

Programmierpraktikum für
Datenstrukturen und Algorithmen
SS 2023

Aufgabe 3: Kürzeste Wege (100 Punkte)

Schreiben Sie ein Programm, das eine Textdatei einliest, die einen gerichteten Graphen in Form von Adjazenzlisten mit gewichteten Kanten enthält. Kantengewichte sind aus dem Intervall $[-10000, 10000]$, Zyklen mit negativem Gesamtgewicht kommen in keiner Eingabe vor. Das Eingabeformat wird unten beschrieben. Das Programm soll für die Eingabe das All-Pairs-Shortest-Path (APSP) Problem lösen und die Lösung in Form eines vollständigen gerichteten Graphen mit Kantengewichten=Entfernung ausgeben. Das Ausgabeformat wird ebenfalls unten beschrieben. Implementieren Sie den Floyd-Warshall Algorithmus.

Eingabeformat:

```
<Kommentarzeile>
n = <ganze Zahl>
<nr> : <Adjazenzliste>
...
<nr> : <Adjazenzliste>
```

Dabei ist `<Kommentarzeile>` eine Zeile, die mit einem `#` beginnt.

`n = ...` gibt die Anzahl der Knoten des Graphen an. Es folgen die Adjazenzlisten der Startknoten `<nr>`, die durch `<Leerzeichen>:<Leerzeichen>` vom Startknoten getrennt sind.

Eine `<Adjazenzliste>` ist eine durch ein einzelnes Leerzeichen getrennte Liste von `<Kanten>`, eine `<Kante>` besteht aus `<nr>w<gewicht>`, wobei `<nr>` eine Knotennummer ist und `<gewicht>` das Kantengewicht als ganze Zahl ist. Knotennummer und Kantengewicht werden durch das Zeichen `w` getrennt.

Ausgabeformat:

Die Ausgabe erfolgt als gerichteter Graph ohne Kommentarzeilen entsprechend dem Eingabeformat.

Das Kantengewicht eines Knoten j in der Adjazenzliste von Knoten i ist jetzt die Entfernung von i zu j im Eingabegraphen (vgl. Beispielinstantz 1 auf nächster Seite).

Ist ein Knoten j von einem Knoten i in G nicht erreichbar, so ist die Distanz $\delta(i, j) = \infty$ und die Kante wird nicht ausgegeben (vgl. Beispielinstantz 2 auf nächster Seite).

Die Adjazenzlisten werden sortiert nach Ausgangsknoten jeweils als eine Zeile ausgegeben. Die einzelnen Kanten innerhalb einer Adjazenzliste werden ebenfalls sortiert (nach Zielknoten) ausgegeben.

Da Ihre Lösung textuell mit einer Musterlösung verglichen wird, ist die strikte Beachtung des Ausgabeformats notwendig. Bei Fragen wenden Sie sich per Email an obelix@upb.de.

Das Programm wird auf dem Rechner `dua.cs.upb.de` automatisch evaluiert und dazu in einer Shell gestartet. Eingaben werden durch Argumente an das Programm übergeben. Kompilierbare Programme werden vor dem Start durch ein Shell-Kommando neu kompiliert. Der Name der zu lesenden Datei wird dem Programm als Argument übergeben. Ihre Lösung laden Sie bitte als gezipptes Archiv in PANDA hoch. Das Archiv muß enthalten:

- 1) Sämtliche Quellcodes Ihres Programms
- 2) Eine Informationsdatei mit Namen `info.txt`, die die Emailadressen, Matrikelnummern und Namen der Programmierer im Team, den Befehl zum Compilieren des Programms sowie den Befehl zum Starten des Programms enthält.

Beispielinstanz 1:

Eingabe:

```
# negative Kantengewichte
n = 3
0 : 1w11 2w-3
1 : 2w6
2 : 1w-5 0w9
```

Ausgabe:

```
n = 3
0 : 0w0 1w-8 2w-3
1 : 0w15 1w0 2w6
2 : 0w9 1w-5 2w0
```

Beispielinstanz 2:

Eingabe:

```
# unerreichbare Knoten
n = 4
0 : 1w27 3w91
1 : 2w36
2 : 0w10
3 : 3w29
```

Ausgabe:

```
n = 4
0 : 0w0 1w27 2w63 3w91
1 : 0w46 1w0 2w36 3w137
2 : 0w10 1w37 2w0 3w101
3 : 3w0
```

Weitere Instanzen mit Lösungen finden Sie auf PANDA.

Warnungen:

- Kopieren Sie keine Beispielprogramme und keine Programme ihrer StudienkollegInnen. Die Abgaben werden auf Plagiate untersucht. Plagiate werden mit 0 Punkten bewertet.
- Sorgen Sie dafür, dass Sie niemals in zwei Programmierteams einer Aufgabe genannt werden. Mehrfach genannte Autoren werden mit 0 Punkten bewertet.
- Verwenden Sie keine Graphbibliotheken. Eine solche Verwendung führt zu einer Bewertung mit 0 Punkten.
- Nutzen Sie keine in der von Ihnen verwendeten Programmiersprache vordefinierten Bibliotheken. Das Verfahren für die Berechnung der kürzesten Wege muß selbst implementiert sein. Anderenfalls wird ihre Abgabe mit 0 Punkten bewertet.