können für jede diese Möglichkeit so viele Kreise wie möglich entfernt werden. Die Kreise, die entfernt werden, sind dabei die Kreise, die keine Knoten aus V' enthalten.

Nachdem alle Möglichkeiten durchgegangen wurden, wird die Tour mit der geringsten Summe der Distanzen aller Kanten zurückgegeben.

2 Umsetzung

2.1 Allgemeines

Im Folgenden wird die Umsetzung, der in Abschnitt 1 beschriebene Lösungsidee, näher erläutert. Grundsätzlich wurde diese Idee dabei in Python, genauer gesagt in der Datei "Aufgabe_5.py" implementiert. Diese Datei befindet sich im Verzeichnis `./source/`.

Um das implementierte Programm zu starten, kann das Batch-Script genutzt werden. Dieses befindet sich im Verzeichnis `./executables/`. Eine direkte Ausführung der Python-Datei ist auch möglich, allerdings, nur wenn sie man sich im `./source/` Verzeichnis befindet.

Unter dem Verzeichnis `./beispieleingaben/` befinden sich alle in dieser Dokumentation aufgeführten Beispiele und unter dem Verzeichnis `./beispielausgaben/` befinden sich dementsprechend die gesicherten Ausgaben, die auch bei Ausführung des Scripts ausgegeben werden. Damit Letztere besser von den Beispieldaten unterschieden werden können, werden diese mit der Dateiendung `"_out.txt"` gespeichert.

2.2 Implementierung der Lösungsidee

2.2.1 Main-Funktion

Die Main-Funktion ist der Einstiegspunkt des Programms. Sie liest zuerst den Namen der Input-Datei ein und startet die Funktion `"read_file"`. Diese Funktion liest die Datei ein und gibt die Tour T zurück. Danach wird die Funktion `"kill_front"` aufgerufen und diese gibt alle Möglichkeiten zurück, den Startpunkt der Tour zu verändern. Diese Möglichkeiten werden dann alle durchgegangen und für jede Möglichkeit wird die Funktion `"kill_circle"` aufgerufen, solange bis keine Kreise mehr entfernt werden können. Dann wird am Schluss noch die Tour mit der geringsten Summe der Distanzen aller Kanten in der Liste gesucht und dann mithilfe der Funktion `"print_results"` ausgegeben.

2.2.2 kill_front-Funktion

Die Funktion "kill_front" bekommt die Tour T übergeben und gibt eine Liste von Touren zurück. Diese Liste enthält alle Möglichkeiten, den Startpunkt der Tour zu verändern.

Um diese Möglichkeiten zu berechnen, wird zuerst die Teiltour vom Start bis zum ersten Knoten aus V' berechnet und dann die Teiltour vom Ende bis zum ersten Knoten aus V' berechnet. Die Knoten aus den beiden Teiltouren werden dann kombiniert und immer, wenn die Kombination den gleichen Ort hat, wird aus der Tour T alle Knoten bis zu diesem Knoten entfernt bzw. alle Knoten ab diesem Knoten entfernt. Die so entstandenen Touren werden dann in der Liste gespeichert und am Ende zurückgegeben.

2.2.3 kill_circle-Funktion

Diese Funktion bekommt eine Tour T übergeben und löscht einen Kreis aus dieser Tour, wenn es möglich ist. Dazu wird ein Dictionary erstellt und dann jeder Knoten der Tour durchgegangen. Wenn der Knoten noch nicht im Dictionary ist, wird er hinzugefügt. Wenn der Knoten schon im Dictionary ist, wird der Kreis, der durch den Knoten entsteht, aus der Tour entfernt.

Dabei werden auch die kumulierten Distanzen der übrigen Kanten aktualisiert.

2.2.4 print_results-Funktion

Diese Funktion bekommt die beste Tour und die ursprüngliche Tour übergeben und gibt die Ergebnisse aus. Dabei wird zuerst die ursprüngliche Tour ausgegeben und dann die beste Tour. Außerdem wird bei beiden Touren die Summe der Distanzen aller Kanten ausgegeben.

3 Beispiele

Im Folgenden werden alle Ausgaben der Beispiele aufgeführt. Die Ausgaben sind dabei der Inhalt der Dateien im Verzeichnis `./beispielausgaben/`. Mein eigenes Beispiel (Beispiel 6) zeigt auch noch, was passiert, wenn die Tour keine essenziellen Punkte enthält. Das zweite eigene Beispiel (Beispiel 7) zeigt, das Verhalten bei einem einzigen essenziellen Punkt.

3.1 Beispiel 1

```

1 Original Path:
  Total distance: 2060
3 Start -> Brauerei 1613 -> Karzer 1665 -> Rathaus 1678 -> Gründungsstein 1685 ->
    ↳ Wallanlage 1690 -> Rathaus 1739 -> Euler-Brücke 1768 -> Fibonacci-Gaststätte 1820
    ↳ -> Schiefes Haus 1823 -> Theater 1880 -> Emmy-Noether-Campus 1912 -> Fibonacci-
    ↳ Gaststätte 1923 -> Hilbert-Raum 1945 -> Schiefes Haus 1950 -> Gauß-Turm 1952 ->
    ↳ Emmy-Noether-Campus 1998 -> Euler-Brücke 1999 -> Brauerei 2012

5 Best Path with new distances:
  Total distance: 1020
7 Start -> Brauerei 1613 -> Karzer 1665 -> Rathaus 1739 -> Euler-Brücke 1768 -> Fibonacci-
    ↳ Gaststätte 1820 -> Schiefes Haus 1823 -> Theater 1880 -> Emmy-Noether-Campus 1998
    ↳ -> Euler-Brücke 1999 -> Brauerei 2012

```

Listing 1: Ausgabe Beispiel 1

3.2 Beispiel 2

```

1 Original Path:
  Total distance: 2060
3 Start -> Brauerei 1613 -> Karzer 1665 -> Rathaus 1678 -> Gründungsstein 1685 ->
    ↳ Wallanlage 1690 -> Rathaus 1739 -> Euler-Brücke 1768 -> Fibonacci-Gaststätte 1820
    ↳ -> Schiefes Haus 1823 -> Theater 1880 -> Emmy-Noether-Campus 1912 -> Fibonacci-
    ↳ Gaststätte 1923 -> Hilbert-Raum 1945 -> Schiefes Haus 1950 -> Gauß-Turm 1952 ->
    ↳ Emmy-Noether-Campus 1998 -> Euler-Brücke 1999 -> Brauerei 2012

5 Best Path with new distances:

```

```

Total distance: 1020
7 Start -> Brauerei 1613 -> Karzer 1665 -> Rathaus 1739 -> Euler-Brücke 1768 -> Fibonacci-
    ↳ Gaststätte 1820 -> Schiefes Haus 1823 -> Theater 1880 -> Emmy-Noether-Campus 1998
    ↳ -> Euler-Brücke 1999 -> Brauerei 2012

```

Listing 2: Ausgabe Beispiel 2

3.3 Beispiel 3

```

1 Original Path:
  Total distance: 4560
3 Start -> Talstation 1768 -> Wäldle 1805 -> Mittlere Alp 1823 -> Observatorium 1833 -> Wä
    ↳ ldle 1841 -> Bergstation 1866 -> Observatorium 1874 -> Piz Spitz 1898 ->
    ↳ Panoramasteg 1912 -> Bergstation 1928 -> Ziegenbrücke 1935 -> Panoramasteg 1952 ->
    ↳ Ziegenbrücke 1979 -> Talstation 2005

5 Best Path with new distances:
  Total distance: 2780
7 Start -> Talstation 1768 -> Wäldle 1841 -> Bergstation 1866 -> Observatorium 1874 -> Piz
    ↳ Spitz 1898 -> Panoramasteg 1952 -> Ziegenbrücke 1979 -> Talstation 2005

```

Listing 3: Ausgabe Beispiel 3

3.4 Beispiel 4

```

1 Original Path:
  Total distance: 3200
3 Start -> Blaues Pferd 1523 -> Alte Mühle 1544 -> Marktplatz 1549 -> Zeughaus 1559 ->
    ↳ Marktplatz 1562 -> Springbrunnen 1571 -> Dom 1596 -> Bogenschütze 1610 ->
    ↳ Marktplatz 1622 -> Ruhiges Eck 1625 -> Frauentor 1636 -> Ruhiges Eck 1651 ->
    ↳ Springbrunnen 1675 -> Bogenschütze 1683 -> Schnecke 1698 -> Fischweiher 1710 ->
    ↳ Reiterhof 1728 -> Schnecke 1742 -> Schmiede 1765 -> Große Gabel 1794 -> Schnecke
    ↳ 1829 -> Europapark 1852 -> Große Gabel 1874 -> Fingerhut 1917 -> Stadion 1934 ->
    ↳ Marktplatz 1962 -> Baumschule 1974 -> Polizeipräsidium 1991 -> Blaues Pferd 2004

5 Best Path with new distances:
  Total distance: 1640
7 Start -> Marktplatz 1562 -> Springbrunnen 1571 -> Dom 1596 -> Bogenschütze 1683 ->
    ↳ Schnecke 1698 -> Fischweiher 1710 -> Reiterhof 1728 -> Schnecke 1829 -> Europapark
    ↳ 1852 -> Große Gabel 1874 -> Fingerhut 1917 -> Stadion 1934 -> Marktplatz 1962

```

Listing 4: Ausgabe Beispiel 4

3.5 Beispiel 5

```

1 Original Path:
  Total distance: 5000
3 Start -> Gabelhaus 1638 -> Burgruine 1654 -> Labyrinth 1667 -> Hängepartie 1672 ->
    ↳ Hexentanzplatz 1681 -> Gabelhaus 1699 -> Hexentanzplatz 1703 -> Eselsbrücke 1711
    ↳ -> Dreibannstein 1724 -> Alte Wache 1733 -> Palisadenhaus 1740 -> Dreibannstein
    ↳ 1752 -> Schmetterling 1760 -> Dreibannstein 1781 -> Märchenwald 1793 -> Fuchsbau
    ↳ 1811 -> Torfmoor 1817 -> Gartenschau 1825 -> Märchenwald 1840 -> Eselsbrücke 1855
    ↳ -> Heimatmuseum 1863 -> Eselsbrücke 1877 -> Reiterdenkmal 1880 -> Riesenrad 1881
    ↳ -> Hochsitz 1885 -> Gartenschau 1898 -> Riesenrad 1902 -> Dreibannstein 1911 ->
    ↳ Olympisches Dorf 1924 -> Haus der Zukunft 1927 -> Stellwerk 1931 -> Dreibannstein
    ↳ 1938 -> Stellwerk 1942 -> Labyrinth 1955 -> Gauklerstadl 1961 -> Planetarium 1971
    ↳ -> Känguruhfarm 1976 -> Balzplatz 1978 -> Dreibannstein 1998 -> Labyrinth 2013 ->
    ↳ CO2-Speicher 2022 -> Gabelhaus 2023

5 Best Path with new distances:
  Total distance: 2010

```

```

7 Start -> Gabelhaus 1699 -> Hexentanzplatz 1703 -> Eselsbrücke 1711 -> Dreibannstein 1752
  ↳ -> Schmetterling 1760 -> Dreibannstein 1911 -> Olympisches Dorf 1924 -> Haus der
  ↳ Zukunft 1927 -> Stellwerk 1942 -> Labyrinth 1955 -> Gauklerstadl 1961 ->
  ↳ Planetarium 1971 -> Känguruhfarm 1976 -> Balzplatz 1978 -> Dreibannstein 1998 ->
  ↳ Labyrinth 2013 -> CO2-Speicher 2022 -> Gabelhaus 2023

```

Listing 5: Ausgabe Beispiel 5

3.6 Eigene Beispiele

3.6.1 Beispiel 6

```

1 18
  Brauerei,1613, ,0
3 Karzer,1665, ,80
  Rathaus,1678, ,150
5 Gründungsstein,1685, ,310
  Wallanlage,1690, ,410
7 Rathaus,1739, ,500
  Euler-Brücke,1768, ,680
9 Fibonacci-Gaststätte,1820, ,710
  Schiefes Haus,1823, ,830
11 Theater,1880, ,960
  Emmy-Noether-Campus,1912, ,1090
13 Fibonacci-Gaststätte,1923, ,1180
  Hilbert-Raum,1945, ,1300
15 Schiefes Haus,1950, ,1400
  Gauß-Turm,1952, ,1450
17 Emmy-Noether-Campus,1998, ,1780
  Euler-Brücke,1999, ,1910
19 Brauerei,2012, ,2060

```

Listing 6: Eingabe Beispiel 6

```

1 There was no essential node in this path!

```

Listing 7: Ausgabe Beispiel 6

3.6.2 Beispiel 7

```

1 18
  Brauerei,1613, ,0
3 Karzer,1665, ,80
  Rathaus,1678, ,150
5 Gründungsstein,1685, ,310
  Wallanlage,1690, ,410
7 Rathaus,1739, ,500
  Euler-Brücke,1768, ,680
9 Fibonacci-Gaststätte,1820, ,710
  Schiefes Haus,1823, ,830
11 Theater,1880, ,960
  Emmy-Noether-Campus,1912,X,1090
13 Fibonacci-Gaststätte,1923, ,1180
  Hilbert-Raum,1945, ,1300
15 Schiefes Haus,1950, ,1400
  Gauß-Turm,1952, ,1450
17 Emmy-Noether-Campus,1998, ,1780
  Euler-Brücke,1999, ,1910
19 Brauerei,2012, ,2060

```

Listing 8: Eingabe Beispiel 7

```

1 Original Path:
  Total distance: 2060
3 Start -> Brauerei 1613 -> Karzer 1665 -> Rathaus 1678 -> Gründungsstein 1685 ->
    ↳ Wallanlage 1690 -> Rathaus 1739 -> Euler-Brücke 1768 -> Fibonacci-Gaststätte 1820
    ↳ -> Schiefes Haus 1823 -> Theater 1880 -> Emmy-Noether-Campus 1912 -> Fibonacci-
    ↳ Gaststätte 1923 -> Hilbert-Raum 1945 -> Schiefes Haus 1950 -> Gauß-Turm 1952 ->
    ↳ Emmy-Noether-Campus 1998 -> Euler-Brücke 1999 -> Brauerei 2012

5 Best Path with new distances:
  Total distance: 0
7 Start -> Brauerei 2012

```

Listing 9: Ausgabe Beispiel 7

4 Quellcode

```

1 from copy import deepcopy

3 def read_file(filename: str):
    inp = []
5     with open('../beispieleingaben/'+filename+'.txt', encoding='utf-8') as f:
        for line in f.readlines():
7             line = line.strip()
            if line.isnumeric():
9                 continue
            splitted = line.split(",")
11            inp.append([splitted[0], int(splitted[1]), splitted[2] == "X", int(splitted
    ↳ [3])])
        return inp

13 def kill_circle(inpList):
15     seen = {}
    for i in range(len(inpList)):
17         if inpList[i][0] in seen:
            front = inpList[0:seen[inpList[i][0]]]
19             dif = inpList[i][3] - inpList[seen[inpList[i][0]]][3]
            last = inpList[i:]
21             for x in last:
                x[3] -= dif
23             front += last
            return front, True
25         if inpList[i][2]:
            seen = {}
27         seen[inpList[i][0]] = i
    return inpList, False

29 def kill_front(inp):
31     inpList = []
    front = []
33     back = []
    for i in range(len(inp)):
35         if inp[i][2]:
            front = inp[0:i+1]
37             break
    for i in reversed(range(len(inp))):
39         if inp[i][2]:
            back = inp[i:]
41             break
    for i,f in enumerate(front):
43         for j,b in enumerate(back):
            if f[0] == b[0]:
45             inpList.append(deepcopy(inp[i:len(inp)-len(back)+j+1]))
            dif = inpList[-1][0][3]
47             for i in range(len(inpList[-1])):
                inpList[-1][i][3] -= dif
49     return inpList

51 def print_path(inp, f):

```

```

print("Total distance: " + str(inp[-1][3]))
53 f.write("Total distance: " + str(inp[-1][3]) + "\n")
print("Start", end="")
55 f.write("Start")
for x in inp:
57     print(" -> ", end="")
    f.write(" -> ")
59     print(x[0] + " " + str(x[1]), end="")
    f.write(x[0] + " " + str(x[1]))
61
def print_results(filename, inp, bestinp):
63     with open('../beispielausgaben/' + filename + '_out.txt', 'w', encoding='utf-8') as f:
        print("Original Path: ")
65         f.write("Original Path: \n")
        print_path(inp, f)
67         print("\n\nBest Path with new distances: ")
        f.write("\n\nBest Path with new distances: \n")
69         print_path(bestinp, f)

71 def main():
    filename = input("What file? -> ")
73     inp = read_file(filename)
    inpList = kill_front(inp)
75     for i in range(len(inpList)):
        run = True
77         while run:
            inpList[i], run = kill_circle(inpList[i])
79     if len(inpList) == 0:
        print("There was no essential node in this path!")
81         open('../beispielausgaben/' + filename + '_out.txt', 'w', encoding='utf-8').write("
↪ There was no essential node in this path!")
        return
83     bestinp = inpList[0]
    for x in inpList:
85         if x[-1][3] < bestinp[-1][3]:
            bestinp = x
87     print_results(filename, inp, bestinp)

89 if __name__ == '__main__':
    main()

```
