

# Übung 8: Graphen

Abgabetermin: 12.05.2015

Name: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

Gruppe:  G1 Di 10:15

G2 Di 11:00

G3 Di 12:45

Aufgabe	Punkte	gelöst	abzugeben schriftlich Postkasten 2. Stock Science Park 3	abzugeben elektronisch	Korr.	Punkte
Aufgabe 1	8	<input type="checkbox"/>	Simulation		<input type="checkbox"/>	
Aufgabe 2	16	<input type="checkbox"/>	Java-Programm Tests und Ergebnisse	Java-Programm Tests und Ergebnisse	<input type="checkbox"/>	

## Aufgabe 1: Transitive Hülle mit dem Warshall-Algorithmus bestimmen

Berechnen Sie die transitive Hülle des Graphen mit dem Warshall-Algorithmus. Geben Sie alle Zwischenergebnisse als Adjazenzmatrix an. Markieren Sie neu hinzugekommene Kanten durch Einkreisen.

	A	B	C	D	E	F
A						
B						
C						
D						
E						
F						

↓ Spalte A

	A	B	C	D	E	F
A						
B						
C						
D						
E						
F						

↓ Spalte B

	A	B	C	D	E	F
A						
B						
C						
D						
E						
F						

↓ Spalte C

	A	B	C	D	E	F
A						
B						
C						
D						
E						
F						

↓ Spalte D

	A	B	C	D	E	F
A						
B						
C						
D						
E						
F						

↓ Spalte E

	A	B	C	D	E	F
A						
B						
C						
D						
E						
F						

↓ Spalte F

	A	B	C	D	E	F
A						
B						
C						
D						
E						
F						

Abzugeben ist (schriftlich): Simulation

**Aufgabe 2: Implementierung des Warshall-Algorithmus**

Implementieren Sie den Warshall-Algorithmus in Java. Fügen Sie dazu Ihrer Graphs-Klasse die Methode *makeTransitiveHull* hinzu. (Sie müssen für diese Übung die Methoden aus Übung 7 nicht abgeben.)

```
class Graphs {
    // ...
    public static Graph makeTransitiveHull(Graph graph) {...}
}
```

Kopieren Sie zunächst den übergebenen Graph wie in Übung 7. Erstellen Sie dann eine Adjazenzmatrix mit den Kanten im Graph. Sie können dafür entweder ein *boolean[][]*-Array verwenden oder die *BitSet*-Klasse der Java-Klassenbibliothek benutzen. Verwenden Sie als Indizes in die Matrix die Position der Knoten in der *vertices*-Liste des *Graph*-Objekts. Bestimmen Sie dann mit dem Warshall-Algorithmus schrittweise die Kanten der transitiven Hülle. Fügen Sie schließlich in Ihren kopierten Graph alle neuen Kanten in der Adjazenzmatrix ein. Geben Sie den kopierten Graph mit den zusätzlichen Kanten zurück.

Optional: für ein schöneres (grafisches) Ergebnis können Sie zwei gerichtete Kanten  $A \rightarrow B$  und  $B \rightarrow A$  zu einer ungerichteten Kante  $A-B$  zusammenfassen.

Implementierungshinweise:

- Berücksichtigen Sie, dass bei einer ungerichteten Kante zwischen A und B sowohl A direkt von B als auch B direkt von A erreichbar ist.
- Die *getEdge(start, end)*-Methode von Graph gibt *null* zurück, wenn keine Kante von *start* nach *end* existiert. Die Methode berücksichtigt auch ungerichtete Kanten von *end* nach *start*.
- Ignorieren Sie eventuell vorhandene Kantengewichte.
- Die *indexOf*-Methode von *Graph.vertices* liefert den Index eines Vertex-Objekts.

Abzugeben ist (schriftlich und elektronisch): Java-Programm und Testergebnisse