

## DAP2 – Präsenzübung 12

Besprechung: 12.07.2017 — 14.07.2017

### Präsenzaufgabe 12.1: Floyd–Warshall-Algorithmus

In dieser Aufgabe widmen wir uns dem Floyd–Warshall-Algorithmus.

**Input:** Ein gewichteter, gerichteter Graph  $G$  ohne negativen Zyklen, dessen Kantengewichte durch  $W$  gegeben sind und die Anzahl der Knoten  $n$

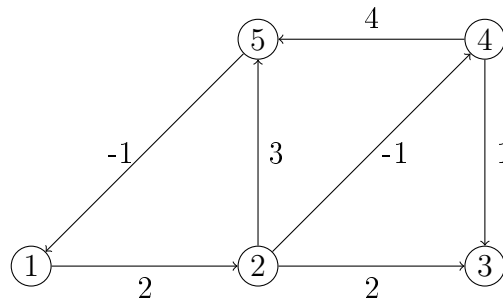
**Output:** Für jedes Paar von Knoten  $u, v \in V[G]$  die Distanz von  $u$  nach  $v$  sowie einen kürzesten Weg.

```

1 FLOYDWARSHALL( $W, n$ )
2    $D^{(0)} \leftarrow W$ 
3   for  $k \leftarrow 1$  to  $n$  do
4     for  $i \leftarrow 1$  to  $n$  do
5       for  $j \leftarrow 1$  to  $n$  do
6          $d_{ij}^{(k)} \leftarrow \min\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\}$ 
7   return  $D^{(n)}$ 

```

- a) Führen Sie den Floyd–Warshall-Algorithmus für den folgenden Graphen  $G = (V, E)$  aus. Geben Sie dabei nach der Initialisierung, also für  $k = 0$ , und nach jeder Iteration  $k$  der äußersten **for**-Schleife die Distanzmatrix  $D^{(k)}$  an.

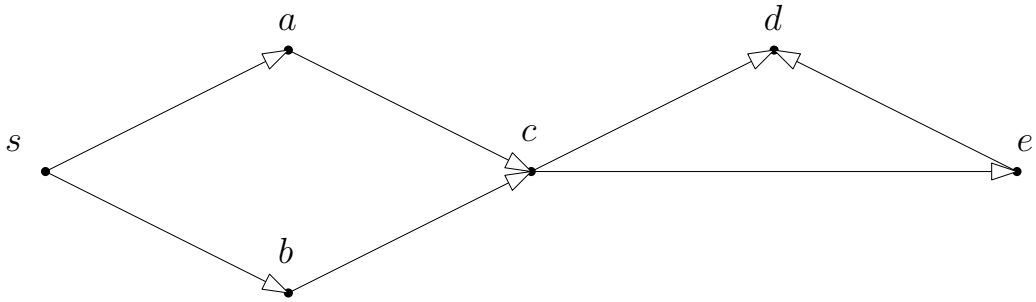


- b) Im Floyd–Warshall-Algorithmus benutzen wir die Distanzmatrizen  $D^{(k)}$ ,  $0 \leq k \leq n$ . Zeigen Sie, dass man mit genau einer Distanzmatrix  $D$  auskommen kann. Geben Sie dazu die entsprechende Modifikation am Pseudocode an und begründen Sie die Korrektheit dieser Modifikation.

### Präsenzaufgabe 12.2: (Graphenalgorithmien)

Gegeben sei ein gerichteter azyklischer Graph  $G = (V, E)$  und der Startknoten  $s \in V$ . Uns interessiert die Anzahl der unterschiedlichen Pfade von  $s$  zu allen anderen Knoten aus  $V$ . Sei  $P(u)$  die Anzahl der Pfade zum Knoten  $u \in V$ .

Im unten abgebildeten Graph ist z. B.  $P(c) = 2$  und  $P(d) = 4$ .



- a) Für alle Knoten  $u \in V$  mit  $(u, v) \in E$  sei  $P(u)$  berechnet. Wie kann man  $P(v)$  berechnen?
- b) Entwerfen Sie einen Algorithmus  $\text{AnzahlPfade}(V, E, s)$ , der die Anzahl unterschiedlicher Pfade  $P(u)$  vom Startknoten  $s$  zu jedem Knoten  $v \in V$  berechnet. Beschreiben Sie den Algorithmus zunächst mit eigenen Worten. Setzen Sie den Algorithmus dann in Pseudocode um. Für die volle Punktzahl wird ein Algorithmus erwartet, dessen Laufzeit durch  $\mathcal{O}(|V| + |E|)$  beschränkt ist.
- c) Analysieren Sie die Laufzeit Ihres Algorithmus.
- d) Beweisen Sie die Korrektheit Ihres Algorithmus.