

# Algorithmische Graphentheorie für Informatiker

## Labor 3

### (Aufgabe 1)

**(2 Punkte)** Implementieren Sie einen neuen Konstruktor für die Klasse Graph mit einem einzigen Parameter, einen Dateinamen, der die Eingabedaten für einen ungerichteten Graphen enthält. Die Datei sollte die folgende Struktur haben:

#### 1. Kantenliste

Die erste Zeile enthält zwei Ganzzahlen,  $n$  und  $m$ , wobei  $n$  die Anzahl der Knoten und  $m$  die Anzahl der Kanten darstellt. Die folgenden  $m$  Zeilen repräsentieren die Kanten ( $\text{Knoten1} \text{ Knoten2}$ ), wobei jede Zeile zwei durch ein Leerzeichen getrennte Ganzzahlen enthält, die die verbundenen Knoten angeben.

#### 2. Adjazenzmatrizen

Die erste Zeile enthält eine Ganzzahl  $n$ , die die Anzahl der Knoten darstellt. Die nächsten  $n$  Zeilen bestehen aus der Adjazenzmatrix des Graphen, wobei jede Zeile  $n$  Ganzzahlwerte enthält.

Die Methode sollte die Daten aus der Datei lesen und sie in einer geeigneten Datenstruktur speichern, die den Graphen repräsentiert.

### (Aufgabe 2)

a) **(2 Punkte)** Implementieren Sie eine Methode `void BFS(int x)` welche eine Breitensuche aus dem Knoten  $x$  ausführt und die besuchten Knoten auf dem Bildschirm schreibt.

b) **(4 Punkte)** Implementieren Sie eine Methode

`void printConexComponent(Graph g)`, die die Anzahl der Zusammenhangskomponenten im Graphen  $g$  ausgibt und die Liste der Knoten anzeigt, die jede Zusammenhangskomponente bestimmen. Die Methode sollte zuerst die Anzahl der zusammenhängenden Komponenten im Graphen bestimmen und diese in der ersten Zeile ausgeben. Für jede zusammenhängende Komponente sollte die Methode die Liste der Knoten ausgeben, die sie bestimmen. Die Liste der Knoten für jede zusammenhängende Komponente sollte in eine neue Zeile gedruckt werden, getrennt durch ein einzelnes Leerzeichen.

### (Aufgabe 3)

**(1 Punkt)** Erklären Sie den implementierten Code und beantworten Sie die Fragen der Lehrkraft.