Wintersemester 2017/18
Prof. Dr. J. Vygen
Dr. U. Brenner

Forschungsinstitut für Diskrete Mathematik
Universität Bonn

Einführung in die Diskrete Mathematik
1. Übung

1. Sei G ein gerichteter Graph, der für alle $v, w \in V(G)$ genau einen v-w-Weg enthält. Zeigen Sie, dass G eulersch ist. (4 Punkte)

2. Welche ungerichteten Graphen haben eine Orientierung G, für die Bedingung (a) bzw. (b) gilt?

   (a) $|\delta^+(v)| - |\delta^-(v)| \in \{-1, 0, 1\}$ für alle $v \in V(G)$;

   (b) $|\delta^+(X)| - |\delta^-(X)| \in \{-1, 0, 1\}$ für alle $X \subseteq V(G)$.

   Die Antworten zu (a) und (b) können natürlich verschieden ausfallen. Begründen Sie sie! (3+3 Punkte)

3. An einem Tennisturnier nehmen genau n Spieler teil. Jeder spielt genau einmal gegen jeden anderen. Es gibt keine Unentschieden. Am Ende soll eine Rangliste der Spieler aufgestellt werden, d.h. eine Nummerierung mit $s_1, \ldots, s_n$, und zwar so, dass s_i gegen s_{i+1} gewonnen hat für alle $i = 1, \ldots, n-1$.

   (a) Beweisen Sie, dass dies immer möglich ist.

   (b) Finden Sie einen Algorithmus, der die Ergebnisliste als Eingabe bekommt und eine solche Rangliste in $O(n^k)$ Rechenschritten bestimmt, wobei k eine Konstante sei. Wie klein kann k gewählt werden? (2+3 Punkte)

4. Schreiben Sie ein Programm, das zu einem gegebenen ungerichteten Graphen entweder einen eulerschen Spaziergang berechnet oder entscheidet, dass keiner existiert. Das Programm soll lineare Laufzeit erreichen. Nähere Angaben zur Programmieraufgabe finden Sie auf der Rückseite. (10 Punkte)

Abgabe der Theoriaufgaben: Dienstag, den 17.10.2017, **vor** der Vorlesung.

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 0 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. 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 Reihenfolge der Knotennummern in der Zeile spielt keine Rolle. Es können parallele Kanten vorkommen. Die Sortierung der Kanten in der Eingabedatei kann beliebig sein.
Ausgabeformat: Wenn der Graph nicht eulersch ist, soll eine entsprechende Meldung ausgegeben werden.
Wenn der Graph eulersch ist, soll die Knotenfolge eines eulerschen Spaziergangs ausgegeben werden, wobei in jeder Zeile der Ausgabe der Index eines Knotens stehen soll.
Beispiel: Eine Eingabedatei für einen Graphen mit 5 Knoten und 7 Kanten kann so aussehen:

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

Die Ausgabe der Programms kann dann so aussehen:

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

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 24. Oktober, 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.
Testinstanzen befinden sich auf der Seite

`http://www.or.uni-bonn.de/lectures/ws17/edm_17_uebung.html`

Sollten weitere Hinweise zu der Programmierübung notwendig sein, werden diese ebenfalls auf dieser Homepage bekanntgegeben.