



**Tell me and I'll forget
Show me and I'll remember
Involve me and I'll understand.**
(Confucius)

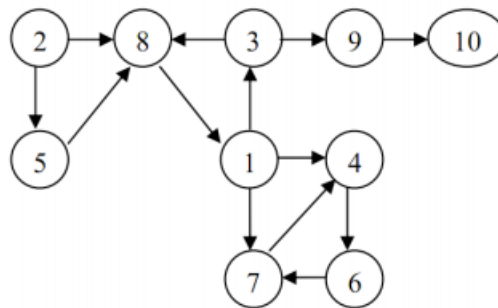
Praktikum 5

Graphen 1

[9+3 Punkte]

Fingerübungen

Gegeben ist folgender Graph G:



- 1) Geben Sie den Graph G als Adjazenz-Matrix an.
- 2) Geben Sie den Graph G in Adjazenzlisten-Darstellung an.
- 3) Führen Sie für den Graphen G eine Breitensuche durch, beginnend im Knoten 1.

Bewertete Aufgaben

Aufgabe 1: Graph-Repräsentation [1 Punkt]

Ziel: Sie untersuchen, welche Darstellungsform von Graphen für welche Anwendung sinnvoll ist.

Graphen kann man als Adjazenz-Matrix oder mit Hilfe von Adjazen-Listen darstellen.



- Geben Sie mindestens drei verschiedene Operationen (wie z.B. Einfügen eines Knoten, Löschen einer Kante etc.) an, die in einer Adjazenzmatrix schneller sind als in Adjazenz-Listen. Begründen Sie, warum.
- Geben Sie umgekehrt mindestens drei Operationen an, die mit Adjazenz-Listen schneller sind.

Abgabe: Dokumentation

Aufgabe 3: Datenstruktur für Graphen [5 Punkte]

Ziel: Sie entwickeln einen Graphen, mit dem man die Distanzen von Schweizer Städten verwalten kann.

Teilaufgaben:

- Entwickeln Sie ein Interface für einen gerichteten Graphen. Das Interface soll unter anderem folgende Methoden anbieten:
 - Einen gerichteten Graphen aus einem Textfile einlesen. Der Graph soll gewichtet oder ungewichtet sein - je nachdem ob im Textfile Gewichte stehen oder nicht.
 - Anzahl Knoten des Graphen zurückgeben
 - Anzahl Kanten des Graphen zurückgeben
 - Liste aller Knoten zurückgeben
 - Für einen Knoten die Liste aller Kanten zurückgeben (ggf. mit Gewichten)
- Implementieren Sie einen Graphen, der das Interface aus Teilaufgabe a) implementiert und intern eine Adjazenz-Matrix verwendet.
- [Optional] Implementieren Sie einen Graphen, der das Interface aus Teilaufgabe a) implementiert und intern Adjazenz-Listen verwendet.

Hinweis: Diese Datenstruktur wird in nachfolgenden Praktika wieder verwendet. Achten Sie bereits jetzt auf eine saubere und korrekte Implementierung (CleanCode, Testing).

Abgabe: Java-Code



Aufgabe 4: Breitensuche [3 Punkte]

Ziel: Sie implementieren die Breitensuche.

Als Grundlage verwenden Sie Ihren Graphen aus der vorherigen Aufgabe.

Teilaufgaben:

- Implementieren Sie eine Breitensuche, die für einen Graphen die Knoten in einer richtigen Reihenfolge zurückgibt.
- Wenden Sie Ihre Breitensuche auf den Graphen aus Aufgabe 1 an, und vergleichen Sie die Ergebnisse.

Abgabe: Java-Quelltext + Ergebnis Breitensuche.

Challenge [optional, 3 Punkte]



In einer Box liegen 12 Tennisbälle. Elf davon sind gleichschwer, der zwölfte hat ein anderes Gewicht. Sie wissen nicht ob er leichter oder schwerer ist.

Sie sollen diesen Ball finden. Dazu können Sie nur eine alte Balkenwaage (siehe Bild) verwenden. Sie haben keine Gewichte, können also nur jeweils verschiedene Mengen von Bällen miteinander vergleichen.

Schaffen Sie es, mit 3 Messungen den Ball zu finden?