Algorithmische Mathematik I

Wintersemester 2017/18
Prof. Dr. Ira Neitzel
AR. Dr. Tino Ullrich

universitätbonn

Übungsblatt 10.

Abgabe am **18.12.2017 vor der Vorlesung**.

Hinweis zur Bearbeitung. Beginnen Sie mit Aufgabe 1 und der Programmieraufgabe 1. "Starker und schwacher Zusammenhang" sowie "Adjazenzlisten" und "-matrixen" werden am Mittwoch in der Vorlesung besprochen.

Aufgabe 1. (Wege)

Es sei $G = (V, E)$ mit ein einfacher ungerichteter Graph. Wir definieren die folgende Relation auf $V \times V$:

$$x \sim y \quad \text{es existiert ein } x\text{-}y \text{ Weg in } G.$$

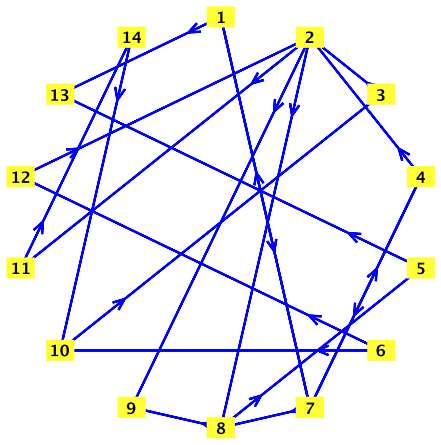
a. Zeigen Sie, dass $\sim$ eine Äquivalenzrelation ist.

b. Bestimmen Sie alle Äquivalenzklassen von $\sim$.

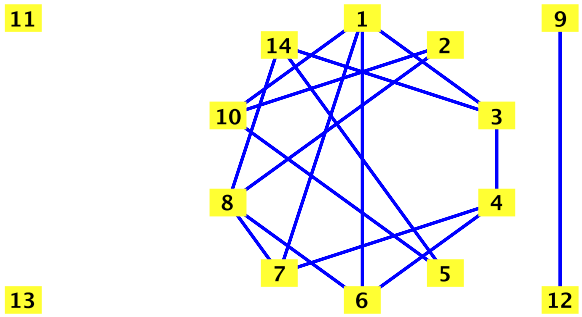
(5 Punkte)

Aufgabe 2. (Zusammenhangskomponenten)

a. Bestimmen Sie in diesem Graph alle starken Zusammenhangskomponenten.



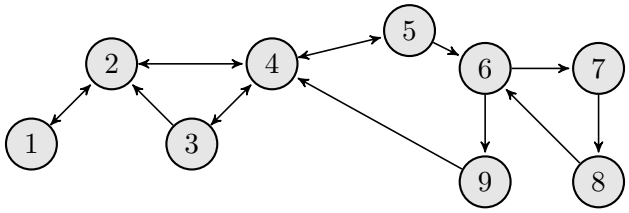
b. Bestimmen Sie in diesem Graph alle Zusammenhangskomponenten.



(3 + 2 = 5 Punkte)

Aufgabe 3. (Adjazenzmatrizen)

Wir betrachten den folgenden gerichteten Graphen, in dem ↔ für eine Doppelkante steht (d.h. Kurzform für je eine Kante *vom* Knoten und eine Kante *zum* Knoten).



a. Geben Sie alle einfachen Pfade von Knoten 1 zu Knoten 8 sowie von Knoten 8 zu Knoten 1 an.

b. Eine Tabelle der nebenstehenden Form nennt man Adjazenzmatrix, wenn als Eintrag G_{ij} jeweils 1 für "es existiert eine Kante von Knoten i zu Knoten j" und entsprechend 0 für "es existiert *keine* Kante von i zu j" steht. Geben Sie die Adjazenzmatrix für obigen Graphen an. Geben Sie auch die Adjazenzmatrix an, wenn alle Kanten *ungerichtet* wären.

	1	2	3	...	10
1					
2					
3					
⋮					
10					

c. Zeichnen Sie einen Graph mit Knoten $1, 2, \ldots, 9$ und der Adjazenzmatrix:

	1	2	3	4	5	6	7	8	9
1		1			1				
2	1		1				1		
3		1							
4					1				
5	1			1					
6			1				1		
7		1				1		1	1
8							1		
9				1			1		

Muss der Graph gerichtet gezeichnet werden oder kann er auch ungerichtet sein?

d. Leiten Sie Formeln her, die mit Hilfe der Adjazenzmatrix A bestimmt,

- ob ein Weg vom Knoten i zum Knoten j der Länge $\leq n$ existiert,
- ob ein Graph zusammenhängend ist,
- ob ein gerichteter Graph kreisfrei ist.

$(2 + 2 + 1 + 5 = 10$ Punkte)

Programmieraufgabe 1. (Summe und Produkt von Matrizen)

Implementieren Sie C/C++ Routinen, die sowohl die Summe als auch das Produkt von Matrizen $A \in \mathbb{R}^{n \times m}$

$$A = (a_{ij})_{\substack{i=1,\ldots,n \\ j=1,\ldots,m}} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}$$

realisiert. Es können nur Matrizen gleicher Dimension addiert werden. Für $A \in \mathbb{R}^{n \times m}$ und $B \in \mathbb{R}^{m \times \ell}$ (verkettete Matrizen) ist

$$A \cdot B = (c_{i,j})_{\substack{i=1,\ldots,n \\ j=1,\ldots,\ell}} \in \mathbb{R}^{n \times \ell}$$

mit

$$c_{i,j} = \sum_{k=1}^{m} a_{ik} \cdot b_{kj} \quad , \quad i = 1, ..., n, j = 1, ..., \ell\,.$$

Die Matrix vom Format $n \times m$ ist dabei als dynamisches zweidimensionales Array `double**` implementiert . Implementieren Sie folgende Routinen. Nutzen Sie die Vektoroperationen von Blatt 6.

```
1
double** init(int n, int m) {
3     double **mat;
      int i;
5     mat = new double*[n];
      for (i=0;i<n;i++) {
7         mat[i] = new double [m];
      }
9     return mat;
}
11

13 double** sum(double **mat1, double **mat2, int n , int m) {
...
15 }
```

```
17 double** = product(double **mat1, double **mat2,
   int n1, int m1, int n2, int m2) {
19 ...
   }
21
   void delete(double **mat, int n, int m) {
23  Speicher wieder freigeben (zweistufig)
   }
```

(4 Punkte)

Die Programmieraufgabe wird in der Woche vor Weihnachten bepunktet

Programmieraufgabe 2. (Adjazenzmatrix)

Wenn Graphen mittels einer Adjazenzmatrix gespeichert werden, dann kann damit sehr einfach entschieden werden, ob ein Graph gerichtet oder ungerichtet ist, (stark, schwach) zusammenhängend, oder ob er Kreise enthält (siehe Aufgabe 3 oben).

a. Implementieren Sie Funktionen in C/C++, die die untenstehenden Fragen für zwei Beispielgraphen beantwortet (`graph1.dat` und `graph2.dat` auf der Webseite). Nutzen Sie dabei die Einsichten aus Aufgabe 3 und die Implementierungen für die Programmieraufgabe 1. Welche Funktionen benötigen Sie noch?

b. Testen Sie ihre Implementierung mit den beiden Graphen auf der Webseite. Diese sind über die Adjazenzmatrixmatrix als Textdatei gegeben. Eine Routine, die Graphen in ein `int** matrix` einzulesen, finden Sie auf der Webseite implementiert in `input.h`.

c. Ihr Programm muss am Ende ausgeben ob beide Graphen:

   (a) gerichtet oder ungerichtet,

   (b) (stark, schwach) zusammenhängend sind,

   (c) Kreise beinhalten.

$(5 + 1 + 3$ Punkte)

Die Programmieraufgabe wird in der Woche vor Weihnachten bepunktet