

Algorithmische Graphentheorie für Informatiker

Labor 2

(Aufgabe 1)

- a) **(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 (`Knoten1 Knoten2`), wobei jede Zeile zwei durch ein Leerzeichen getrennte Ganzzahlen enthält, die die verbundenen Knoten angeben. Die Liste wird als dynamischer Vektor von Pairs (`std::pair`) repräsentiert.

2. Adjazenzlisten

Die erste Zeile enthält eine Ganzzahl n , die die Anzahl der Knoten darstellt. Die nächsten n Zeilen bestehen aus der Adjazenzliste des Graphen, wobei jede Zeile einem Knoten und seinen angrenzenden Knoten entspricht, getrennt durch Leerzeichen. Die Adjazenzliste wird als ein Array von dynamischen Vektoren repräsentiert.

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

- b) **(2 Punkte)** Implementieren Sie für die Klasse `Graph` einen Konstruktor ohne Parameter, der einen Graph ohne Knoten und Kanten erstellt, und einen Copy-Konstruktor, der eine Kopie eines als Parameter gegebenen Graphen erstellt, mit der Signatur `Graph(Graph &g)`

(Aufgabe 2)

- a) **(3 Punkte)** Implementieren Sie die Methode `komplement` in der Klasse `Graph`. Diese bekommt keine Parameter und liefert das Komplement des Graphen zurück. Die Methode erstellt einen neuen Graphen (d.h. Verändert den ursprünglichen Graphen nicht).
- b) **(1 Punkt)** Implementieren Sie eine Methode, die die Kanten des ursprünglichen Graphen und des Komplementgraphen auf dem Bildschirm anzeigt.

(Aufgabe 3)

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

