

---

# Algorithmen für die Speicherhierarchie

---

*Abgabetermin: 15.7.2009 vor der Vorlesung*

## BFS

### Aufgabe 1

Ein beliebiger gerichteter Graph (ohne isolierte Knoten) kann auf verschiedene Arten repräsentiert werden:

1. Als eine einfache Liste der Kanten.
2. Durch Sortieren der Kanten nach Startknoten, kann man auf die explizite Angabe der Startknoten jeder einzelnen Kante verzichten, wenn in einer zweiten Tabelle für jeden Knoten notiert ist, ab welcher Position alle von diesem Knoten ausgehenden Kanten gespeichert sind.

Der in der Vorlesung vorgestellte Algorithmus zum Bestimmen der BFS-Level benötigte die zweite Repräsentation.

Wie kann man die beiden Repräsentationen ineinander umwandeln?

Wie kann man einen Graphen der in letzterer Form repräsentiert ist, um antiparallele Kanten ergänzen? Welche I/O-Komplexität hat Ihre Lösung?

### Aufgabe 2

Gegeben sei ein ungerichteter Graph  $G$  mit einer Wurzel  $w$ . Alle Knoten des Graphen sind mit ihrer Entfernung zu  $w$  (Level) markiert. Alle Kanten haben eine eindeutige Nummer. Gesucht ist die BFS-Nummerierung der Knoten, nach folgenden Regeln:

- Die Kinder von Knoten mit kleinerer BFS-Nummer werden vor Kindern von Knoten mit größerer Nummer nummeriert.
- Die Kinder werden sortiert nach ihrer Kantenummer nummeriert.

Beschreiben Sie detailliert, wie sich direkt eine solche BFS-Nummerung der Knoten aus der bekannten Level-Struktur erzeugen lässt.

Welche I/O-Komplexität ergibt sich für die Nummerierung des gesamten Graphen?

Kann der Algorithmus eine fehlerhafte Levelnummerierung erkennen?