

Einführung in die Diskrete Mathematik

3. Programmierübung

Implementieren Sie den SUKZESSIVE-KÜRZESTE-WEGE-ALGORITHMUS, um das ZUORDNUNGSPROBLEM zu lösen. Das Programm soll Laufzeit $O(n^3)$ haben, wobei $n = |V(G)|$ sei.

Das Programm muss in C++ geschrieben sein. Man kann man zum Einlesen und Speichern der Graphen die Klasse `Graph` (eventuell modifiziert) aus dem Buch “Algorithmische Mathematik” (Hougardy/Vygen) benutzen. Diese finden Sie (wie alle Programme aus diesem Buch) hier:

<http://www.or.uni-bonn.de/~hougardy/alma/alma2nd.html>

Außerdem dürfen Sie bei Bedarf Teile der C++-Standardbibliothek einbinden. Andere externe Bibliotheken dürfen nicht verwendet werden.

Eingabeformat:

Die Einträge der Datei sind ausschließlich ganze Zahlen. Sie können voraussetzen, dass die Summe der Absolutbeträge aller Zahlen in der Eingabe kleiner als 2^{31} ist. In der ersten Zeile steht eine einzelne positive gerade Zahl n , welche die Anzahl der Knoten angibt. Die Knoten werden von 0 bis $n - 1$ durchnummeriert. Jede folgende Zeile kodiert genau eine Kante. Die ersten beiden Einträge einer Zeile sind die Nummern der Endknoten der Kante (wobei die Kante vom jeweils ersten angegebenen Knoten zum zweiten gerichtet sei). Die erste Nummer liegt dabei in $\{0, \dots, \frac{n}{2} - 1\}$ und die zweite in $\{\frac{n}{2}, \dots, n - 1\}$. Der dritte Eintrag in der Zeile gibt die Kosten der Kante an.

Parallele Kanten kommen in den Instanzen nicht vor. Der bipartite Graph muss nicht vollständig sein.

Ausgabeformat: Das Programm muss in der ersten Zeile der Ausgabe die Kosten der berechneten Lösung ausgeben. Danach folgen $\frac{n}{2}$ weitere Zeilen, die jeweils genau zwei Zahlen enthalten und eine Kante der berechneten Lösung kodieren. Die Nummern in der jeweiligen Zeile geben dabei die Nummern der Endknoten an (wobei die kleinere Nummer zuerst stehen soll).

Beispiel: Eine Eingabedatei für einen Graphen mit sechs Knoten und sechs Kanten kann so aussehen:

```
6
2 5 37
0 3 12
0 4 -14
1 4 8
2 3 4
1 5 2
```

Die Ausgabe der Programms kann dann so aussehen (die letzten drei Zeilen können anders sortiert sein):

```
-8  
0 4  
2 3  
1 5
```

Testinstanzen befinden sich auf der Übungsseite der Vorlesung:
http://www.or.uni-bonn.de/lectures/ws25/edm_uebung_ws25.html

Für diese Programmieraufgabe gibt es 20 Punkte.

Abgabe: Der Quelltext des Programms muss bis Donnerstag, 22.1.2026, 16:00 Uhr über eCampus hochgeladen sein.