

# Algorithmische Graphentheorie für Informatiker

## Labor 3

### (Aufgabe 1)

(1 Punkt) 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. Adjazenzliste

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 ( $\text{Knoten1 Knoten2}$ ), wobei jede Zeile zwei durch ein Leerzeichen getrennte Ganzzahlen enthält, die die verbundenen Knoten angeben. Die Adjazenzliste wird als dynamischer Vektor von Vektoren repräsentiert.

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) (5 Punkte) Implementieren Sie eine Methode `int weg(int start, int stop)`, welche die Länge eines Weges von minimaler Länge mit Hilfe von Breitensuche zwischen `start` und `stop` findet und diese Länge zurückgibt. Falls so ein Weg nicht existiert, wie die Nachricht "kein Weg" auf dem Bildschirm geschrieben.

Verwende dabei die zugefügten Testdateien:

- Auf der Datei 10k.txt: `weg(0,7738)`, `weg(793, 6174)`
- Auf der Datei 100k.txt: `weg(81768, 55850)`, `weg(19126, 53548)`
- Auf der Datei 1m.txt: `weg(696751, 505396)`, `weg(0, 895812)`

### (Aufgabe 3)

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

