Übungsblatt 8.

Bearbeiten bis: **Montag, 12.05.2025**

Aufgabe 1 (Alle-Paare-kürzeste-Wege | 4 Punkte). Gegeben Sei ein Digraph $G = (V, E, \omega)$ dargestellt durch die Matrix $\mathbf{W}$ mit

$$w_{ij} = \begin{cases} 0, & i = j, \\ \omega(i,j), & (i,j) \in E, \\ \infty, & \text{sonst.} \end{cases}$$

Die Ausgabe eines alle-Paare-kürzeste-Wege Algorithmus kann mit Hilfe der Matrix $\mathbf{L}$ dargestellt werden, wobei $l_{i,j} = \delta(i,j)$ der kürzeste Weg von i nach j ist:

$$l_{ij} = \begin{cases} 0, & i = j \\ \delta(i,j), & (i,j) \in E, \\ \infty, & \text{sonst.} \end{cases}$$

Für einen Digraphen mit nicht negativen Zyklen hat der kürzeste Weg zwischen zwei beliebigen Knoten maximal Länge $|V| - 1$. Um alle Paare kürzester Wege zu berechnen kann man rekursiv vorgehen. Hierzu bestimmt man Matrizen $\mathbf{L}^{(m)}$ welche die kürzesten Wege mit Länge m enthalten, angefangen mit $\mathbf{L}^{(0)}$, wobei

$$l_{ij}^{(0)} = \begin{cases} 0, & i = j, \\ \infty, & \text{sonst.} \end{cases}$$

Gegeben $\mathbf{L}^{(m)}$ kann man folgende Rekursion anwenden um die Matrix $\mathbf{L}^{(m+1)}$ auszurechnen

$$l_{ij}^{(m+1)} = \min\left\{ l_{ij}^{(m)}, \min_{k \in V} \left\{ l_{ik}^{(m)} + w_{kj} \right\} \right\}.$$

(a) Wie sehen die Matrizen $\mathbf{L}^{(m)}$, $m = 0, \dots, |V| - 1$, für den Digraphen G_1 aus Abbildung 1 aus?

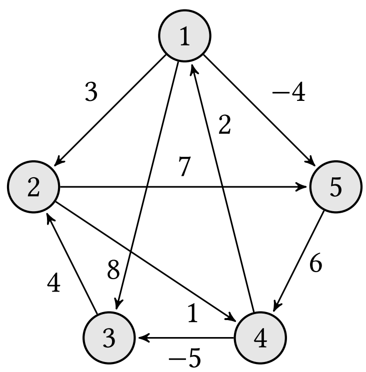

Abbildung 1: Digraph G_1

(b) Implementieren Sie einen Algorithmus, welcher als Eingabe die Matrizen $\mathbf{W}$ und $\mathbf{L}^{(m)}$ nimmt und mit Aufwand $\mathcal{O}(|V|^3)$ die Matrix $\mathbf{L}^{(m+1)}$ ausrechnet.

(c) Die einfachste Methode um das Endergebnis $\mathbf{L}^{(|V|-1)}$ zu erhalten, ist alle $\mathbf{L}^{(m)}$ für $m = 1, \dots, |V| - 1$ auszurechnen. Dies ergibt einen Aufwand von $\mathcal{O}(|V|^4)$. Implementieren Sie diesen „einfachen“ Algorithmus und wenden Sie ihn auf den Digraphen G_1 aus Abbildung 1 an.

Aufgabe 2 (Transitive Hülle). Sei $G = (V, E)$ ein Digraph. Die transitive Hülle von G ist gegeben durch $G' = (V, E')$ mit $E' = \{(u, v) \in V \times V \ : \text{ es gibt einen } u\text{-}v\text{-Weg in } G\}$. Geben Sie einen Algorithmus zur Berechnung der transitiven Hülle eines Digraphen an. Können Sie den Algorithmus verbessern, sodass er weniger Speicher braucht und in den meisten Fällen auch schneller terminiert, wenn Sie davon ausgehen, dass logische Operationen schneller sind als arithmetische Operationen?

Aufgabe 3 (Floyd-Warshall).

(a) Implementieren Sie den Algorithmus von Floyd-Warshall und testen Sie ihn auf dem Digraphen G_1 aus Abbildung 1. Wie sehen die Matrizen $\mathbf{L}$ und $\mathbf{P}$ in jedem Schritt aus?

(b) Vergleichen Sie die Laufzeit des Algorithmus von Floyd-Warshall mit den beiden alternativen Algorithmen aus Aufgabe 1. Generieren Sie hierfür eine Matrix $\mathbf{W}$ durch Anwendung des folgenden Codes:

```
sizeW = 100; %dimension of weight matrix
maxW = 50;   %maximum weight
minW = -50;  %minimum weight
W = rand(sizeW,sizeW);
W(find(W > 0.5)) = Inf;
W = minW +2*(maxW-minW).*W;
W(1:sizeW+1:end) = 0;
```

Hinweis. Verwenden Sie für Zeitmessungen die `MATLAB`-Befehle `tic` und `toc`.

Aufgabe 4 (Minimaler Schnitt | 4 Punkte). Gegeben sei das Netzwerk $N = (V, E, c, s, t)$ aus Abbildung 2. Bestimmen Sie durch Vergleichen aller möglichen Schnitte S den Schnitt mit minimaler Kapazität, und finden Sie einen Fluss mit maximalem Fluss.

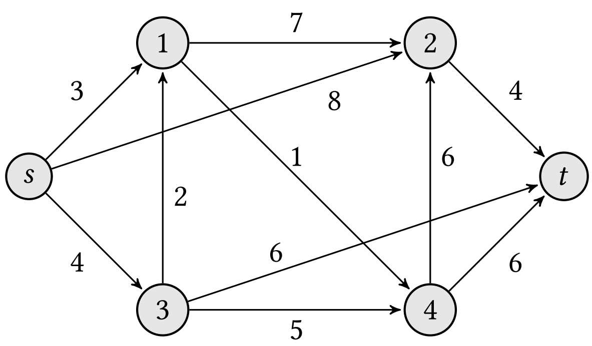

Abbildung 2: Netzwerk N.