

# Übungsserie 10

## Aufgabe 1: Adjazenz-Liste und Adjazenz-Matrix

Würden Sie in den folgenden Fällen eine *Adjazenz-Liste* oder eine *Adjazenz-Matrix* verwenden? Begründen Sie ihre Antwort aufgrund Folie 13 'Performance'.

- Der Graph hat 10'000 Knoten und 20'000 Kanten und es ist wichtig so wenig Platz wie möglich zu verwenden.
- Der Graph hat 10'000 Knoten und 20'000'000 Kanten und es ist wichtig, dass das Löschen von Knoten so schnell wie möglich ist.
- Es ist wichtig die Anfrage auf *Nachbarschaft* so schnell wie möglich zu beantworten. Der verwendete Platz spielt keine Rolle.

## Aufgabe 2: Implementation Graph

Es sollen in einer Graphen-Klasse die Methoden

`Graph.areAdjacent()` / `are_adjacent()` resp. `Graph.opposite()` implementiert werden.

Die Ausgangslage liegt auf ILIAS.

Es müssen nur in der Datei `Graph.java` resp. `graph.py` die beiden mit "*TODO*" markierten Methoden implementiert werden.

Hinweise:

- Es sollen jeweils nur die ADT-Graphen-Methoden gemäss der Skript-Folie 9 verwendet werden (Ausnahme: Bei Java kann anstelle von `Graph.incidentEdges()` die entsprechende Methode auf dem Vertex verwendet werden, z.B. `vi.getIncidentEdges()`).
- Die bestehenden Datei-Namen und weitere Definitionen dürfen nicht verändert werden (Package-Deklarationen, Imports, Klassennamen, Attribute, Methoden, etc.). Dies bedeutet somit insbesondere auch, dass sich die Dateien bei Java im Verzeichnis `uebung10/as/aufgabe02` befinden resp. bei Python im Verzeichnis `uebung10/al/aufgabe02`.

### Testat

Diese Aufgabe 2 wird als Testat abgegeben.

Geprüft wird die Funktionalität der Methoden:

- `opposite(Vertex, Edge)`
- `areAdjacent(Vertex, Vertex)`

### Abgabe

Die Abgabe erfolgt gemäss:

ILIAS: Unterlagen zum Modul > Administration > Testate > Testat 1