Claus/Lewandowski	**Theoretische Informatik I**	Wintersemester 1998/99
Übungsblatt 1	(Softwaretechnik)	Abgabe: 2. November

Termine:

- Gruppe 1 (1.026):
  Mittwoch, 8.30 - 10.00 Uhr: 4.11., 18.11., 2.12., 16.12., 13.1., 27.1., 10.2.
- Gruppe 2 (1.040):
  Donnerstag, 13.15 - 14.45 Uhr: 5.11., 19.11., 3.12., 17.12., 14.1., 28.1., 11.2.
- Gruppe 3 (1.040):
  Donnerstag, 13.15 - 14.45 Uhr: 12.11., 26.11., 10.12., 7.1., 21.1., 4.2., 11.2.

Abgabe der Übungsblätter vor der Vorlesung oder im Kasten, Foyer, 1. Etage.
Die Voraussetzungen für den Erwerb eines Übungsscheins sind:

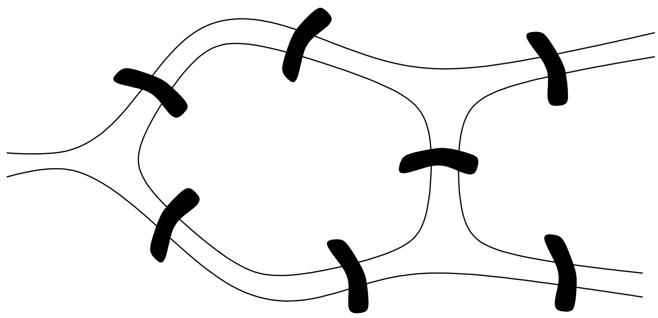
- Regelmäßige Teilnahme an den Übungen.
- Jede(r) muß mindestens zweimal eine Lösung vorgetragen haben.
- 50% der maximalen Punktzahl (160 Punkte – verteilt auf 7 Übungsblätter und 7 Multiple Choice Tests) sind zu erreichen. Die Multiple Choice Tests finden während der Vorlesung statt.

Die Übungsblätter und Termine sind auch im WWW zugänglich:

http://www.informatik.uni-stuttgart.de/ifi/fk/uebungen/ws9899/ti2st.html

Aufgabe 1: (2+3+1 Punkte)

Das Königsberger Brückenproblem besteht in der Frage, ob es eine Möglichkeit gibt, einen Spaziergang durch Königsberg zu machen, so daß jede Brücke genau einmal überquert wird. Ein solcher Weg durch einen (ungerichteten) Graphen, bei dem jede Kante genau einmal vorkommt, heißt Eulerscher Weg.



a) Gibt es einen Eulerschen Weg über die Brücken in Königsberg? Welche Bedingung muß ein Graph haben, damit solch ein Weg existiert? Wie lautet die Bedingung, wenn der Graph gerichtet ist und die Kanten somit nur in einer Richtung durchlaufen werden dürfen?

b) Entwickeln Sie einen Algorithmus, der zu einem gegebenem ungerichteten Graphen G und Startknoten v entscheidet, ob ein Eulerscher Weg existiert und ihn ggf. konstruiert. Wie unterscheidet sich der Algorithmus bei gerichteten Graphen?

c) Welche Darstellung des Eingabegraphen eignet sich für Ihren Algorithmus am besten (für die ungerichtete, für die gerichtete Version)? (Begründung!)

Aufgabe 2: (1+1 Punkte)

Zeigen oder widerlegen Sie (mit Beweis!):

a) Es ist entscheidbar, ob Ihr Algorithmus aus obiger Aufgabe für alle endlichen Graphen anhält.

b) Es gibt einen Algorithmus, der entscheidet, ob ein Programm Eulersche Wege berechnet.

Aufgabe 3: (2+2+2 Punkte)

Die Zahlen 0 bis $2^n - 1$ können in binärer Darstellung kodiert werden durch $\gamma : \{0, 1, \dots, 2^n - 1\} \to \{0,1\}^n$ mit

$$\gamma(i) = \text{binäre Darstellung von } i \text{ mit führenden Nullen.}$$

Bei dieser Darstellung unterscheidet sich der Code $\gamma(i)$ und $\gamma(i+1)$ u.U. an allen Stellen (z.B. für $n = 4$ bei $\gamma(7) = 0111$ und $\gamma(8) = 1000$).

a) Existiert eine Kodierung δ der Zahlen 0 bis $2^n - 1$ mit n (Binär-)Stellen so, daß sich $\delta(i)$ und $\delta(i+1)$ nur an einer Stelle unterscheiden? Formulieren Sie dieses Problem für $n = 3$ als Graphenproblem und finden Sie eine geeignete Kodierung $\delta_3 : \{0, \dots, 7\} \to \{0,1\}^3$.

b) Versuchen Sie aus der Kodierung δ_3 eine geeignete Kodierung δ_4 für das Problem mit $n = 4$ herzuleiten. Ist das Problem für alle n lösbar?

c) Wie verändert sich das Problem, wenn zusätzlich gefordert wird, daß $\delta_n(0) = 0^n$ und $\delta_n(2^n - 1) = 1^n$ gelten soll? Ist es dann für alle n lösbar?

Aufgabe 4: (2+2 Punkte)

Sie wollen zu einer Party sieben Ihrer Freunde (Antje, Bernd, Christine, Dirk, Erica, Fabian und Gudrun) einladen, leider vertragen sich nicht alle acht untereinander, folgende Paare von Personen vertragen sich nicht:

$$\{\{A., C.\}, \{A., E.\}, \{A., F.\}, \{C., E.\}, \{C., G.\}, \{D., G.\}, \{E., F.\}\}$$

a) Da Sie ein harmoniebedürftiger Mensch sind, wollen Sie, daß auf Ihrer Party sich alle paarweise gut verstehen. Sie wollen möglichst viele Ihrer Freunde einladen; formulieren Sie dieses als Graphenproblem und finden Sie eine Lösung, bei der möglichst viele Personen eingeladen werden können.

b) Da Sie ein geiziger Mensch sind, aber nur wenige Ihrer Freunde dadurch, daß Sie sie nicht einladen, verärgern wollen, wollen Sie, daß sich auf Ihrer Party alle paarweise nicht vertragen, in der Hoffnung, daß alle früh gehen und Sie so nicht viel für Getränke etc. bezahlen müssen; formulieren Sie dieses als Graphenproblem und finden Sie eine Lösung, bei der möglichst viel Personen eingeladen werden, die aber alle paarweise schlecht miteinander auskommen.