operations research
QUANTITATIVE INFRASTRUKTURSYSTEMMODELLIERUNG

Tutorium

Operations Research – Grundlagen

Technische Universität Berlin
Fachgebiet **W**irtschafts- und **I**nfrastruktur**P**olitik

Greedy, Kürzeste-Wege und Bellman-Ford-Algorithmus

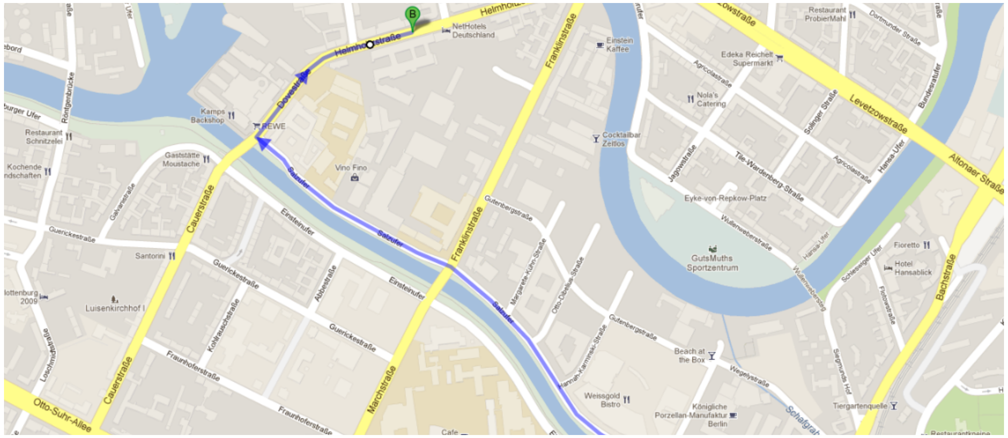
Tutoriumsaufgaben:

3.5 a)	Greedy Algorithmus
2.32	Kürzeste Wege
3.12	Bellman-Ford-Algorithmus I

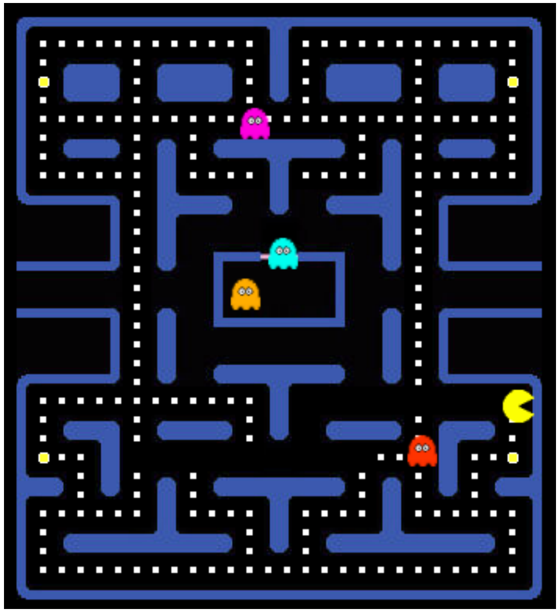
Freiwillige Hausaufgabe:

3.13	Bellman-Ford-Algorithmus mit Tree-Matrix
3.14	Bellman-Ford-Algorithmus II

Was verbindet die Abbildungen?



→ „kürzeste“ Wege Problem



Minimaler Spannbaum vs. das Kürzeste-Wege-Problem

<u>Minimaler Spannbaum</u>

- Verbinden **<u>alle Knoten</u>** mit möglichst geringer Kantenzahl und geringem Kantengewicht
- Bestimmt nicht unbedingt den kürzesten Weg zwischen 2 Knoten, sondern den kürzesten Weg, der alle Knoten verbindet.

WIE?

- Minimaler Spannbaum als LP
- Kruskal Algorithmus

<u>Das Kürzeste-Wege-Problem</u>

- Finde den **<u>kürzesten Weg</u>** von einem bestimmten Knoten zu einem bestimmten Knoten
- Verbindet **<u>nicht alle Knoten</u>**!

WIE?

- Kürzeste Wege als LP
- Greedy Algorithmus
- Bellman-Ford Algorithmus

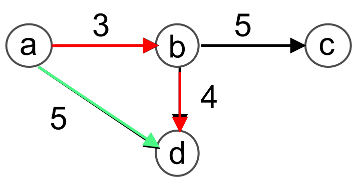
Greedy Algorithmus

- **Wozu?**
  - Findet eventuell den kürzesten Weg zwischen zwei gegebenen Knoten.
  - Aufbau nach dem kleinsten Kantengewicht.

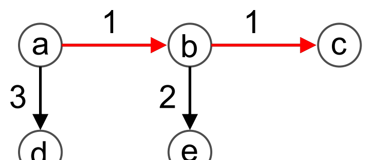
- **Funktionsweise**
  - Beim Startknoten anfangen und sehen, welche Kanten daraus gehen.
  - Wähle als nächstes immer die Kante mit dem geringsten Kantengewicht ohne Rücksicht auf Folgeknoten zu nehmen.

- **Vorteil**
  - Schnell und einfach.

- **Nachteil**
  - Findet nicht immer den kürzesten bzw. überhaupt einen Weg.
  - Bsp. Finde den kürzesten Weg von a nach d.



Weg : a → b → d
Weglänge = 3+4 = 7
Aber
Weg : a → d
Weglänge = 5



Der Weg führt nicht zum Endknoten d

3.5 Greedy-Algorithmus in ungerichteten Graphen

Gegeben ist der Graph eines ambulanten Pflegedienstes. Finden Sie den „kürzesten Weg“ einer Krankenschwester von Patient a zu Patient e mit Hilfe des **Greedy- Algorithmus**.

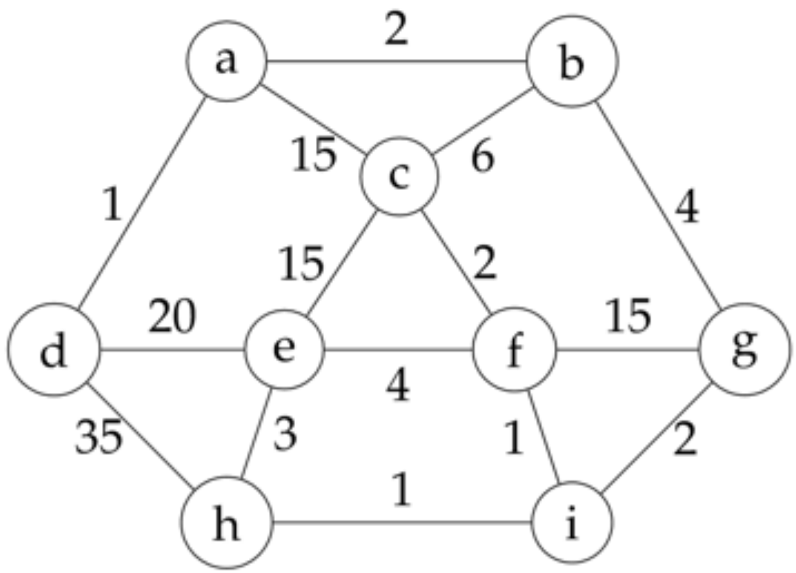
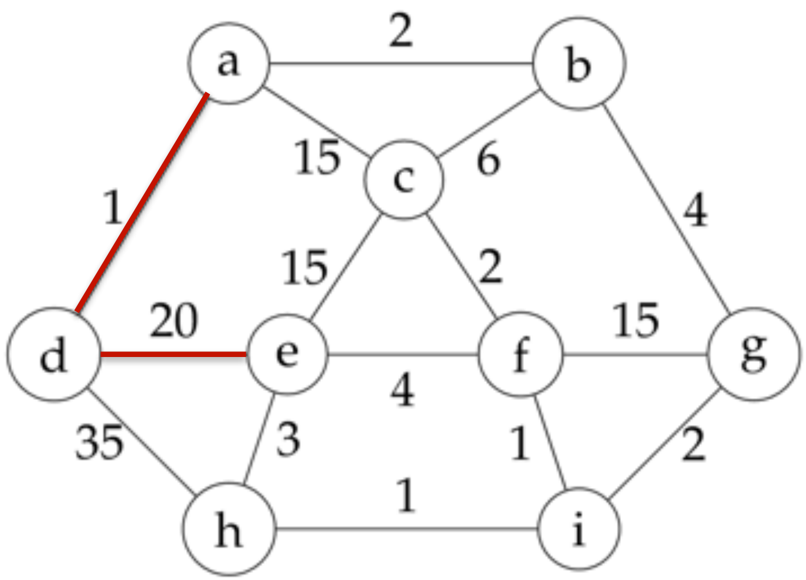


Abbildung: Patientennetzwerk

3.5 Greedy-Algorithmus in ungerichteten Graphen

Gesucht ist der kürzeste Weg von a → e

a) Greedy Algorithmus:



Weg von a nach e: a → d → e

Länge L= 21

2.32 Kürzeste Wege

Sei V die Menge aller Knoten eines Graphen und i, j, s, t ∈ V Knoten, wobei s den Startknoten und t den Zielknoten bezeichnet. Des Weiteren sei x_{ij} ∈ {0,1} und beschreibt, ob eine Kante vom Knoten i zum Knoten j Teil des kürzesten Weges (=1) ist oder nicht (=0).

w_{ij} bezeichnet das Gewicht der Kante von i nach j.

a) Stellen Sie das Kürzeste-Wege-Problem allgemein als Optimierungsproblem auf.

b) Nun stellen sie das spezielle Problem für den gegebenen Graphen auf. Die Lösung des Problems soll dabei den Pfad und die Länge des kürzesten Weges **von Knoten a nach Knoten f** ergeben.

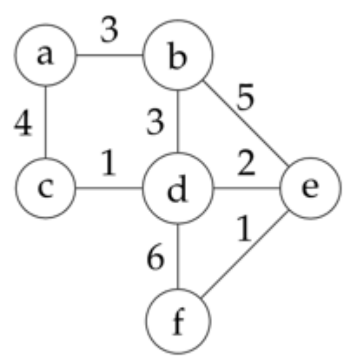


Abbildung 1: Ungerichteter Graph

2.32 a) Kürzeste Wege

V: Knoten

E: Kanten

$i,j,s,t \in V$

s: Startknoten

t: Endknoten

$w_{i,j} \in \mathbb{R}$

→ Kantengewicht

$x_{i,j} \in \{0,1\}$

→ Entscheidungsvariable

→ (Kante ist im KW enthalten $x_{i,j} = 1$, Kante ist nicht im KW enthalten $x_{i,j} = 0$)

2.32 a) Kürzeste Wege

Zielfunktion

$$\min z = \sum_{(i,j)\in E} w_{ij} * x_{ij}$$

- **z: Länge des KW**

V: Knoten

E: Kanten

i, j, s, t ∊ V

s: Startknoten

t: Endknoten

2.32 a) Kürzeste Wege

Nebenbedingung

(Alles was rein geht – Alles was rausgeht)

Startknoten

$$\sum_{j:(j,s)\in E} x_{js} - \sum_{j:(s,j)\in E} x_{sj} = -1$$

- **In den Startknoten geht eine Kante mehr raus als wieder rein**

V: Knoten

E: Kanten

i, j, s, t ∊ V

s: Startknoten

t: Endknoten

2.32 a) Kürzeste Wege

Nebenbedingung

(Alles was rein geht – Alles was rausgeht)

Transportknoten

$$\sum_{j:(j,i)\in E} x_{ji} - \sum_{j:(i,j)\in E} x_{ij} \quad = 0 \;\; für\ alle\ V\backslash\{s,t\}$$

V: Knoten

E: Kanten

i, j, s, t ∊ V

s: Startknoten

t: Endknoten

- **Diese NB muss für jeden Knoten außer den Start- und Endknoten aufgestellt werden**
- **In einen Transportknoten gehen genau so viele Kanten rein wie raus**

2.32 a) Kürzeste Wege

Nebenbedingung

(Alles was rein geht – Alles was rausgeht)

Endknoten

$$\sum_{j:(j,t)\in E} x_{jt} - \sum_{j:(t,j)\in E} x_{tj} \quad = 1$$

- **In den Endknoten gehen eine Kante mehr rein als raus**

V: Knoten

E: Kanten

i, j, s, t ∈ V

s: Startknoten

t: Endknoten

2.32 a) Kürzeste Wege

Definitionsbereich

$x_{ij} \in \{0,1\}$

$w_{ij} \in \mathbb{R}$

V: Knoten

E: Kanten

i, j, s, t ∈ V

s: Startknoten

t: Endknoten

2.32 Kürzeste Wege

b) Nun stellen sie das spezielle Problem für den gegebenen Graphen auf. Die Lösung des Problems soll dabei den Pfad und die Länge des kürzesten Weges von Knoten a nach Knoten f ergeben.

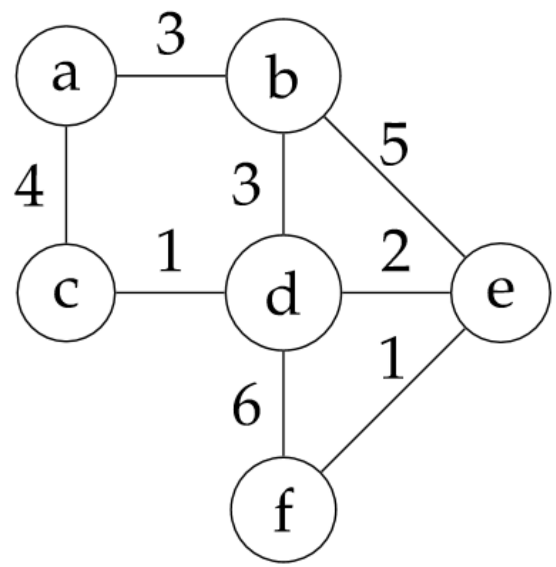


Abbildung 1: Ungerichteter Graph

2.32 Kürzeste Wege

Zielfunktion

$$\min z = \sum_{(i,j) \in E} w_{ij} * x_{ij}$$

$$\begin{aligned}
\min z = \quad & 3 \cdot (x(a,b) + x(b,a)) & +4 \cdot (x(a,c) + x(c,a)) \\
& +1 \cdot (x(c,d) + x(d,c)) & +3 \cdot (x(b,d) + x(d,b)) \\
& +5 \cdot (x(b,e) + x(e,b)) & +2 \cdot (x(d,e) + x(e,d)) \\
& +6 \cdot (x(d,f) + x(f,d)) & +1 \cdot (x(e,f) + x(f,e))
\end{aligned}$$

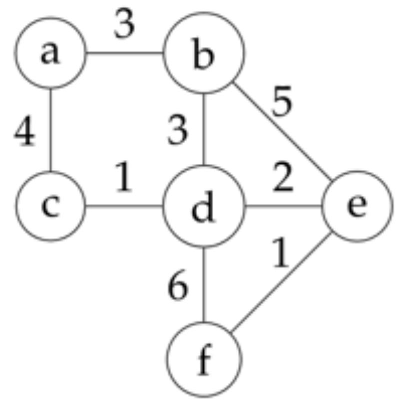


Abbildung 1: Ungerichteter Graph

2.32 Kürzeste Wege

Nebenbedingung - Startknoten

$$\sum_{j:(j,s)\in E} x_{js} - \sum_{j:(s,j)\in E} x_{sj} = -1$$

Nebenbedingung für den Startknoten a:

$$x(c,a) + x(b,a) - (x(a,b) + x(a,c)) \quad = -1$$

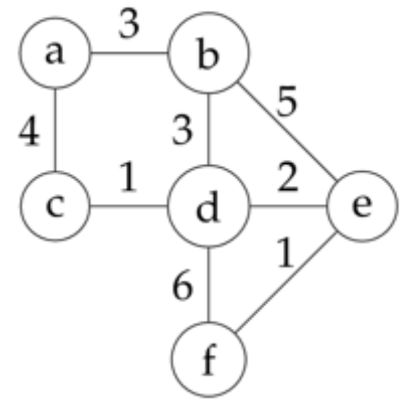


Abbildung 1: Ungerichteter Graph

2.32 Kürzeste Wege

Nebenbedingung - Transportknoten

$$\sum_{j:(j,i)\in E} x_{ji} - \sum_{j:(i,j)\in E} x_{ij} \quad = 0 \;\; f\ddot{u}r\ alle\ V \backslash \{s,t\}$$

Beispielhafte Nebenbedingung für den Transportknoten d:

$$x(c,d) + x(b,d) + x(f,d) + x(e,d) - (x(d,c) + x(d,b) + x(d,f) + x(d,e)) \quad = 0$$

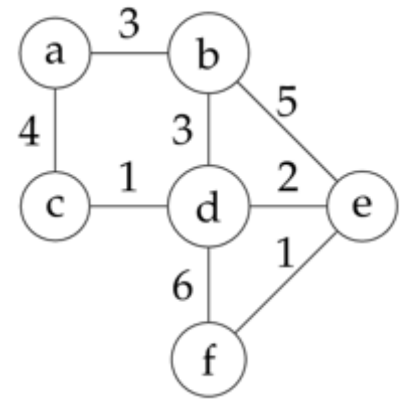


Abbildung 1: Ungerichteter Graph

2.32 Kürzeste Wege

Nebenbedingung - Endknoten

$$\sum_{j:(j,t)\in E} x_{jt} - \sum_{j:(t,j)\in E} x_{tj} \quad = 1$$

Nebenbedingung für den Zielknoten f:

$$x(d,f) + x(e,f) - (x(f,d) + x(f,e)) \quad = 1$$

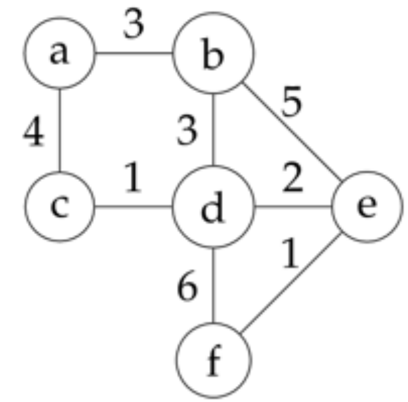


Abbildung 1: Ungerichteter Graph

Bellman-Ford Algorithmus

Wozu?

- Findet alle kürzesten Wege zwischen **allen Knoten**

Voraussetzungen

- Gewichteter Digraph
- Keine negativen Zyklen

Vorteil

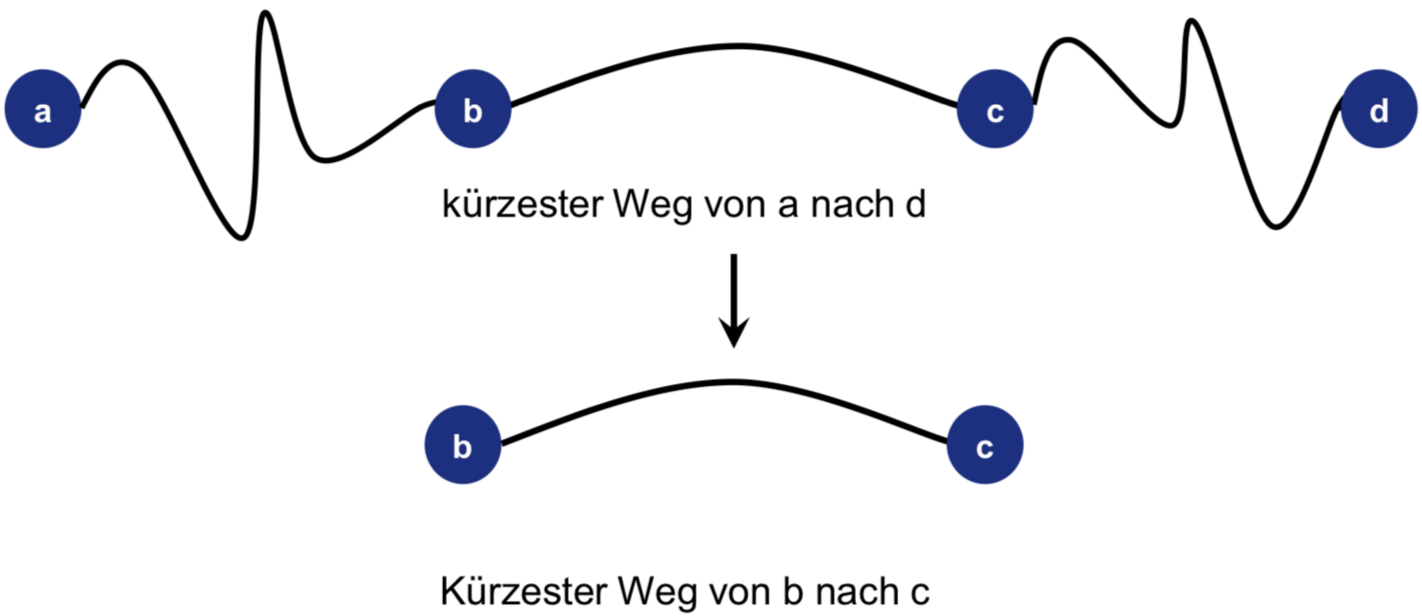
- Negative Kantengewichte sind erlaubt (im Gegensatz zum Dijkstra-Algorithmus)!
- Kein fester Startknoten

Funktionsweise

- Bsp. #3.12

Prinzip der optimalen Substruktur

Ein kürzester Weg zwischen zwei beliebigen Knoten *i* und *j* in einem Graphen *g(V,E,w)* setzt sich immer aus kürzesten Teilwegen zusammen.



Bewertungsmatrix

- Gibt das Kantengewicht zwischen zwei Knoten an
- Eintrag b_{ij} ist das Gewicht der Kante e(i,j)
- Für $i = j$, $b_{ij} = 0$
- Falls die Kante nicht existiert, $b_{ij} = \infty$

U-Matrix

- U^n : Gibt den kürzesten Weg über maximal n Kanten an
- Berechnung über Bellman-Gleichung

$$u_{ij}{}^1 = b_{ij}$$
$$u_{ij}{}^{m+1} = min_k[u_{ik}^m + b_{kj}]$$

Abbruchkriterien

a) Negative Einträge auf der Hauptdiagonalen → Negativer Zyklus
b) Keine Veränderung der U-Matrix nach Iteration
c) Algorithmus durchläuft maximal V Iterationen

Tree-Matrix

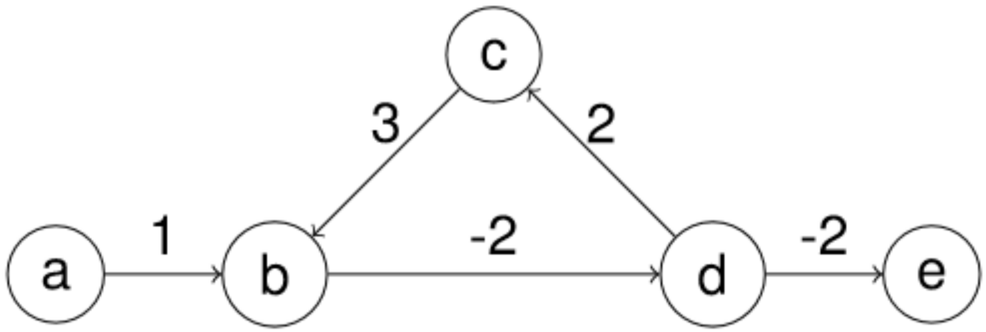
- Speichert alle kürzesten Vorgängerknoten

Initialisierung:

- Wenn die Kante existiert, dann ist der Eintrag $tree_{ij} = i$
- Ansonsten $tree_{ij} = -1$
- Wenn sich U-Matrix ändert, muss sich auch Tree-Matrix ändern

3.12 Bellman-Ford-Algorithmus I

a) Finden Sie den kürzesten Weg von a nach e mit dem Bellman-Ford-Algorithmus.

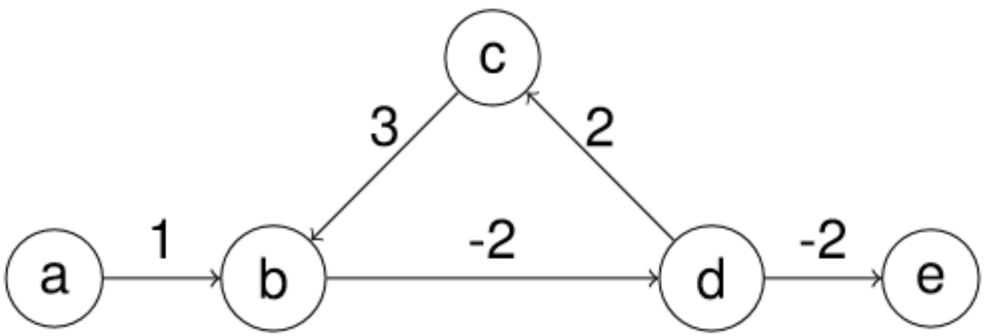


Bellman-Ford-Algorithmus

1. Bewertungsmatrix B:

b_{ij} gibt an, welches <u>Gewicht</u> eine Kante zwischen den Knoten i und j hat.

B = $\begin{pmatrix} \\ \\ \\ \\ \end{pmatrix}$



Bellman-Ford-Algorithmus

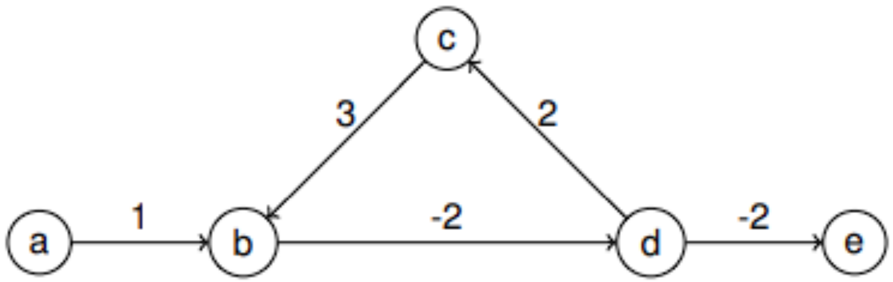


Abbildung: Graph ohne negativem Zyklus

1. Bewertungsmatrix

$$B = \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{pmatrix}$$

Bellman-Ford-Algorithmus

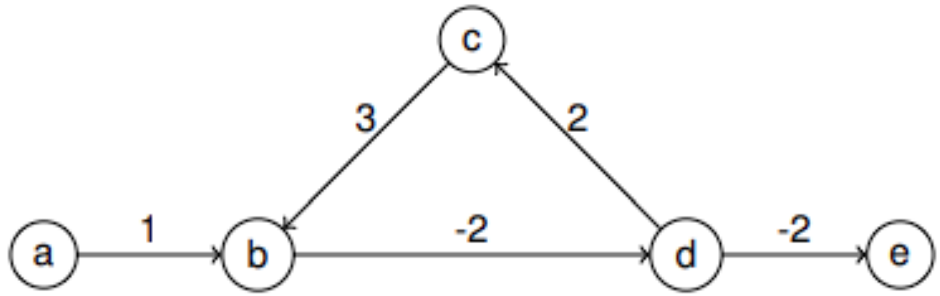


Abbildung: Graph ohne negativem Zyklus

1. Bewertungsmatrix

$$B = \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ & & & & \\ & & & & \\ & & & & \end{pmatrix}$$

Bellman-Ford-Algorithmus

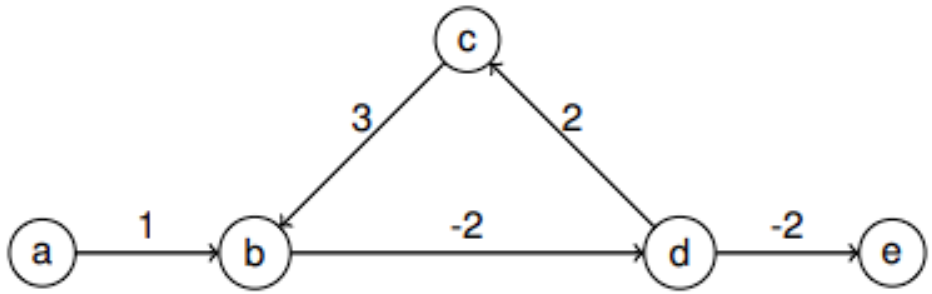


Abbildung: Graph ohne negativem Zyklus

1. Bewertungsmatrix

$$B = \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ & & & & \\ & & & & \end{pmatrix}$$

Bellman-Ford-Algorithmus

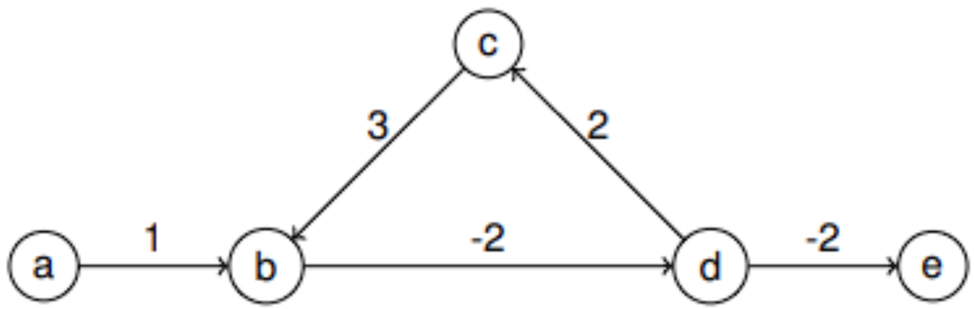


Abbildung: Graph ohne negativem Zyklus

1. Bewertungsmatrix

$$B = \left(\begin{matrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ & & & & \end{matrix} \right.$$

Bellman-Ford-Algorithmus

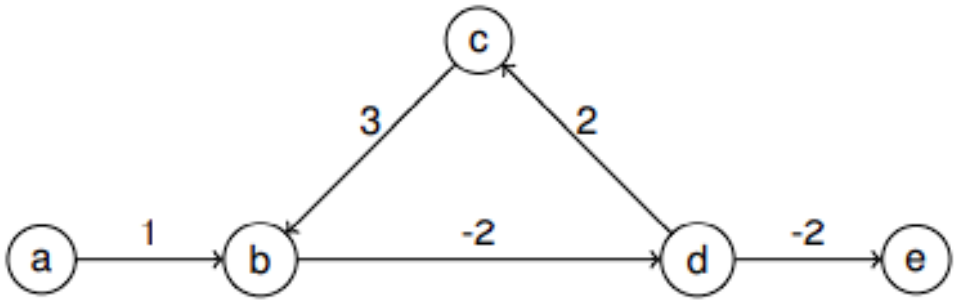


Abbildung: Graph ohne negativem Zyklus

1. Bewertungsmatrix

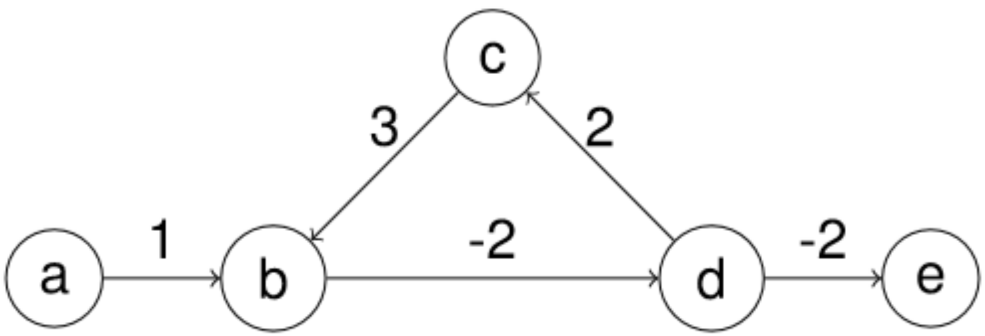
$$B = \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix};$$

Bellman-Ford-Algorithmus

2. Tree Matrix T:

t_{ij} speichert den <u>Vorgängerknoten</u> auf dem kürzesten Weg von Knoten i nach j.

$$T^{(1)} = \begin{pmatrix} - & & & & \\ & - & & & \\ & & - & & \\ & & & - & \\ & & & & - \end{pmatrix}$$



Bellman-Ford-Algorithmus

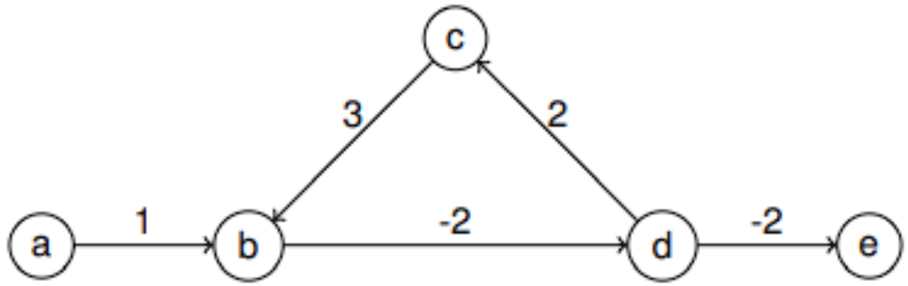


Abbildung: Graph ohne negativem Zyklus

1. Bewertungsmatrix

$$B = \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}; T^1 = \left(\begin{matrix} -1 & A & -1 & -1 & -1 \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{matrix} \right.$$

Bellman-Ford-Algorithmus

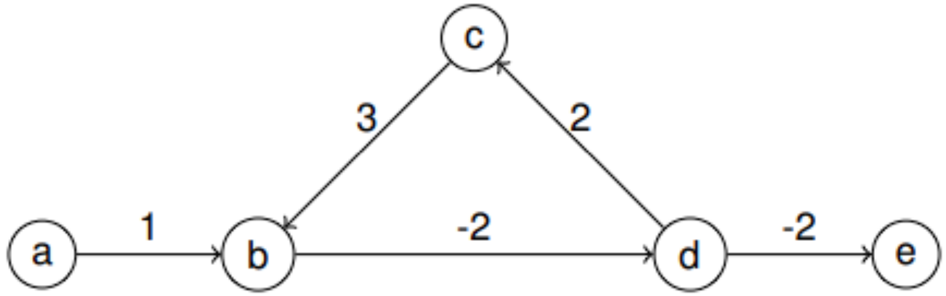


Abbildung: Graph ohne negativem Zyklus

1. Bewertungsmatrix

$$B = \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} ; T^1 = \begin{pmatrix} -1 & A & -1 & -1 & -1 \\ -1 & -1 & -1 & B & -1 \\ & & & & \\ & & & & \\ & & & & \end{pmatrix}$$

Bellman-Ford-Algorithmus

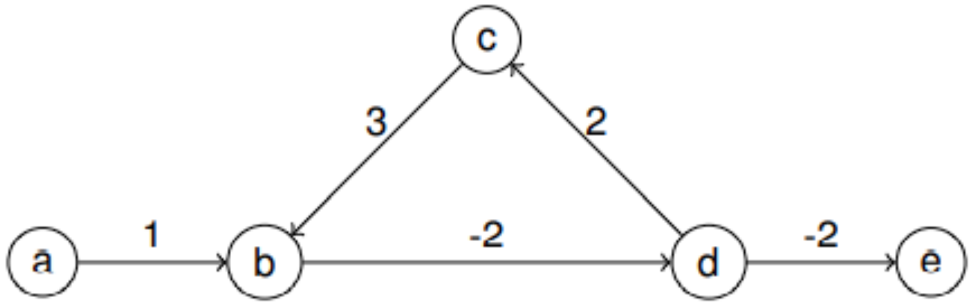


Abbildung: Graph ohne negativem Zyklus

1. Bewertungsmatrix

$$B = \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}; T^1 = \left(\begin{matrix} -1 & A & -1 & -1 & -1 \\ -1 & -1 & -1 & B & -1 \\ -1 & C & -1 & -1 & -1 \\ & & & & \\ & & & & \end{matrix} \right.$$

Bellman-Ford-Algorithmus

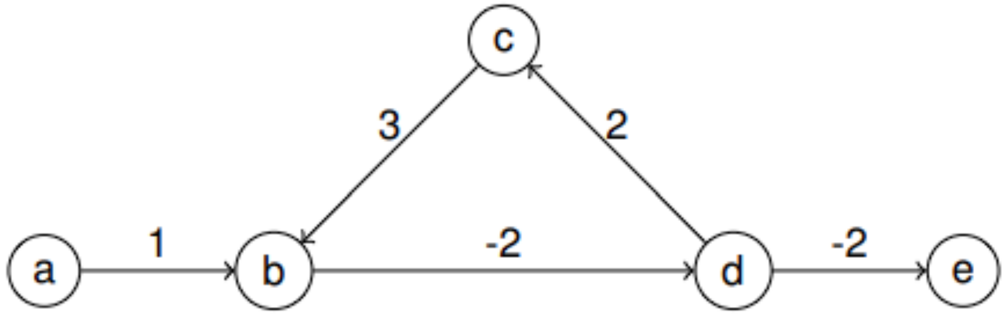


Abbildung: Graph ohne negativem Zyklus

1. Bewertungsmatrix

$$B = \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}; T^1 = \begin{pmatrix} -1 & A & -1 & -1 & -1 \\ -1 & -1 & -1 & B & -1 \\ -1 & C & -1 & -1 & -1 \\ -1 & -1 & D & -1 & D \end{pmatrix}$$

Bellman-Ford-Algorithmus

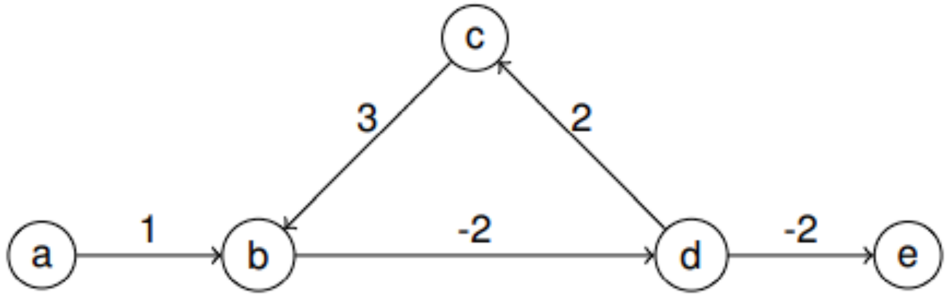


Abbildung: Graph ohne negativem Zyklus

1. Bewertungsmatrix

$$B = \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}; T^1 = \begin{pmatrix} -1 & A & -1 & -1 & -1 \\ -1 & -1 & -1 & B & -1 \\ -1 & C & -1 & -1 & -1 \\ -1 & -1 & D & -1 & D \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

Bellman-Ford-Algorithmus

Berechnungsvorschrift

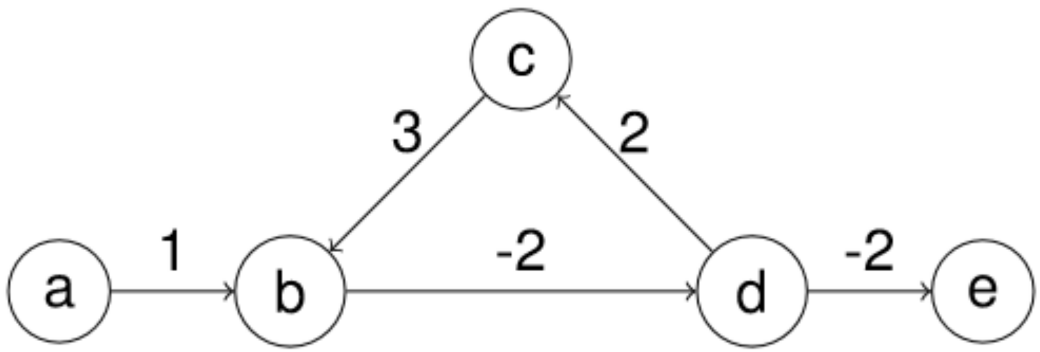
$$\boxed{U^{(m)} \otimes B =}$$

$$U^{(m+1)}$$

$\otimes$: „Bellman-Multiplikation"

$$\Longrightarrow \; u_{ij}^{(m+1)} = min_k \left[u_{ik}^{(m)} + \; b_{kj} \right] \quad für\ alle\ m \geq 1$$

$U^{(m)}$: enthält die Längen aller kürzesten Wege mit max. m Kanten



Bellman-Ford-Algorithmus

Abbruchkriterien

- **$U^{(m+1)} = U^{(m)}$ → keine Verbesserung**
- **Mindestens einen negativen Eintrag auf der Hauptdiagonalen von U → Hinweis auf negative Zyklen**
- **Algorithmus muss maximal |V|-1 Mal durchlaufen werden, danach nur noch ein weiteres Mal zum Test auf negative Zyklen**

终止条件:

- U(m+1) = U(m)：没有改善
- U 的主对角线上至少有一个负条目：指示存在负环
- 算法必须最多执行 |V|-1 次，之后只需再执行一次以检测负环。

Bellman-Ford-Algorithmus

1. Iteration $U^2 = U^1 \otimes B$

$$U^2 = \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix};$$

$$u^2_{a,a} = \min[0 + 0, 1 + \infty, \infty + \infty, \infty + \infty, \infty + \infty] = 0$$

Bellman-Ford-Algorithmus

1. Iteration $U^2 = U^1 \otimes B$

$$U^2 = \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix};$$

$$u^2_{a,b} = \min[0+1, 1+0, \infty+3, \infty+\infty, \infty+\infty] = 1$$

Bellman-Ford-Algorithmus

1. Iteration $U^2 = U^1 \otimes B$

$$U^2 = \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix};$$

$$u^2_{a,d} = \min[0 + \infty, 1 + (-2), \infty + \infty, \infty + 0, \infty + \infty] = -1$$

Bellman-Ford-Algorithmus

1. Iteration $U^2 = U^1 \otimes B$

$$U^2 = \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix};$$

$$u^2_{a,d} = \min[0 + \infty, 1 + (-2), \infty + \infty, \infty + 0, \infty + \infty] = -1$$

$$U^2 = \begin{pmatrix} 0 & 1 & \infty & \boxed{-1} & \infty \\ \infty & 0 & 0 & -2 & -4 \\ \infty & 3 & 0 & 1 & \infty \\ \infty & 5 & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}; T^2 = \begin{pmatrix} -1 & A & -1 & \boxed{B} & -1 \\ -1 & -1 & D & B & D \\ -1 & C & -1 & B & -1 \\ -1 & C & D & -1 & D \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

Bellman-Ford-Algorithmus

1. Iteration $U^2 = U^1 \otimes B$

$$U^2 = \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix};$$

$$u^2_{b,c} = \min[\infty + \infty, 0 + \infty, \infty + 0, -2 + 2, \infty + \infty] = 0$$

$$U^2 = \begin{pmatrix} 0 & 1 & \infty & -1 & \infty \\ \infty & 0 & 0 & -2 & -4 \\ \infty & 3 & 0 & 1 & \infty \\ \infty & 5 & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}; T^2 = \begin{pmatrix} -1 & A & -1 & B & -1 \\ -1 & -1 & D & B & D \\ -1 & C & -1 & B & -1 \\ -1 & C & D & -1 & D \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

Bellman-Ford-Algorithmus

1. Iteration $U^2 = U^1 \otimes B$

$$U^2 = \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix};$$

$$u^2_{b,e} = \min[\infty + \infty, 0 + \infty, \infty + \infty, -2 + (-2), \infty + 0] = -4$$

$$U^2 = \begin{pmatrix} 0 & 1 & \infty & -1 & \infty \\ \infty & 0 & 0 & -2 & -4 \\ \infty & 3 & 0 & 1 & \infty \\ \infty & 5 & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}; T^2 = \begin{pmatrix} -1 & A & -1 & B & -1 \\ -1 & -1 & D & B & D \\ -1 & C & -1 & B & -1 \\ -1 & C & D & -1 & D \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

Bellman-Ford-Algorithmus

1. Iteration $U^2 = U^1 \otimes B$

$$U^2 = \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix};$$

$$u^2_{c,d} = \min[\infty + \infty, 3 + (-2), 0 + \infty, \infty + 0, \infty + \infty] = 1$$

$$U^2 = \begin{pmatrix} 0 & 1 & \infty & -1 & \infty \\ \infty & 0 & 0 & -2 & -4 \\ \infty & 3 & 0 & 1 & \infty \\ \infty & 5 & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}; T^2 = \begin{pmatrix} -1 & A & -1 & B & -1 \\ -1 & -1 & D & B & D \\ -1 & C & -1 & B & -1 \\ -1 & C & D & -1 & D \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

Bellman-Ford-Algorithmus

1. Iteration $U^2 = U^1 \otimes B$

$$U^2 = \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix};$$

$$u^2_{d,b} = \min[\infty + 1, \infty + 0, 2 + 3, 0 + \infty, -2 + \infty] = 5$$

$$U^2 = \begin{pmatrix} 0 & 1 & \infty & -1 & \infty \\ \infty & 0 & 0 & -2 & -4 \\ \infty & 3 & 0 & 1 & \infty \\ \infty & 5 & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}; T^2 = \begin{pmatrix} -1 & A & -1 & B & -1 \\ -1 & -1 & D & B & D \\ -1 & C & -1 & B & -1 \\ -1 & C & D & -1 & D \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

Bellman-Ford-Algorithmus

2. Iteration $U^3 = U^2 \otimes B$

$$U^3 = \begin{pmatrix} 0 & 1 & \infty & -1 & \infty \\ \infty & 0 & 0 & -2 & -4 \\ \infty & 3 & 0 & 1 & \infty \\ \infty & 5 & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix};$$

$u^3_{a,c} =$

Bellman-Ford-Algorithmus

2. Iteration $U^3 = U^2 \otimes B$

$$U^3 = \begin{pmatrix} 0 & 1 & \infty & -1 & \infty \\ \infty & 0 & 0 & -2 & -4 \\ \infty & 3 & 0 & 1 & \infty \\ \infty & 5 & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix};$$

$$u^3_{a,c} = \min[0+\infty, 1+\infty, \infty+0, -1+2, \infty+\infty] = 1$$

Bellman-Ford-Algorithmus

2. Iteration $U^3 = U^2 \otimes B$

$$U^3 = \begin{pmatrix} 0 & 1 & \infty & -1 & \infty \\ \infty & 0 & 0 & -2 & -4 \\ \infty & 3 & 0 & 1 & \infty \\ \infty & 5 & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix};$$

$$u^3_{a,c} = \min[0 + \infty, 1 + \infty, \infty + 0, -1 + 2, \infty + \infty] = 1$$

$$U^3 = \begin{pmatrix} 0 & 1 & 1 & -1 & -3 \\ \infty & 0 & 0 & -2 & -4 \\ \infty & 3 & 0 & 1 & -1 \\ \infty & 5 & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}; T^3 = \begin{pmatrix} -1 & A & \mathrm{D} & B & D \\ -1 & -1 & D & B & D \\ -1 & C & -1 & B & D \\ -1 & C & D & -1 & D \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

Bellman-Ford-Algorithmus

2. Iteration $U^3 = U^2 \otimes B$

$$U^3 = \begin{pmatrix} 0 & 1 & \infty & -1 & \infty \\ \infty & 0 & 0 & -2 & -4 \\ \infty & 3 & 0 & 1 & \infty \\ \infty & 5 & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix};$$

Bellman-Ford-Algorithmus

2. Iteration $U^3 = U^2 \otimes B$

$$U^3 = \begin{pmatrix} 0 & 1 & \infty & -1 & \infty \\ \infty & 0 & 0 & -2 & -4 \\ \infty & 3 & 0 & 1 & \infty \\ \infty & 5 & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix};$$

$$u^3_{a,e} = \min[0+\infty, 1+\infty, \infty+\infty, -1+(-2), \infty+0] = -3$$

$$U^3 = \begin{pmatrix} 0 & 1 & 1 & -1 & \text{-3} \\ \infty & 0 & 0 & -2 & -4 \\ \infty & 3 & 0 & 1 & -1 \\ \infty & 5 & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}; T^3 = \begin{pmatrix} -1 & A & D & B & \mathrm{D} \\ -1 & -1 & D & B & D \\ -1 & C & -1 & B & D \\ -1 & C & D & -1 & D \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

Bellman-Ford-Algorithmus

2. Iteration $U^3 = U^2 \otimes B$

$$U^3 = \begin{pmatrix} 0 & 1 & \infty & -1 & \infty \\ \infty & 0 & 0 & -2 & -4 \\ \infty & 3 & 0 & 1 & \infty \\ \infty & 5 & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix};$$

Bellman-Ford-Algorithmus

2. Iteration $U^3 = U^2 \otimes B$

$$U^3 = \begin{pmatrix} 0 & 1 & \infty & -1 & \infty \\ \infty & 0 & 0 & -2 & -4 \\ \infty & 3 & 0 & 1 & \infty \\ \infty & 5 & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix};$$

$$u^3_{c,e} = \min[\infty + \infty, 3 + \infty, 0 + \infty, 1 + (-2), \infty + 0] = -1$$

$$U^3 = \begin{pmatrix} 0 & 1 & 1 & -1 & -3 \\ \infty & 0 & 0 & -2 & -4 \\ \infty & 3 & 0 & 1 & \text{-1} \\ \infty & 5 & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}; T^3 = \begin{pmatrix} -1 & A & D & B & D \\ -1 & -1 & D & B & D \\ -1 & C & -1 & B & \text{D} \\ -1 & C & D & -1 & D \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

Bellman-Ford-Algorithmus

3. Iteration $U^4 = U^3 \otimes B$

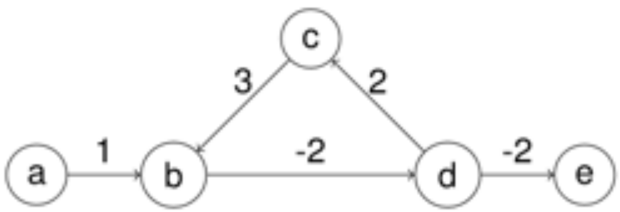
$$U^4 = \begin{pmatrix} 0 & 1 & 1 & -1 & -3 \\ \infty & 0 & 0 & -2 & -4 \\ \infty & 3 & 0 & 1 & -1 \\ \infty & 5 & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix};$$

$$U^4 = \begin{pmatrix} 0 & 1 & 1 & -1 & -3 \\ \infty & 0 & 0 & -2 & -4 \\ \infty & 3 & 0 & 1 & -1 \\ \infty & 5 & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}; T^4 = \begin{pmatrix} -1 & A & D & B & D \\ -1 & -1 & D & B & D \\ -1 & C & -1 & B & D \\ -1 & C & D & -1 & D \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

In der letzten Iteration trat keine Verbesserung auf. $\Rightarrow$ **Abbruch!**

3.12 Bellman-Ford-Algorithmus I

a) Finden Sie den kürzesten Weg von a nach e mit dem Bellman-Ford-Algorithmus



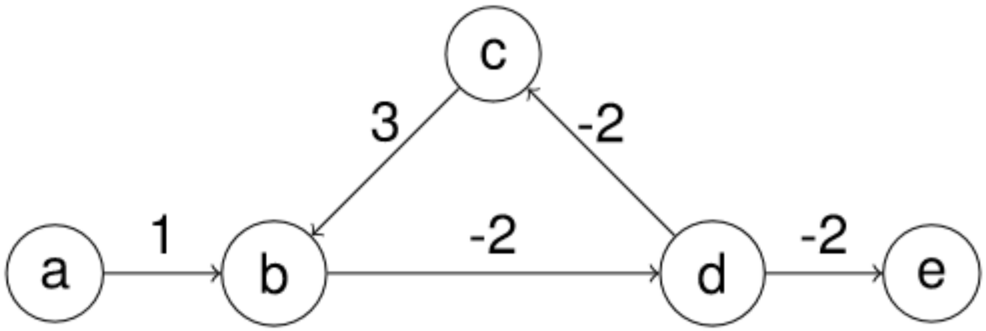
$$U^4 = \begin{pmatrix} 0 & 1 & 1 & -1 & \mathbf{-3} \\ \infty & 0 & 0 & -2 & -4 \\ \infty & 3 & 0 & 1 & -1 \\ \infty & 5 & 2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} ; T^4 = \begin{pmatrix} -1 & \mathbf{A} & D & \mathbf{B} & \mathbf{D} \\ -1 & -1 & D & B & D \\ -1 & C & -1 & B & D \\ -1 & C & D & -1 & D \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

Der kürzeste Weg von a nach e lautet: a → b → d → e

Die Weglänge beträgt -3

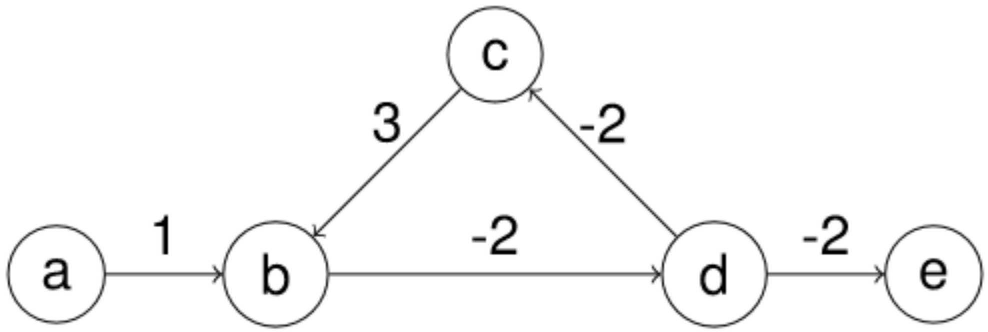
3.12 Bellman-Ford-Algorithmus I

b) Finden Sie erneut den kürzesten Weg von a nach e mit dem Bellman-Ford-Algorithmus. Achten Sie bei der Ausführung auf negative Zyklen.



3.12 Bellman-Ford-Algorithmus I

b) Es existieren keine kürzesten Wege, weil negative Einträge auf der Hauptdiagonalen vorhanden sind.



$$U^3 = \begin{pmatrix} 0 & 1 & -3 & -1 & -3 \\ \infty & -1 & -4 & -2 & -4 \\ \infty & 3 & -1 & 1 & -1 \\ \infty & 1 & -2 & -1 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}$$

3.12 Bellman-Ford-Algorithmus I

c) Führen Sie die Bellman-Multiplikation für **Zeile d** von U² und **Spalte d** von B durch und interpretieren Sie Ihr Ergebnis inhaltlich.

$$\underbrace{\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}\begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 0 & 1 & \infty & -1 & \infty \\ \infty & 0 & -4 & -2 & -4 \\ \infty & 3 & 0 & 1 & \infty \\ \infty & 1 & -2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}\end{array}}_{U^2} \otimes \underbrace{\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}\begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & -2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix}\end{array}}_{B}$$

3.12 Bellman-Ford-Algorithmus I

c) Führen Sie die Bellman-Multiplikation für Zeile d von U^3 und Spalte d von B durch und interpretieren Sie Ihr Ergebnis inhaltlich.

$$\underbrace{\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}\begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 0 & 1 & \infty & -1 & \infty \\ \infty & 0 & -4 & -2 & -4 \\ \infty & 3 & 0 & 1 & \infty \\ \boxed{\infty & 1 & -2 & 0 & -2} \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \end{array}}_{U^3} \otimes \underbrace{\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}\begin{array}{c} \begin{array}{ccccc} a & b & c & \boxed{d} & e \end{array} \\ \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & -2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \end{array}}_{B}$$

3.12 Bellman-Ford-Algorithmus I

c) Es befindet sich in der neuen Bewertungsmatrix nun ein negativer Eintrag auf der Hauptdiagonalen. Das bedeutet, es entsteht ein negativer Zyklus und der Algorithmus muss abgebrochen werden.

$$\min\left[\infty+\infty; 1+(-2); (-2)+\infty; 0+0; (-2)+\infty\right] = -1$$

$$\underbrace{\begin{array}{c|ccccc} & a & b & c & d & e \\ \hline a & 0 & 1 & \infty & -1 & \infty \\ b & \infty & 0 & -4 & -2 & -4 \\ c & \infty & 3 & 0 & 1 & \infty \\ \boxed{d} & \infty & 1 & -2 & 0 & -2 \\ e & \infty & \infty & \infty & \infty & 0 \end{array}}_{U^3} \otimes \underbrace{\begin{array}{c|ccccc} & a & b & c & \boxed{d} & e \\ \hline a & 0 & 1 & \infty & \infty & \infty \\ b & \infty & 0 & \infty & -2 & \infty \\ c & \infty & 3 & 0 & \infty & \infty \\ d & \infty & \infty & -2 & 0 & -2 \\ e & \infty & \infty & \infty & \infty & 0 \end{array}}_{B}$$

3.12 Bellman-Ford-Algorithmus I

c. Führen Sie die Bellman-Multiplikation für Zeile d von U^2 und Spalte d von B durch und interpretieren Sie Ihr Ergebnis inhaltlich.

$$\underbrace{\begin{array}{c c} & \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \begin{pmatrix} 0 & 1 & \infty & -1 & \infty \\ \infty & 0 & -4 & -2 & -4 \\ \infty & 3 & 0 & 1 & \infty \\ \infty & 1 & -2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \end{array}}_{U^2} \otimes \underbrace{\begin{array}{c c} & \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \begin{pmatrix} 0 & 1 & \infty & \infty & \infty \\ \infty & 0 & \infty & -2 & \infty \\ \infty & 3 & 0 & \infty & \infty \\ \infty & \infty & -2 & 0 & -2 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \end{array}}_{B}$$

Die Durchführung der Bellmann-Multiplikation ergibt folgendes Ergebnis

$$\min\left[\infty + \infty; 1 + (-2); (-2) + \infty; 0 + 0; (-2) + \infty\right] = -1 \tag{1}$$

Es befindet sich in der neuen Bewertungsmatrix nun ein negativer Eintrag auf der Hauptdiagonalen. Das bedeutet es entsteht ein negativer Zyklus und der Algorithmus muss abgebrochen werden.

Altklausur: SS19 2. Termin – Aufgabe 5. d)

Kann ein Graph mit den drei Knoten a, b, c existieren für den die $U^{(2)}$-Matrix die Werte:

$$\boldsymbol{u_{aa}^{(2)} = -1 \qquad u_{bb}^{(2)} = 0 \qquad u_{cc}^{(2)} = -2}$$

annimmt?

Begründen Sie ihre Antwort. (2 Punkte)

Antwort:

Nein ein solcher Graph kann nicht existieren! Wenn sich ein negativer Kreis bildet, sind daran immer mehrere Knoten beteiligt, welche den selben negativen Wert auf der Hauptdiagonalen der U-Matrix aufweisen müssen.

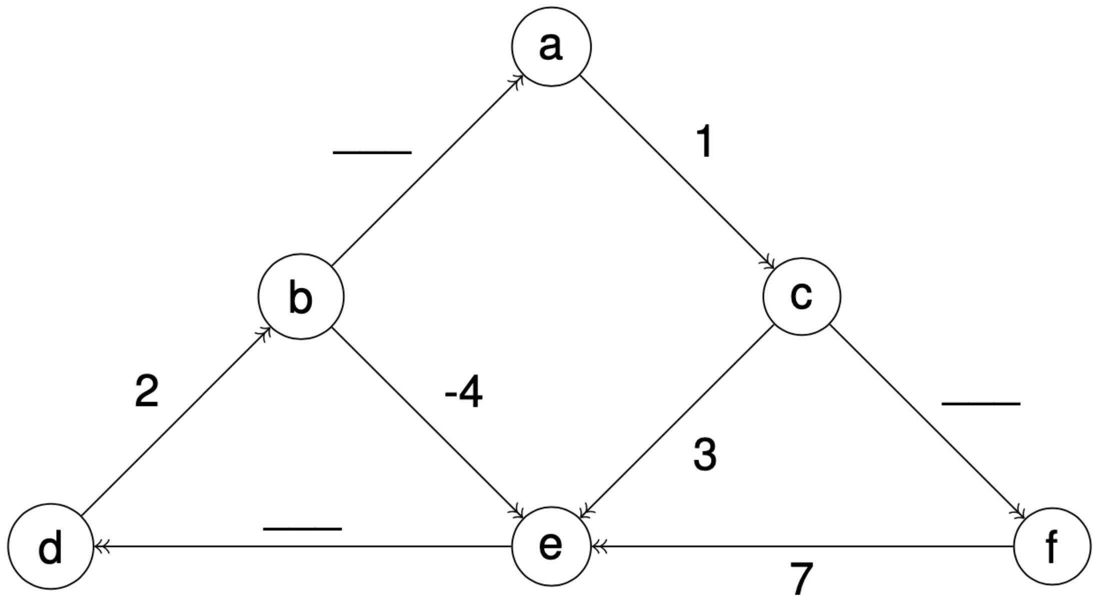
Generell:

Wenn Knoten x mit Knoten y einen Kreis bildet, gilt: $\boldsymbol{u_{xx}^{(2)} = u_{yy}^{(2)}}$

Altklausur: SS19 – 1. Termin – Aufgabe 5e)

e) Ergänzen Sie die fehlenden Kantengewichte so, dass bei Anwendung des Bellman-Ford-Algorithmus negative Einträge auf der Hauptdiagonalen zum ersten Mal in der $U^{(5)}$ Matrix auftreten.

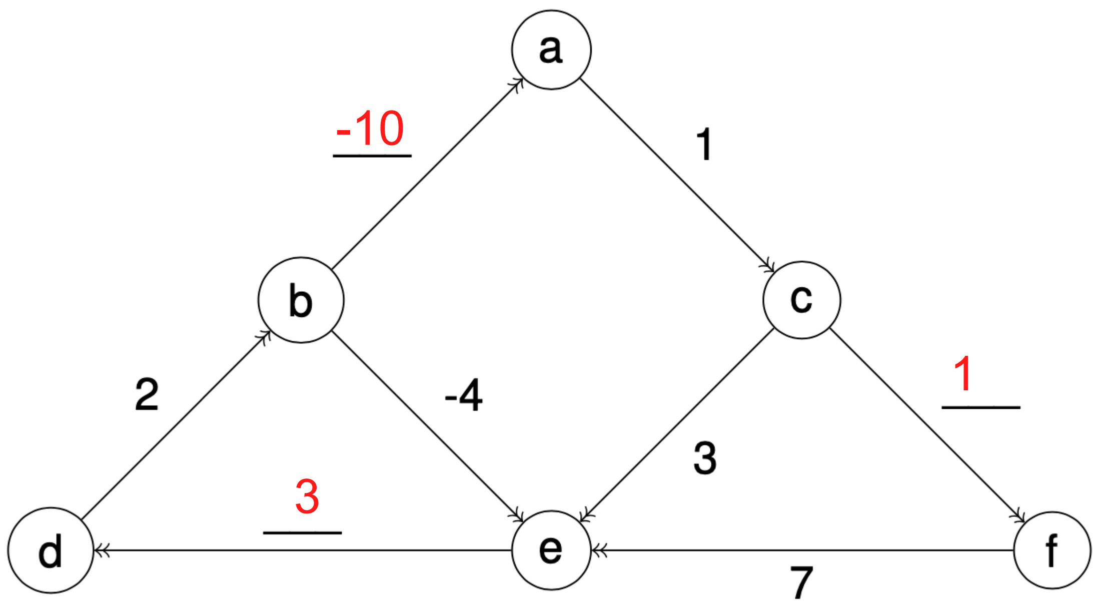
Punkte: / 1



Altklausur: SS19 – 1. Termin – Aufgabe 5e)

e) Ergänzen Sie die fehlenden Kantengewichte so, dass bei Anwendung des Bellman-Ford-Algorithmus negative Einträge auf der Hauptdiagonalen zum ersten Mal in der $U^{(5)}$ Matrix auftreten.

Punkte: / 1



Fragen zum Tutorium?