

8. Übungsblatt zur Vorlesung Algorithmen und Datenstrukturen (Winter 2025/26)

Aufgabe 1 – Binär hochzählen

Die Datenstruktur D enthält eine einzige natürliche Zahl Z in Binärdarstellung (zu Beginn ist $Z = 0$) und stellt lediglich die Methode `Increment` zur Verfügung, die Z um Eins erhöht. Die Laufzeit der Methode entspricht dabei der Anzahl der Bits, die sich in Z durch die Erhöhung um Eins ändern.

Zeigen Sie, dass die *amortisierte* Laufzeit von `Increment` in $O(1)$ ist!

4 Punkte

PABS

Aufgabe 2 – Zusammenhangskomponenten finden

In dieser Aufgabe wollen wir einen Algorithmus implementieren, der die Anzahl der Zusammenhangskomponenten in einem gegebenen ungerichteten Graphen berechnet.

Zu Beginn implementieren wir eine Datenstruktur, die einen ungerichteten Graph repräsentiert.

- a) Implementieren Sie im Paket `graph` die Klasse `Vertex`. Diese soll einen Konstruktor `public Vertex(Integer key)` enthalten, der einen neuen Knoten mit einem Verweis auf eine Zahl `key` der Klasse `Integer` erzeugt.

Außerdem soll die Klasse mindestens über folgende Methoden verfügen:

- `public List<Vertex> getNeighbors()` – gibt die Adjazenzliste des Knotens zurück.
- `public Integer getKey()` – gibt die Zahl zurück, auf die der Knoten verweist.

2 Punkte

- b) Implementieren Sie im Paket `graph` die Klasse `UndirectedGraph`. Diese soll einen Konstruktor `public UndirectedGraph()` enthalten, der einen ungerichteten Graph mit einer leeren Knotenliste erzeugt.

Außerdem soll die Klasse mindestens über folgende Methoden verfügen:

- `public Vertex addVertex(Integer key)` – erzeugt einen neuen Knoten v mit Verweis auf `key` und fügt ihn zur Liste der Knoten hinzu. Anschließend wird v zurückgegeben.

- `public void addEdge(Vertex u, Vertex v)` – Erzeugt eine ungerichtete Kante zwischen Knoten `u` und `v`.
- `public List<Vertex> getVertices()` – Gibt eine Liste aller Knoten im Graphen zurück.

4 Punkte

Nun wollen wir den eigentlichen Algorithmus implementieren, der die Anzahl der Zusammenhangskomponenten für einen gegebenen `UndirectedGraph` zurückgibt.

c) Beschreiben Sie einen Algorithmus in Worten, der dieses Problem in Linearzeit löst.

4 Punkte

d) Implementieren Sie im Paket `algo` die Klasse `ConnectedComponents`, die mindestens die Methode `public static int getNumComponents(UndirectedGraph graph)` enthält. Die Methode soll die Anzahl der Zusammenhangskomponenten von `graph` zurückgeben. Die Laufzeit soll linear in der Größe des Graphs sein.

6 Punkte

Hinweis: Sie dürfen Datenstrukturen wie `ArrayList`, `LinkedList` oder `ArrayDeque` aus der Standardbibliothek benutzen.

Bitte geben Sie Ihre Lösungen bis **Donnerstag, 8. Januar 2026, 14:00 Uhr** einmal pro Gruppe über Wuecampus als pdf-Datei ab. Vermerken Sie dabei stets die Namen und Übungsgruppen aller BearbeiterInnen auf der Abgabe.

Grundsätzlich sind stets alle Ihrer Aussagen zu begründen und Ihr Pseudocode ist stets zu kommentieren.

Die Lösungen zu den mit PABS gekennzeichneten Aufgaben, geben Sie bitte nur über das PABS-System ab. Vermerken Sie auf Ihrem Übungsblatt, in welchem Repository (sXXXXXX-Nummer) die Abgabe zu finden ist. Geben Sie Ihre Namen hier als Kommentare in den Quelltextdateien an.