

---

## Grundlagen: Algorithmen und Datenstrukturen

---

*Abgabetermin: In der jeweiligen Tutorübung*

### Aufgabe 1

Bestimmen Sie die Laufzeit des BFS-Algorithmus. Betrachten Sie zwei Implementierungen: Zum einen basierend auf einer Adjazenzmatrix und zum anderen basierend auf einer Adjazenzliste.

### Aufgabe 2

Gegeben Sei die Adjazenzmatrix  $A = (a_{ij})_{1 \leq i, j \leq n}$  eines gerichteten, einfachen Graphen  $G$  mit  $n$  Knoten. Geben Sie einen Algorithmus an, der eine Senke in  $\mathcal{O}(n)$  Schritten findet. Eine Senke sei (abweichend von der üblichen Definition) ein Knoten  $v$ , der Eingrad  $n - 1$  und keine ausgehenden Kanten hat.

### Aufgabe 3

In dieser Aufgabe wollen wir ein alternatives Verfahren zur topologischen Sortierung auf einem DAG  $G$  betrachten. Dazu nehmen wir an, dass uns weder die üblichen Graphoperationen zum Traversieren einer Kante oder Abfragen der Nachbarschaft eines Knotens, noch irgendwelche „höheren Datenstrukturen“ wie z.B. Queues, Listen, PriorityQueues oder Stacks zur Verfügung stehen.

Die einzige Operation, die wir benutzen können ist eine Tiefensuche (DFS), die Arrays `dfs_num` und `finish_num` zurückgibt, die zu den jeweiligen Knoten die entsprechenden Werte beinhalten. Außerdem stehen uns `for`-Schleifen und die übliche Addition bzw. Subtraktion zur Verfügung.

- Geben Sie einen Algorithmus (Pseudocode reicht) an, der dieselbe asymptotische Laufzeit erreicht, wie der in der Vorlesung vorgestellte Algorithmus.
- Begründen Sie die Laufzeit Ihres Algorithmus kurz.
- Beweisen Sie die Korrektheit Ihres Algorithmus.