

Einführung in die Diskrete Mathematik

5. Übung

1. Die Zeitsteuerungsbedingungen („timing constraints“) eines Logikchips lassen sich durch einen gerichteten Graphen G mit Kantengewichten $c : E(G) \rightarrow \mathbb{R}_+$ darstellen. Dabei entsprechen die Knoten den Speicherelementen und die Kanten gewissen durch die kombinatorische Logik definierten Wegen, während die Gewichte (Schätzungen der) Signallaufzeiten entsprechen. Eine Teilaufgabe des Chip-Designs ist es, einen optimalen Takt-Zeitplan zu finden, d.h. eine möglichst kleine Zahl T und eine Abbildung $a : V(G) \rightarrow \mathbb{R}$ mit der Eigenschaft, dass $a(v) + c((v, w)) \leq a(w) + T$ für alle $(v, w) \in E(G)$. Hierbei ist T die Zykluszeit des Chips, und $a(v)$ bzw. $a(v) + T$ sind die Startzeit bzw. die späteste zulässige Ankunftszeit des Signals in v .

- Reduzieren Sie das Problem, das optimale T zu finden, auf das MINIMUM-MEAN-CYCLE-PROBLEM.
- Zeigen Sie, wie man die Zahlen $a(v)$ einer optimalen Lösung effizient bestimmen kann.
- Typischerweise sind einige der Zahlen $a(v)$ vorab festgelegt. Man zeige, wie man in diesem Fall das Problem lösen kann. (2+2+2 Punkte)

2. Man beweise den Zirkulationssatz von Hoffman: Gegeben seien ein gerichteter Graph G und untere bzw. obere Schranken $l, u : E(G) \rightarrow \mathbb{R}_+$ mit $l(e) \leq u(e)$ für alle $e \in E(G)$. Es gibt genau dann eine Zirkulation $f : E(G) \rightarrow \mathbb{R}_+$ mit $l(e) \leq f(e) \leq u(e)$ für alle $e \in E(G)$, wenn

$$\sum_{e \in \delta^-(X)} l(e) \leq \sum_{e \in \delta^+(X)} u(e) \quad \text{für alle } X \subseteq V(G) \text{ gilt.}$$

(5 Punkte)

3. Sei G ein gerichteter Graph mit konservativen Gewichten $c : E(G) \rightarrow \mathbb{R}$. Es seien $s, t \in V(G)$ zwei Knoten, so dass t von s aus erreichbar ist. Außerdem gelte für jede Kante $e \in E(G)$, dass $\text{dist}_{G-e,c}(s, t) = \text{dist}_{G,c}(s, t)$. Zeigen Sie, dass es dann zwei kantendisjunkte kürzeste s - t -Wege in (G, c) gibt. (3 Punkte)

4. Schreiben Sie ein Programm, dem ein gerichteter Graph G und zwei Knoten $s, t \in V(G)$ mit $s \neq t$ und $(s, t) \notin E(G)$ übergeben werden und das berechnet, wie viele paarweise intern disjunkte s - t -Wege es in G gibt. Ihr Programm soll eine Laufzeit von $O(|V(G)||E(G)|)$ haben. Nähere Angaben zur Programmieraufgabe finden Sie auf der Rückseite. (12 Punkte)

Zur Programmieraufgabe:

Das Programm muss in C oder C++ geschrieben sein. Es wird empfohlen, C++ zu verwenden. In diesem Fall kann man zum Einlesen und Speichern der Graphen die Klasse `Graph` aus der Vorlesung "Algorithmische Mathematik I" aus dem Wintersemester 2016/2017 verwenden.

Alle Datenstrukturen, die in dieser Vorlesung vorgestellt wurden, finden Sie hier:

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

Sie können alle diese Programme und Datenstrukturen verwenden. Außerdem dürfen Sie die STL verwenden, aber keine weiteren externen Bibliotheken.

Eingabeformat: In der ersten Zeile steht eine einzelne natürliche Zahl (größer als 1 und kleiner als 2^{31}), welche die Anzahl der Knoten angibt. Wir nehmen an, dass, wenn wir n Knoten haben, die Knoten von 0 bis $n - 1$ durchnummeriert sind. Der Knoten s hat die Nummer 0, der Knoten t die Nummer 1. Jede weitere Zeile spezifiziert genau eine Kante. Die beiden Einträge einer Zeile sind zwei verschiedene nichtnegative ganze Zahlen, welche die Nummern der Endknoten der Kante sind. Die Kanten sind vom ersten in der Zeile kodierten Knoten zum zweiten orientiert. Sie können annehmen, dass der Graph keine parallelen Kanten enthält. Paare gegenläufiger Kanten können aber vorkommen. Die Sortierung der Kanten in der Eingabedatei kann beliebig sein.

Ausgabeformat: In der ersten Zeile der Ausgabe soll die Zahl der paarweise intern disjunkten s - t -Wege stehen. Jeder weitere Zeile kodiert genau einen solchen Weg. In einer Zeile sollen dabei die Nummern der Knoten des Weges bei Traversierung von s nach t (aber ohne die Nummern 0 für s und 1 für t) angegeben werden.

Beispiel: Eine Eingabedatei für einen Graphen mit 5 Knoten und 7 Kanten kann so aussehen:

```
5
2 1
3 2
1 4
0 4
2 4
4 1
0 3
```

Die Ausgabe der Programms muss dann, bis auf Vertauschung der zweiten und dritten Zeile, so aussehen:

```
2
3 2
4
```

Das Programm muss korrekt arbeiten und ohne Fehlermeldung kompiliert werden können. Der Code muss auf einem gängigen Linuxsystem funktionieren.

Abgabe: Der Quelltext des Programms muss bis 28. November, 16:15 Uhr per E-Mail beim jeweiligen Tutor eingegangen sein. Außerdem ist bis zu diesem Zeitpunkt ein Ausdruck des Quelltextes zusammen mit den Theorieaufgaben abzugeben.

Hinweis der Fachschaft Mathematik: Die Fachschaft Mathematik feiert am 23.11. ihre Matheparty in der N8schicht. Der Vorverkauf findet am Mo. 20.11., Di. 21.11. und Mi 22.11. in der Mensa Poppelsdorf statt. Alle weitere Infos auch auf fsmath.uni-bonn.de.