Übung 8: Graphen

Abgabetermin: 13.05.2014

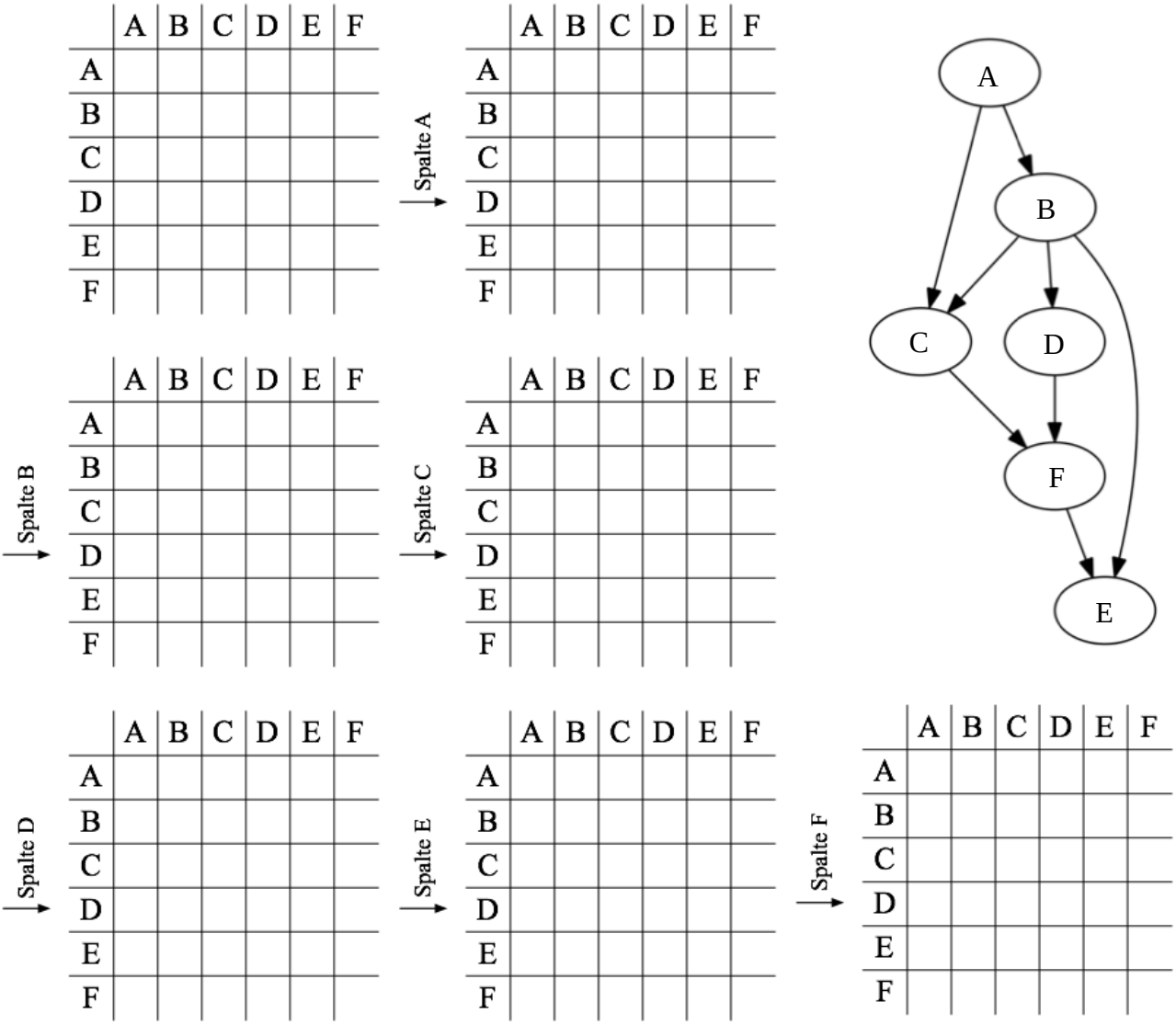
Name: _____ **Matrikelnummer:** _____

Gruppe: ☐ **G1** Di 10:15 ~~☐ **G2** Di 11:00~~ ☐ **G3** Di 12:45

Aufgabe	Punkte	gelöst	abzugeben schriftlich	abzugeben elektronisch	Korr.	Punkte
Aufgabe 1	8	☐	Simulation		☐	
Aufgabe 2	16	☐	Java-Programm Tests und Ergebnisse	Java-Programm	☐	

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.



Abzugeben ist: Simulation

Aufgabe 2: Implementierung des Warshall-Algorithmus

Implementieren Sie jetzt 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 int makeTransitiveHull(Graph graph) {...}
}
```

Ihre Implementierung soll eine Adjazenzmatrix verwenden. Sie können dafür entweder ein *boolean[][]*-Array benutzen 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. Nachdem Sie die transitive Hülle bestimmt haben, fügen Sie alle zusätzlichen Kanten zwischen Knoten ein, die nur indirekt voneinander erreichbar sind. Geben Sie die Anzahl der neu eingefügten Kanten zurück.

Optional: für ein schöneres (grafisches) Ergebnis können Sie zwei gerichtete Kanten A→B und B→A zu einer ungerichteten Kante A–B zusammenfassen.

Implementierungshinweise:

a) Berücksichtigen Sie, dass bei einer ungerichteten Kante zwischen A und B sowohl A direkt von B als auch B direkt von A erreichbar ist.

b) Ignorieren Sie eventuell vorhandene Kantengewichte.

c) Die *indexOf*-Methode von *Graph.vertices* liefert den Index eines Vertex-Objekts.

d) 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*.

Abzugeben ist: Java-Programm und Testergebnisse