Praktikumsaufgaben

Graphentheoretische Konzepte und Algorithmen

SoSe 24

P. Stelldinger

Aufgaben

Allgemeines zu allen Aufgaben

Die Aufgabenstellung ist aus folgenden Gründen nicht ganz genau spezifiziert:

- Sie sollen einen Entwurfsspielraum haben. Das heißt, Sie können relativ frei entscheiden, wie Sie die Aufgabe lösen, allerdings sollten Sie Ihre Entscheidungen bewusst fällen und begründen können. Insofern gibt es auch keine Musterlösung.
- Durch die unterschiedlichen Lösungen können Sie von Ihren Kommilitonen noch lernen und das *Structured Walk-Through* bleibt spannend.

Darüber hinaus dürfen Sie gerne unterschiedliche Quellen nutzen, aber nur wenn Sie diese auch angeben. Sie sollen auch nicht unbedingt alles selber programmieren (den in der jeweiligen Aufgabe geforderten Kernalgorithmus aber schon), nutzen Sie den vorgegeben Datentyp oder gerne auch andere Libaries.
Für die Bearbeitung der Praktikumsaufgaben erhalten Sie im Teams-Kanal der LV

- Beispielgraphen für das Praktikum (absichtlich auch mit Fehlern)
- Links zu verschiedenen Libaries

Das Ergebnis der Bearbeitung einer Aufgabe wird von jeder Gruppe im Praktikum vorgestellt. Das heißt, die Aufgaben muss *vor Praktikumstermin fertig bearbeitet* sein. Über die Vorstellung hinaus wird für jede Aufgabe erwartet,

1. Implementierung der gestellten Aufgabe, also
   - eine korrekte und möglichst effiziente Implementierung in Java, die der vorgegeben Beschreibung entspricht,
   - die Kommentierung der zentralen Eigenschaften/Ereignisse etc. im Code und
   - hinreichende Testfälle in JUnit und ihre Kommentierung.

   Hinweis: Wenn Sie in der Implementierung englische Fachbegriffe nutzen wollen, können Sie diese im Index des Buches von Diestel nachschlagen.

2. Schriftliche Erläuterung Ihrer Lösung (Lösungsdokumentation) *etwa 4-7 Seiten*
    - Geben Sie Ihre Lösungsdokumentation am Vorabend des Praktikumstermins per E-Mail an peer.stelldinger@haw-hamburg.de ab! In dem Betreff-Feld tragen Sie bitte ein "GKA Praktikum Gruppe [1-3] Team [1-10]".
    - Bringen Sie Ihre Lösungsdokumentation zum Praktikumstermin mit!
    - Bitte geben Sie folgende Daten im Kopf Ihrer Lösungsdokumentation an:
        - **Team:** Namen der Teammitglieder
        - **Aufgabenaufteilung:**
            a) Aufgaben, für die Teammitglied 1 verantwortlich ist;
            Dateien, die komplett/zum Teil von Teammitglied 1 implementiert/bearbeitet wurden
            b) Aufgaben, für die Teammitglied 2 verantwortlich ist;
            Dateien, die komplett/zum Teil von Teammitglied 2 implementiert/bearbeitet wurden
        - **Quellenangaben:** Angabe von wesentlichen Quellen, z.B. Web-Seiten/Bücher, von denen Quellcode/Algorithmen übernommen wurden, Namentliche Nennung von Studierenden der HAW, von denen Quellcode übernommen wurde
        - **Bearbeitungszeitraum:** Datum und Dauer der Bearbeitung an der Aufgabe von allen Teammitgliedern und Angabe der gemeinsamen Bearbeitungszeiten
    - kurze Beschreibung der Algorithmen und
    - Datenstrukturen
    - wesentliche Entwurfsentscheidungen ihrer Implementierung
    - Umsetzung der Aspekte der Implementierung
    - umfassende Dokumentation der Testfälle, wobei die Abdeckung der Testfälle diskutiert werden soll
    - Beantwortung der in der Aufgabe gestellten Fragen

Es gibt drei Praktikumsaufgaben, weitere Details werden in der jeweiligen Aufgabenstellung angegeben.

Eine Praktikumsaufgabe gilt als erledigt, wenn

1. Sie die Lösungsdokumentation am Praktikumsanfang abgegeben haben,
2. Sie im Praktikum Ihre Implementierung vorgestellt haben und diese von mir (mindestens) als ausreichend anerkannt wurde.

Ein Bonuspunkt für die Praktikumsaufgabe wird vergeben, wenn die Lösung und Präsentation sehr gut gelungen sind.

Beachten Sie: Die drei Aufgaben bauen teilweise aufeinander auf. Sie können (und sollten!) also dieselbe Codebasis nutzen und weiterentwickeln.

Aufgabe 1: Laden, Speichern, Visualisieren und Traversieren von Graphen

Die Graphen, mit denen Sie arbeiten, sollen in diesem Format (siehe auch VL-Folien) gespeichert und gelesen werden:
gerichtet

```
<name node1>[ -> <name node2>[(<edge name>)][: <edgeweight>]];
```

ungerichtet

```
<name node1>[ -- <name node2> [(<edge name>)][: <edgeweight>]];
```

Die Aufgabe umfasst:

- die Einarbeitung in die gewählte library
- das Einlesen/ Speichern von ungerichteten sowie gerichteten Graphen und die Visualisierung der Graphen,

  (Hinweis: reguläre Aussdrücke in Java nutzen)
- die Implementierung des Breadth-First Search (BFS) Algorithmus zur Traversierung eines Graphen,
- dabei soll als Ergebnis der kürzeste Weg (in Form einer geeigneten Datenstruktur) und die Anzahl der benötigten Kanten angegeben werden.
- einfache JUnit-Tests, die Ihre Methoden überprüfen und
- JUnit-Tests, die das Lesen, das speichern sowie den BFS-Algorithmus überprüfen unter Benutzung der gegeben *.gka*-Dateien
- die Beantwortung der folgenden Fragen:
  1. Was passiert, wenn Knotennamen mehrfach auftreten?
  2. Wie unterscheidet sich der BFS für gerichtete und ungerichtete Graphen?
  3. Wie können Sie testen, dass Ihre Implementierung auch für sehr große Graphen funktioniert?