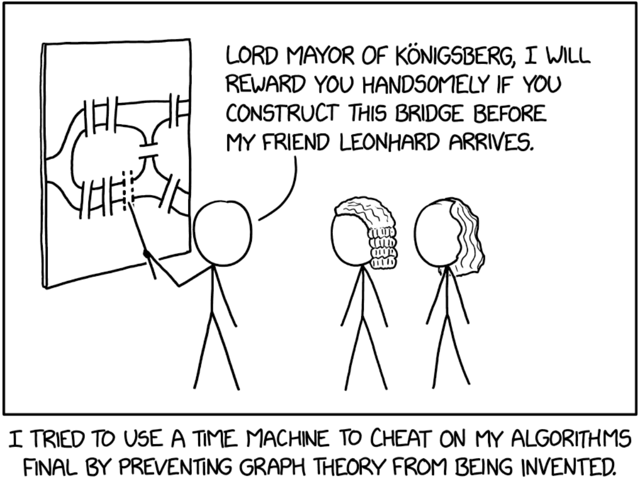
ETH zürich

D&A - Übungsstunde 11

Diese Folien basieren auf denjenigen der Vorlesung, wurden aber durch den Assistenten Adel Gavranović adaptiert und erweitert

Comic der Woche



xkcd

Übersicht

Heutiges Programm

Intro
Follow-up
Feedback zu [code]expert
Graphrecap
All-pairs Shortest Path Problem
Live [code]expert
In-Class-Exercise (theoretisch)
Alte Prüfungsfragen (26.1.2018)
Outro

n.ethz.ch/~agavranovic

- Link zum Material für die Übungsstunden
- Webseite des Assistenten
- Mail an Assistenten

1. Intro

Intro

Intro

- Tasks 9-13 sind relevant um Bonusaufgabe freizuschalten

2. Follow-up

Follow-up aus vorherigen Übungsstunden

Follow-up aus vorherigen Übungsstunden

- Um mit einer DFS die Distanzen zu allen Knoten zu bestimmen, muss man noch ein zusätzliches Array führen

3. Feedback zu [code]expert

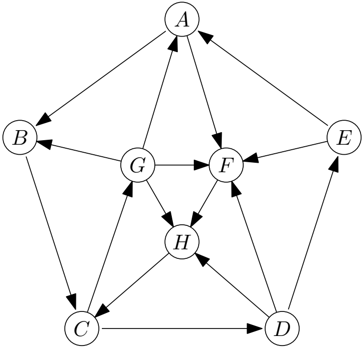
Allgemeines zu [code]expert

Allgemeines zu [code]expert

- Wird alles dieses Wochenende korrigiert
- Sorry für die Wartezeit

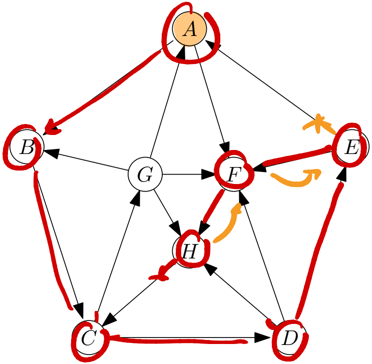
Fragen zu [code]expert eurerseits?

Tiefen- und Breitensuche



Start bei A

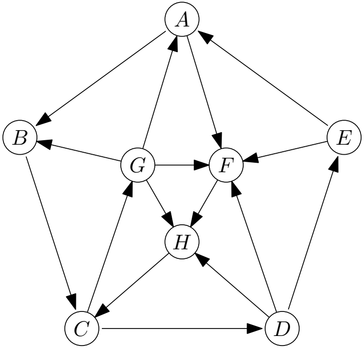
Tiefen- und Breitensuche



Start bei A

DFS: A, B, C, D, E, F,

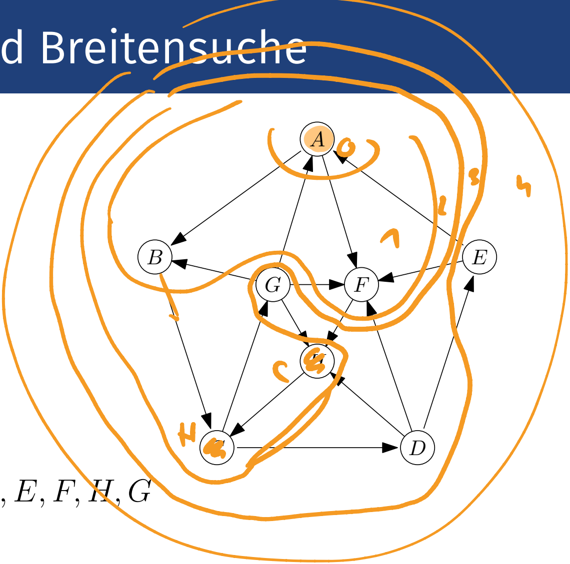
Tiefen- und Breitensuche



Start bei A

DFS: A, B, C, D, E, F, H, G

Tiefen- und Breitensuche



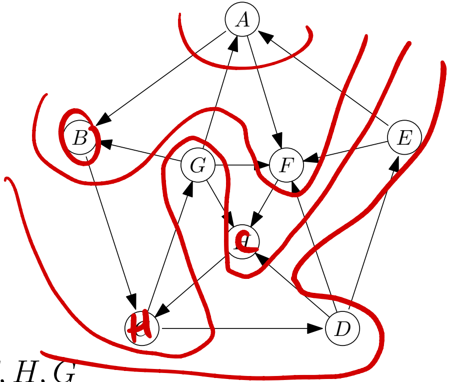
A
B

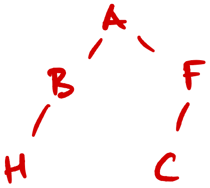
Start bei A

DFS: A, B, C, D, E, F, H, G

BFS:

Tiefen- und Breitensuche





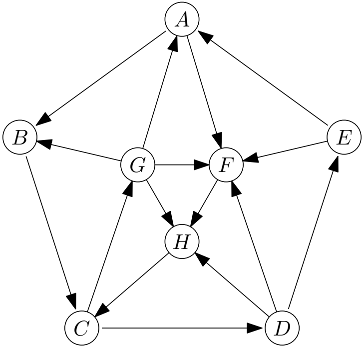
Start bei A

DFS: A, B, C, D, E, F, H, G

BFS: A, B, F, C, H, D, G, E

A, B, F, C, H, D, G, E

Tiefen- und Breitensuche



Start bei A

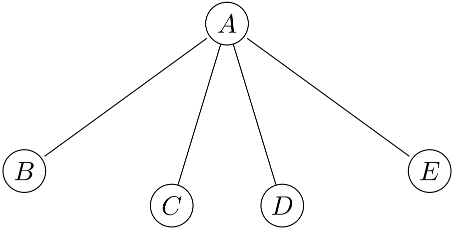
DFS: A, B, C, D, E, F, H, G

BFS: A, B, F, C, H, D, G, E

Es gibt keinen Startknoten, sodass die DFS-Ordnung der BFS-Ordnung entspricht.

Tiefen- und Breitensuche

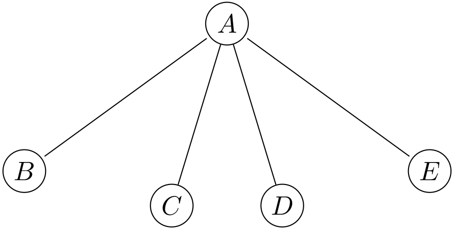
Stern: DFS-Ordnung entspricht BFS-Ordnung



Start bei A

Tiefen- und Breitensuche

Stern: DFS-Ordnung entspricht BFS-Ordnung

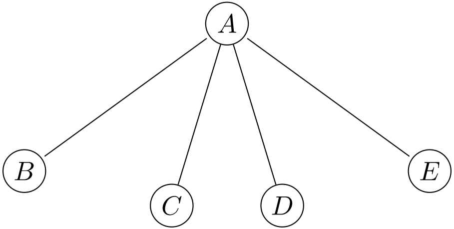


Start bei A

DFS:

Tiefen- und Breitensuche

Stern: DFS-Ordnung entspricht BFS-Ordnung

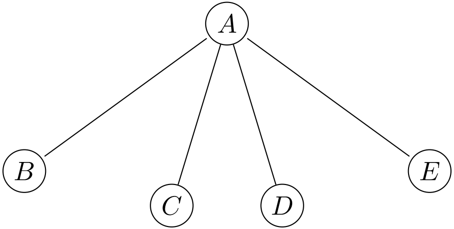


Start bei A

DFS: A, B, C, D, E

Tiefen- und Breitensuche

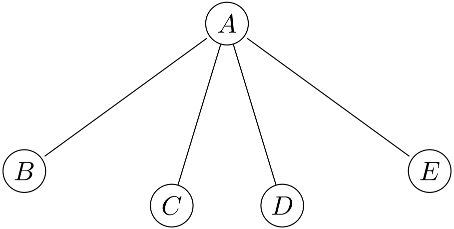
Stern: DFS-Ordnung entspricht BFS-Ordnung



Start bei A
DFS: A, B, C, D, E
BFS:

Tiefen- und Breitensuche

Stern: DFS-Ordnung entspricht BFS-Ordnung



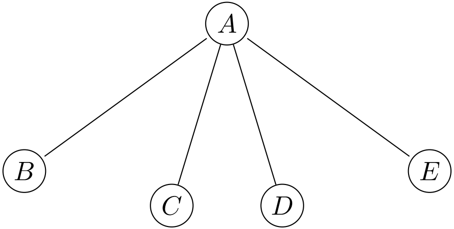
Start bei A

DFS: A, B, C, D, E

BFS: A, B, C, D, E

Tiefen- und Breitensuche

Stern: DFS-Ordnung entspricht BFS-Ordnung



Start bei A

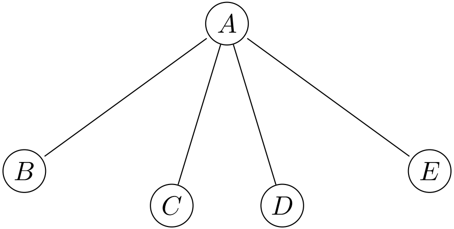
DFS: A, B, C, D, E

BFS: A, B, C, D, E

Start bei C

Tiefen- und Breitensuche

Stern: DFS-Ordnung entspricht BFS-Ordnung

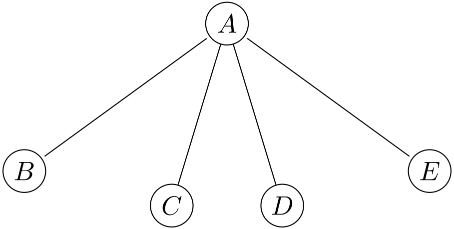


Start bei A
DFS: A, B, C, D, E
BFS: A, B, C, D, E

Start bei C
DFS:

Tiefen- und Breitensuche

Stern: DFS-Ordnung entspricht BFS-Ordnung

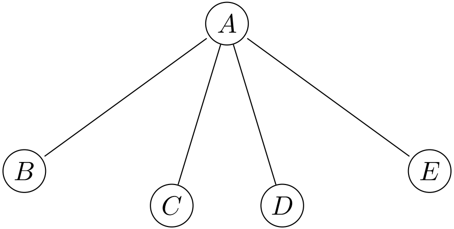


Start bei A
DFS: A, B, C, D, E
BFS: A, B, C, D, E

Start bei C
DFS: C, A, B, D, E

Tiefen- und Breitensuche

Stern: DFS-Ordnung entspricht BFS-Ordnung

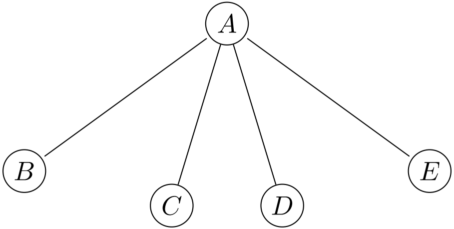


Start bei A
DFS: A, B, C, D, E
BFS: A, B, C, D, E

Start bei C
DFS: C, A, B, D, E
BFS:

Tiefen- und Breitensuche

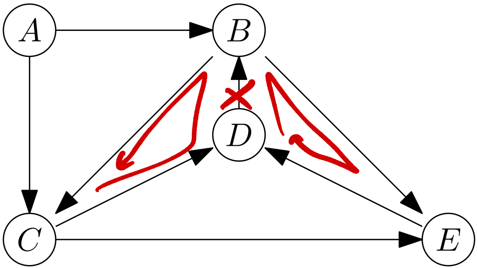
Stern: DFS-Ordnung entspricht BFS-Ordnung



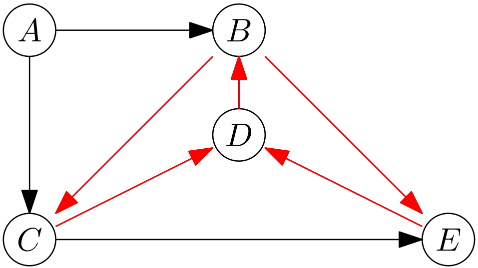
Start bei A
DFS: A, B, C, D, E
BFS: A, B, C, D, E

Start bei C
DFS: C, A, B, D, E
BFS: C, A, B, D, E

Topologische Sortierung

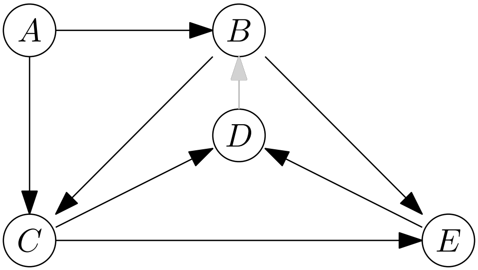


Topologische Sortierung



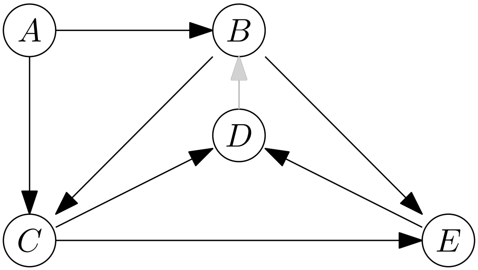
- der Graph hat Kreise

Topologische Sortierung



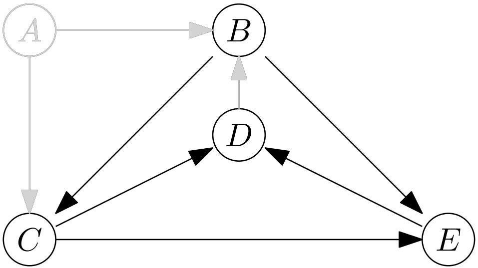
- der Graph hat Kreise
- zwei minimale Kreise teilen sich eine Kante

Topologische Sortierung



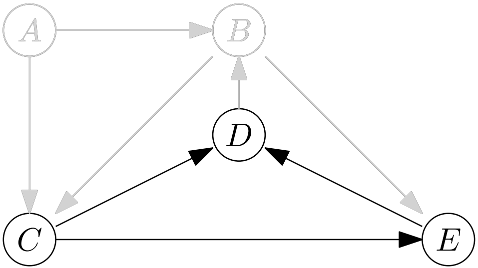
- der Graph hat Kreise
- zwei minimale Kreise teilen sich eine Kante
- entferne die Kante $\Longrightarrow$ kreisfrei

Topologische Sortierung



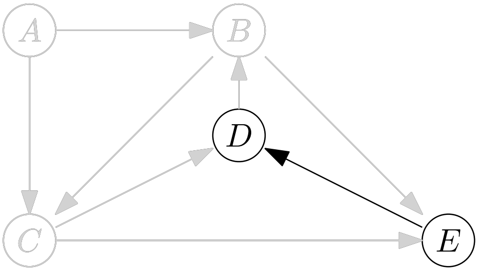
- der Graph hat Kreise
- zwei minimale Kreise teilen sich eine Kante
- entferne die Kante $\implies$ kreisfrei
- topologische Sortierung durch „Entfernen“ von Knoten mit Eingangsgrad 0

Topologische Sortierung



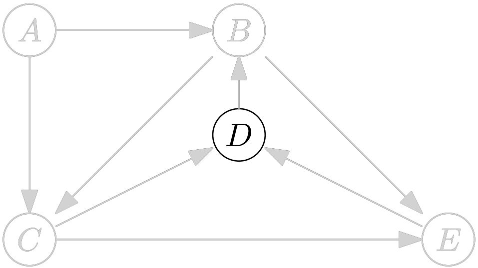
- der Graph hat Kreise
- zwei minimale Kreise teilen sich eine Kante
- entferne die Kante $\implies$ kreisfrei
- topologische Sortierung durch „Entfernen“ von Knoten mit Eingangsgrad 0

Topologische Sortierung



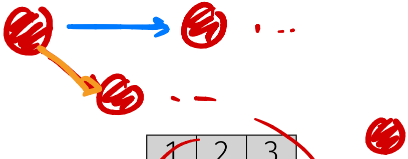
- der Graph hat Kreise
- zwei minimale Kreise teilen sich eine Kante
- entferne die Kante $\implies$ kreisfrei
- topologische Sortierung durch „Entfernen“ von Knoten mit Eingangsgrad 0

Topologische Sortierung



- der Graph hat Kreise
- zwei minimale Kreise teilen sich eine Kante
- entferne die Kante $\implies$ kreisfrei
- topologische Sortierung durch „Entfernen" von Knoten mit Eingangsgrad 0

Sliding Puzzle

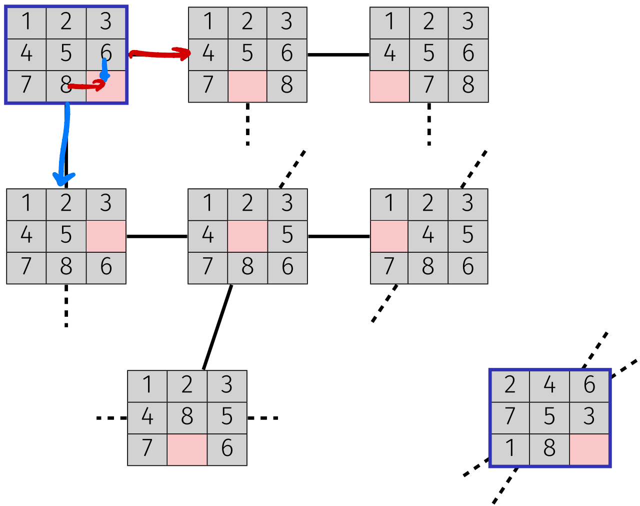


Schnelleste Lösung finden für

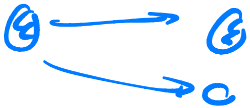
2	4	6
7	5	3
1	8	

1	2	3
4	5	6
7	8	

Problem als Graph



Fragen/Unklarheiten?



5. All-pairs Shortest Path Problem

DP-Algorithmus Floyd-Warshall(G)

Input: Azyklischer Graph $G = (V, E, c)$
Output: Minimale Gewichte aller Pfade d

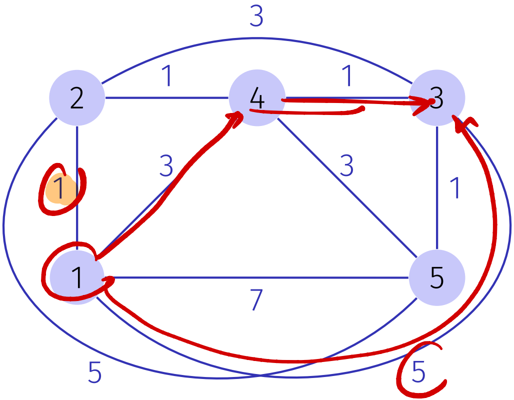
$d^0 \leftarrow c$
for $k \leftarrow 1$ **to** $|V|$ **do**
  for $i \leftarrow 1$ **to** $|V|$ **do**
    for $j \leftarrow 1$ **to** $|V|$ **do**
      $d^k(v_i, v_j) = \min\{d^{k-1}(v_i, v_j), d^{k-1}(v_i, v_k) + d^{k-1}(v_k, v_j)\}$

Laufzeit: $\Theta(|V|^3)$

Bemerkung: Der Algorithmus kann auf einer einzigen Matrix d (in place) ausgeführt werden.

▸ Videoempfehlung

Beispiel



Adjazenzmatrix $M = c$

	1	2	3	4	5
1	0	1	5	3	7
2	1	0	3	1	5
3	5	3	0	1	1
4	3	1	1	0	3
5	7	5	1	3	0

$i = 1$ $j = 3$ $k = 4$

Beispiel

$k = 1$

0	1	5	3	7
1	0	3	1	5
5	3	0	1	1
3	1	1	0	3
7	5	1	3	0

d^0

0	1	5	3	7
1	0	3	1	5
5	3	0	1	1
3	1	1	0	3
7	5	1	3	0

d^1

$d^0 \leftarrow c$
for $k \leftarrow 1$ **to** $|V|$ **do**
  for $i \leftarrow 1$ **to** $|V|$ **do**
    for $j \leftarrow 1$ **to** $|V|$ **do**
      $d^k(v_i, v_j) = \min\{d^{k-1}(v_i, v_j), d^{k-1}(v_i, v_k) + d^{k-1}(v_k, v_j)\}$

Beispiel

$k = 1$

0	1	5	3	7
1	0	3	1	5
5	3	0	1	1
3	1	1	0	3
7	5	1	3	0

d^0

$k = 2$

0	1	5	3	7
1	0	3	1	5
5	3	0	1	1
3	1	1	0	3
7	5	1	3	0

d^1

$d^0 \leftarrow c$
for $k \leftarrow 1$ **to** $|V|$ **do**
  for $i \leftarrow 1$ **to** $|V|$ **do**
    for $j \leftarrow 1$ **to** $|V|$ **do**
      $d^k(v_i, v_j) = \min\{d^{k-1}(v_i, v_j), d^{k-1}(v_i, v_k) + d^{k-1}(v_k, v_j)\}$

$\boxed{1 \to 3}$, $1 \to 2 + 2 \to 3$

0	1	5	3	7
1	0	3	1	5
5	3	0	1	1
3	1	1	0	3
7	5	1	3	0

d^1

0	1	$\boxed{5}$	3	7
1	0	3	1	5
5	3	0	1	1
3	1	1	0	3
4	5	1	3	0

d^2

Beispiel

$k = 1$

0	1	5	3	7
1	0	3	1	5
5	3	0	1	1
3	1	1	0	3
7	5	1	3	0

d^0

$k = 2$

0	1	5	3	7
1	0	3	1	5
5	3	0	1	1
3	1	1	0	3
7	5	1	3	0

d^1

0	1	5	3	7
1	0	3	1	5
5	3	0	1	1
3	1	1	0	3
7	5	1	3	0

d^1

0	1	**4**	3	7
1	0	3	1	5
5	3	0	1	1
3	1	1	0	3
4	5	1	3	0

d^2

Beispiel

$$k = 1 \qquad\qquad k = 2$$

$$\begin{matrix} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{matrix} \qquad\qquad \begin{matrix} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{matrix}$$

$$d^0 \qquad\qquad\qquad d^1$$

$$\begin{matrix} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{matrix} \qquad\qquad \begin{matrix} 0 & 1 & \mathbf{4} & \mathbf{2} & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 4 & 5 & 1 & 3 & 0 \end{matrix}$$

$$d^1 \qquad\qquad\qquad d^2$$

Beispiel

$$k = 1 \qquad\qquad k = 2$$

$$\begin{matrix} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{matrix} \qquad \begin{matrix} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{matrix}$$

$$d^0 \qquad\qquad d^1$$

$$\begin{matrix} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{matrix} \qquad \begin{matrix} 0 & 1 & \mathbf{4} & \mathbf{2} & \mathbf{6} \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 4 & 5 & 1 & 3 & 0 \end{matrix}$$

$$d^1 \qquad\qquad d^2$$

Beispiel

$k = 1$

0	1	5	3	7
1	0	3	1	5
5	3	0	1	1
3	1	1	0	3
7	5	1	3	0

d^0

$k = 2$

0	1	5	3	7
1	0	3	1	5
5	3	0	1	1
3	1	1	0	3
7	5	1	3	0

d^1

0	1	5	3	7
1	0	3	1	5
5	3	0	1	1
3	1	1	0	3
7	5	1	3	0

d^1

0	1	**4**	**2**	**6**
1	0	3	1	5
4	3	0	1	1
2	1	1	0	3
6	5	1	3	0

d^2

Beispiel

$k = 1$

$$d^0 = \begin{matrix} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{matrix}$$

$k = 2$

$$d^1 = \begin{matrix} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{matrix}$$

$k = 3$

$$d^2 = \begin{matrix} 0 & 1 & 4 & 2 & 6 \\ 1 & 0 & 3 & 1 & 5 \\ 4 & 3 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 3 \\ 6 & 5 & 1 & 3 & 0 \end{matrix}$$

$$d^1 = \begin{matrix} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{matrix}$$

$$d^2 = \begin{matrix} 0 & 1 & \mathbf{4} & \mathbf{2} & \mathbf{6} \\ 1 & 0 & 3 & 1 & 5 \\ \mathbf{4} & 3 & 0 & 1 & 1 \\ \mathbf{2} & 1 & 1 & 0 & 3 \\ \mathbf{6} & 5 & 1 & 3 & 0 \end{matrix}$$

Beispiel

$k = 1$

$$\begin{array}{ccccc} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{array}$$

d^0

$k = 2$

$$\begin{array}{ccccc} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{array}$$

d^1

$k = 3$

$$\begin{array}{ccccc} 0 & 1 & 4 & 2 & 6 \\ 1 & 0 & 3 & 1 & 5 \\ 4 & 3 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 3 \\ 6 & 5 & 1 & 3 & 0 \end{array}$$

d^2

$$\begin{array}{ccccc} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{array}$$

d^1

$$\begin{array}{ccccc} 0 & 1 & \mathbf{4} & \mathbf{2} & \mathbf{6} \\ 1 & 0 & 3 & 1 & 5 \\ \mathbf{4} & 3 & 0 & 1 & 1 \\ \mathbf{2} & 1 & 1 & 0 & 3 \\ \mathbf{6} & 5 & 1 & 3 & 0 \end{array}$$

d^2

$$\begin{array}{ccccc} 0 & 1 & 4 & 2 & \mathbf{5} \\ 1 & 0 & 3 & 1 & \mathbf{4} \\ 4 & 3 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & \mathbf{2} \\ \mathbf{5} & \mathbf{4} & 1 & \mathbf{2} & 0 \end{array}$$

d^3

Beispiel

$$k = 1 \qquad\qquad k = 2 \qquad\qquad k = 3 \qquad\qquad k = 4$$

$$\begin{matrix} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{matrix} \qquad \begin{matrix} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{matrix} \qquad \begin{matrix} 0 & 1 & 4 & 2 & 6 \\ 1 & 0 & 3 & 1 & 5 \\ 4 & 3 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 3 \\ 6 & 5 & 1 & 3 & 0 \end{matrix} \qquad \begin{matrix} 0 & 1 & 4 & 2 & 5 \\ 1 & 0 & 3 & 1 & 4 \\ 4 & 3 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 2 \\ 5 & 4 & 1 & 2 & 0 \end{matrix}$$

$$d^0 \qquad\qquad d^1 \qquad\qquad d^2 \qquad\qquad d^3$$

$$\begin{matrix} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{matrix} \qquad \begin{matrix} 0 & 1 & \mathbf{4} & \mathbf{2} & \mathbf{6} \\ 1 & 0 & 3 & 1 & 5 \\ \mathbf{4} & 3 & 0 & 1 & 1 \\ \mathbf{2} & 1 & 1 & 0 & 3 \\ \mathbf{6} & 5 & 1 & 3 & 0 \end{matrix} \qquad \begin{matrix} 0 & 1 & 4 & 2 & \mathbf{5} \\ 1 & 0 & 3 & 1 & \mathbf{4} \\ 4 & 3 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & \mathbf{2} \\ \mathbf{5} & \mathbf{4} & 1 & \mathbf{2} & 0 \end{matrix}$$

$$d^1 \qquad\qquad d^2 \qquad\qquad d^3$$

Beispiel

$k = 1$

$$\begin{matrix} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{matrix}$$

d^0

$k = 2$

$$\begin{matrix} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{matrix}$$

d^1

$k = 3$

$$\begin{matrix} 0 & 1 & 4 & 2 & 6 \\ 1 & 0 & 3 & 1 & 5 \\ 4 & 3 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 3 \\ 6 & 5 & 1 & 3 & 0 \end{matrix}$$

d^2

$k = 4$

$$\begin{matrix} 0 & 1 & 4 & 2 & 5 \\ 1 & 0 & 3 & 1 & 4 \\ 4 & 3 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 2 \\ 5 & 4 & 1 & 2 & 0 \end{matrix}$$

d^3

$$\begin{matrix} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{matrix}$$

d^1

$$\begin{matrix} 0 & 1 & \mathbf{4} & \mathbf{2} & \mathbf{6} \\ 1 & 0 & 3 & 1 & 5 \\ \mathbf{4} & 3 & 0 & 1 & 1 \\ \mathbf{2} & 1 & 1 & 0 & 3 \\ \mathbf{6} & 5 & 1 & 3 & 0 \end{matrix}$$

d^2

$$\begin{matrix} 0 & 1 & 4 & 2 & \mathbf{5} \\ 1 & 0 & 3 & 1 & \mathbf{4} \\ 4 & 3 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & \mathbf{2} \\ \mathbf{5} & \mathbf{4} & 1 & \mathbf{2} & 0 \end{matrix}$$

d^3

$$\begin{matrix} 0 & 1 & \mathbf{3} & 2 & \mathbf{4} \\ 1 & 0 & \mathbf{2} & 1 & \mathbf{3} \\ \mathbf{3} & \mathbf{2} & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 2 \\ \mathbf{4} & \mathbf{3} & 1 & 2 & 0 \end{matrix}$$

d^4

Beispiel

$k = 1$

$$\begin{matrix} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{matrix}$$

d^0

$k = 2$

$$\begin{matrix} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{matrix}$$

d^1

$k = 3$

$$\begin{matrix} 0 & 1 & 4 & 2 & 6 \\ 1 & 0 & 3 & 1 & 5 \\ 4 & 3 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 3 \\ 6 & 5 & 1 & 3 & 0 \end{matrix}$$

d^2

$k = 4$

$$\begin{matrix} 0 & 1 & 4 & 2 & 5 \\ 1 & 0 & 3 & 1 & 4 \\ 4 & 3 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 2 \\ 5 & 4 & 1 & 2 & 0 \end{matrix}$$

d^3

$k = 5$

$$\begin{matrix} 0 & 1 & 3 & 2 & 4 \\ 1 & 0 & 2 & 1 & 3 \\ 3 & 2 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 2 \\ 4 & 3 & 1 & 2 & 0 \end{matrix}$$

d^4

$$\begin{matrix} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{matrix}$$

d^1

$$\begin{matrix} 0 & 1 & \mathbf{4} & \mathbf{2} & \mathbf{6} \\ 1 & 0 & 3 & 1 & 5 \\ \mathbf{4} & 3 & 0 & 1 & 1 \\ \mathbf{2} & 1 & 1 & 0 & 3 \\ \mathbf{6} & 5 & 1 & 3 & 0 \end{matrix}$$

d^2

$$\begin{matrix} 0 & 1 & 4 & 2 & \mathbf{5} \\ 1 & 0 & 3 & 1 & \mathbf{4} \\ 4 & 3 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & \mathbf{2} \\ \mathbf{5} & \mathbf{4} & 1 & \mathbf{2} & 0 \end{matrix}$$

d^3

$$\begin{matrix} 0 & 1 & \mathbf{3} & 2 & \mathbf{4} \\ 1 & 0 & \mathbf{2} & 1 & \mathbf{3} \\ \mathbf{3} & \mathbf{2} & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 2 \\ \mathbf{4} & \mathbf{3} & 1 & 2 & 0 \end{matrix}$$

d^4

$$\begin{matrix} 0 & 1 & 3 & 2 & 4 \\ 1 & 0 & 2 & 1 & 3 \\ 3 & 2 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 2 \\ 4 & 3 & 1 & 2 & 0 \end{matrix}$$

d^5

Kürzester Weg für jedes Paar?

$$M = \begin{matrix} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{matrix} \qquad D := d^5 = \begin{matrix} 0 & 1 & 3 & 2 & 4 \\ 1 & 0 & 2 & 1 & 3 \\ 3 & 2 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 2 \\ 4 & 3 & 1 & 2 & 0 \end{matrix}$$

Frage: Können wir die berechnete Matrix D verwenden, um den kürzesten Weg für jedes Knotenpaar zu bestimmen?

Kürzester Weg für jedes Paar?

$$
\begin{array}{ccc}
M & D := d^5 & D' \\
\begin{array}{ccccc}
0 & 1 & 5 & 3 & 7 \\
1 & 0 & 3 & 1 & 5 \\
5 & 3 & 0 & 1 & 1 \\
3 & 1 & 1 & 0 & 3 \\
7 & 5 & 1 & 3 & 0
\end{array}
&
\begin{array}{ccccc}
0 & 1 & 3 & 2 & 4 \\
1 & 0 & 2 & 1 & 3 \\
3 & 2 & 0 & 1 & 1 \\
2 & 1 & 1 & 0 & 2 \\
4 & 3 & 1 & 2 & 0
\end{array}
&
\begin{array}{ccccc}
0 & 1 & 3 & 2 & 4 \\
1 & 0 & 2 & 1 & 3 \\
3 & 2 & 0 & 1 & 1 \\
2 & 1 & 1 & 0 & 2 \\
4 & 3 & 1 & 2 & 0
\end{array}
\end{array}
$$

Frage: Können wir die berechnete Matrix D verwenden, um den kürzesten Weg für jedes Knotenpaar zu bestimmen?

Direkte Wege klar: $i \to j$ wo $M[i,j] = D[i,j]$ (Siehe Markierungen in D' oben)

Kürzester Weg für jedes Paar?

$$
\begin{array}{c}
M \\
\begin{array}{ccccc}
0 & 1 & 5 & 3 & 7 \\
1 & 0 & 3 & 1 & 5 \\
5 & 3 & 0 & 1 & 1 \\
3 & 1 & 1 & 0 & 3 \\
7 & 5 & 1 & 3 & 0
\end{array}
\end{array}
\qquad
\begin{array}{c}
D := d^5 \\
\begin{array}{ccccc}
0 & 1 & 3 & 2 & 4 \\
1 & 0 & 2 & 1 & 3 \\
3 & 2 & 0 & 1 & 1 \\
2 & 1 & 1 & 0 & 2 \\
4 & 3 & 1 & 2 & 0
\end{array}
\end{array}
\qquad
\begin{array}{c}
D' \\
\begin{array}{ccccc}
0 & 1 & 3 & 2 & 4 \\
1 & 0 & 2 & 1 & 3 \\
3 & 2 & 0 & 1 & 1 \\
2 & 1 & 1 & 0 & 2 \\
4 & 3 & 1 & 2 & 0
\end{array}
\end{array}
\qquad
\begin{array}{c}
D'' \\
\begin{array}{ccccc}
0 & 1 & 3 & 2 & 4 \\
1 & 0 & 2 & 1 & 3 \\
3 & 2 & 0 & 1 & 1 \\
2 & 1 & 1 & 0 & 2 \\
4 & 3 & 1 & 2 & 0
\end{array}
\end{array}
$$

Frage: Können wir die berechnete Matrix D verwenden, um den kürzesten Weg für jedes Knotenpaar zu bestimmen?

Direkte Wege klar: $i \to j$ wo $M[i,j] = D[i,j]$ (Siehe Markierungen in D' oben)

Könnten versuchen, den Algorithmus rückwärts laufen zu lassen. Beispiel $1 \to 3$ oben in D'': Kandidaten sind markiert. Finde mit absteigendem k den ersten passenden Kandidaten.

Kürzester Weg für jedes Paar?

$$M = \begin{array}{ccccc} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{array} \qquad D := d^5 = \begin{array}{ccccc} 0 & 1 & 3 & 2 & 4 \\ 1 & 0 & 2 & 1 & 3 \\ 3 & 2 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 2 \\ 4 & 3 & 1 & 2 & 0 \end{array}$$

$$D' = \begin{array}{ccccc} 0 & 1 & 3 & 2 & 4 \\ 1 & 0 & 2 & 1 & 3 \\ 3 & 2 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 2 \\ 4 & 3 & 1 & 2 & 0 \end{array} \qquad D'' = \begin{array}{ccccc} 0 & 1 & 3 & 2 & 4 \\ 1 & 0 & 2 & 1 & 3 \\ 3 & 2 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 2 \\ 4 & 3 & 1 & 2 & 0 \end{array}$$

Frage: Können wir die berechnete Matrix D verwenden, um den kürzesten Weg für jedes Knotenpaar zu bestimmen?

Direkte Wege klar: $i \to j$ wo $M[i,j] = D[i,j]$ (Siehe Markierungen in D' oben)

Könnten versuchen, den Algorithmus rückwärts laufen zu lassen. Beispiel $1 \to 3$ oben in D'': Kandidaten sind markiert. Finde mit absteigendem k den ersten passenden Kandidaten.

Kompliziert. Und uneffizient.

Idee

Idee

Merken uns jeweils das beste k für jedes (i, j) im Algorithmus in einer Matrix K.

Idee

Merken uns jeweils das beste k für jedes (i, j) im Algorithmus in einer Matrix K.

Starten mit Matrix der existierenden Direktverbindungen (Kanten)

Beispiel

$$k = 1$$

$$B = \begin{array}{ccccc} 0 & 1 & 5 & 3 & 7 \\ 1 & 0 & 3 & 1 & 5 \\ 5 & 3 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 3 \\ 7 & 5 & 1 & 3 & 0 \end{array} \qquad K = \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{array}$$

$$k = 2$$

$$\begin{array}{ccccc} 0 & 1 & \mathbf{4} & \mathbf{2} & \mathbf{6} \\ 1 & 0 & 3 & 1 & 5 \\ \mathbf{4} & 3 & 0 & 1 & 1 \\ \mathbf{2} & 1 & 1 & 0 & 3 \\ \mathbf{6} & 5 & 1 & 3 & 0 \end{array} \qquad \begin{array}{ccccc} 1 & 2 & \mathbf{2} & \mathbf{2} & \mathbf{2} \\ 1 & 2 & 3 & 4 & 5 \\ \mathbf{2} & 2 & 3 & 4 & 5 \\ \mathbf{2} & 2 & 3 & 4 & 5 \\ \mathbf{2} & 2 & 3 & 4 & 5 \end{array}$$

$$k = 3$$

$$\begin{array}{ccccc} 0 & 1 & 4 & 2 & \mathbf{5} \\ 1 & 0 & 3 & 1 & \mathbf{4} \\ 4 & 3 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & \mathbf{2} \\ \mathbf{5} & \mathbf{4} & 1 & \mathbf{2} & 0 \end{array} \qquad \begin{array}{ccccc} 1 & 2 & 2 & 2 & \mathbf{3} \\ 1 & 2 & 3 & 4 & \mathbf{3} \\ 2 & 2 & 3 & 4 & 5 \\ 2 & 2 & 3 & 4 & \mathbf{3} \\ \mathbf{3} & \mathbf{3} & 3 & \mathbf{3} & 5 \end{array}$$

$$k = 4$$

$$\begin{array}{ccccc} 0 & 1 & \mathbf{3} & 2 & \mathbf{4} \\ 1 & 0 & \mathbf{2} & 1 & \mathbf{3} \\ \mathbf{3} & \mathbf{2} & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 2 \\ \mathbf{4} & \mathbf{3} & 1 & 2 & 0 \end{array} \qquad \begin{array}{ccccc} 1 & 2 & \mathbf{4} & 2 & \mathbf{4} \\ 1 & 2 & \mathbf{4} & 4 & \mathbf{4} \\ \mathbf{4} & \mathbf{4} & 3 & 4 & 5 \\ 2 & 2 & 3 & 4 & 3 \\ \mathbf{4} & \mathbf{4} & 3 & 3 & 5 \end{array}$$

$$k = 5$$

$$\begin{array}{ccccc} 0 & 1 & 3 & 2 & 4 \\ 1 & 0 & 2 & 1 & 3 \\ 3 & 2 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 2 \\ 4 & 3 & 1 & 2 & 0 \end{array} \qquad \begin{array}{ccccc} 1 & 2 & 4 & 2 & 4 \\ 1 & 2 & 4 & 4 & 4 \\ 4 & 4 & 3 & 4 & 5 \\ 2 & 2 & 3 & 4 & 3 \\ 4 & 4 & 3 & 3 & 5 \end{array}$$

Beispiel

Wie liest man diese Matrix K?

$$K$$

	1	2	3	4	5
1	1	2	4	2	4
2	1	2	4	4	4
3	4	4	3	4	5
4	2	2	3	4	3
5	4	4	3	3	5

Beispiel

K

	1	2	3	4	5
1	1	2	4	2	4
2	1	2	4	4	4
3	4	4	3	4	5
4	2	2	3	4	3
5	4	4	3	3	5

Wie liest man diese Matrix K?

Am Beispiel $1 \to 5$:

$1 \to 5$

$1 \to 4 \to 5$

$1 \to 2 \to 4 \to 3 \to 5$

$\boxed{1 \to 2 \to 4 \to 3 \to 5}$

Beispiel

K

	1	2	3	4	5
1	1	2	4	2	4
2	1	2	4	4	4
3	4	4	3	4	5
4	2	2	3	4	3
5	4	4	3	3	5

Wie liest man diese Matrix K?
Am Beispiel $1 \to 5$:

- Weg $1 \to 5$ geht über Knoten 4.

Beispiel

K

	1	2	3	4	5
1	1	2	4	2	4
2	1	2	4	4	4
3	4	4	3	4	5
4	2	2	3	4	3
5	4	4	3	3	5

Wie liest man diese Matrix K?
Am Beispiel $1 \to 5$:

- Weg $1 \to 5$ geht über Knoten 4.
- Weg $1 \to 4$ geht über Knoten 2.

Beispiel

K

	1	2	3	4	5
1	1	2	4	2	4
2	1	2	4	4	4
3	4	4	3	4	5
4	2	2	3	4	3
5	4	4	3	3	5

Wie liest man diese Matrix K?
Am Beispiel $1 \to 5$:

- Weg $1 \to 5$ geht über Knoten 4.
- Weg $1 \to 4$ geht über Knoten 2.
- Weg $4 \to 5$ geht über Knoten 3.

Beispiel

K

	1	2	3	4	5
1	1	2	4	2	4
2	1	2	4	4	4
3	4	4	3	4	5
4	2	2	3	4	3
5	4	4	3	3	5

Wie liest man diese Matrix K?
Am Beispiel $1 \to 5$:

- Weg $1 \to 5$ geht über Knoten 4.
- Weg $1 \to 4$ geht über Knoten 2.
- Weg $4 \to 5$ geht über Knoten 3.
- Pfade $1 \to 2$ und $2 \to 4$ Sind direkt.

Beispiel

K

	1	2	3	4	5
1	1	2	4	2	4
2	1	2	4	4	4
3	4	4	3	4	5
4	2	2	3	4	3
5	4	4	3	3	5

Wie liest man diese Matrix K?
Am Beispiel $1 \to 5$:

- Weg $1 \to 5$ geht über Knoten 4.
- Weg $1 \to 4$ geht über Knoten 2.
- Weg $4 \to 5$ geht über Knoten 3.
- Pfade $1 \to 2$ und $2 \to 4$ Sind direkt.
- Pfade $4 \to 3$ und $3 \to 5$ Sind direkt.

Beispiel

K	1	2	3	4	5
1	1	2	4	2	4
2	1	2	4	4	4
3	4	4	3	4	5
4	2	2	3	4	3
5	4	4	3	3	5

Wie liest man diese Matrix K?
Am Beispiel $1 \to 5$:

- Weg $1 \to 5$ geht über Knoten 4.
- Weg $1 \to 4$ geht über Knoten 2.
- Weg $4 \to 5$ geht über Knoten 3.
- Pfade $1 \to 2$ und $2 \to 4$ Sind direkt.
- Pfade $4 \to 3$ und $3 \to 5$ Sind direkt.

Beispiel

$$K$$

	1	2	3	4	5
1	1	2	4	2	4
2	1	2	4	4	4
3	4	4	3	4	5
4	2	2	3	4	3
5	4	4	3	3	5

Wie liest man diese Matrix K?
Am Beispiel $1 \to 5$:

- Weg $1 \to 5$ geht über Knoten 4.
- Weg $1 \to 4$ geht über Knoten 2.
- Weg $4 \to 5$ geht über Knoten 3.
- Pfade $1 \to 2$ und $2 \to 4$ Sind direkt.
- Pfade $4 \to 3$ und $3 \to 5$ Sind direkt.

Insgesamt

$$1 \xrightarrow{4} 5 \quad \Rightarrow \quad 1 \xrightarrow{2} 4 \xrightarrow{3} 5 \quad \Rightarrow \quad 1 \to 2 \to 4 \to 3 \to 5$$

Beispiel

$$K$$

	1	2	3	4	5
1	1	2	4	2	4
2	1	2	4	4	4
3	4	4	3	4	5
4	2	2	3	4	3
5	4	4	3	3	5

Wie liest man diese Matrix K?
Am Beispiel $1 \to 5$:

- Weg $1 \to 5$ geht über Knoten 4.
- Weg $1 \to 4$ geht über Knoten 2.
- Weg $4 \to 5$ geht über Knoten 3.
- Pfade $1 \to 2$ und $2 \to 4$ Sind direkt.
- Pfade $4 \to 3$ und $3 \to 5$ Sind direkt.

Insgesamt

$$1 \xrightarrow{4} 5 \quad \Rightarrow \quad 1 \xrightarrow{2} 4 \xrightarrow{3} 5 \quad \Rightarrow \quad 1 \to 2 \to 4 \to 3 \to 5$$

Rekonstruktion via Rekursion.

Beispiel

$$K$$

	1	2	3	4	5
1	1	2	4	2	4
2	1	2	4	4	4
3	4	4	3	4	5
4	2	2	3	4	3
5	4	4	3	3	5

Wie liest man diese Matrix K?
Am Beispiel $1 \to 5$:

- Weg $1 \to 5$ geht über Knoten 4.
- Weg $1 \to 4$ geht über Knoten 2.
- Weg $4 \to 5$ geht über Knoten 3.
- Pfade $1 \to 2$ und $2 \to 4$ Sind direkt.
- Pfade $4 \to 3$ und $3 \to 5$ Sind direkt.

Insgesamt

$$1 \xrightarrow{4} 5 \quad \Rightarrow \quad 1 \xrightarrow{2} 4 \xrightarrow{3} 5 \quad \Rightarrow \quad 1 \to 2 \to 4 \to 3 \to 5$$

Rekonstruktion via Rekursion.
Alternative?

Beispiel

$$K$$

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 1 & 2 & 4 & 2 & 4 \\ 2 & 1 & 2 & 4 & 4 & 4 \\ 3 & 4 & 4 & 3 & 4 & 5 \\ 4 & 2 & 2 & 3 & 4 & 3 \\ 5 & 4 & 4 & 3 & 3 & 5 \end{array}$$

Wie liest man diese Matrix K?
Am Beispiel $1 \to 5$:

- Weg $1 \to 5$ geht über Knoten 4.
- Weg $1 \to 4$ geht über Knoten 2.
- Weg $4 \to 5$ geht über Knoten 3.
- Pfade $1 \to 2$ und $2 \to 4$ Sind direkt.
- Pfade $4 \to 3$ und $3 \to 5$ Sind direkt.

Insgesamt

$$1 \xrightarrow{4} 5 \quad \Rightarrow \quad 1 \xrightarrow{2} 4 \xrightarrow{3} 5 \quad \Rightarrow \quad 1 \to 2 \to 4 \to 3 \to 5$$

Rekonstruktion via Rekursion.
Alternative? Im Algorithmus Nachfolger merken

Beispiel

	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
B	0 1 5 3 7	0 1 **4** **2** **6**	0 1 4 2 **5**	0 1 **3** 2 **4**	0 1 3 2 4
	1 0 3 1 5	1 0 3 1 5	1 0 3 1 **4**	1 0 **2** 1 **3**	1 0 2 1 3
	5 3 0 1 1	**4** 3 0 1 1	4 3 0 1 1	**3** **2** 0 1 1	3 2 0 1 1
	3 1 1 0 3	**2** 1 1 0 3	2 1 1 0 **2**	2 1 1 0 2	2 1 1 0 2
	7 5 1 3 0	**6** 5 1 3 0	**5** **4** 1 **2** 0	**4** **3** 1 2 0	4 3 1 2 0
K	1 2 3 4 5	1 2 **2** **2** **2**	1 2 2 2 **2**	1 2 **2** 2 **2**	1 2 2 2 2
	1 2 3 4 5	1 2 3 4 5	1 2 3 4 **3**	1 2 **4** 4 **4**	1 2 4 4 4
	1 2 3 4 5	**2** 2 3 4 5	2 2 3 4 5	**4** **4** 3 4 5	4 4 3 4 5
	1 2 3 4 5	**2** 2 3 4 5	2 2 3 4 **3**	2 2 3 4 3	2 2 3 4 3
	1 2 3 4 5	**2** 2 3 4 5	**3** **3** 3 **3** 5	**3** **3** 3 3 5	3 3 3 3 5

Vergleich der Verfahren

Single Source Short. Path

All-pairs sh. path.

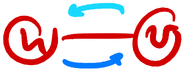
Algorithmus	Bedingung		Laufzeit
Dijkstra (Heap)	$c_v \geq 0$	1:n	$\mathcal{O}(\lvert E\rvert \log \lvert V\rvert)$
Dijkstra (Fibonacci-Heap)	$c_v \geq 0$	1:n	$\mathcal{O}(\lvert E\rvert + \lvert V\rvert \log \lvert V\rvert)$ *
Bellman-Ford		1:n	$\mathcal{O}(\lvert E\rvert \cdot \lvert V\rvert)$
Floyd-Warshall		n:n	$\Theta(\lvert V\rvert^3)$
Johnson		n:n	$\mathcal{O}(\lvert V\rvert \cdot \lvert E\rvert \cdot \log \lvert V\rvert)$
Johnson (Fibonacci-Heap)		n:n	$\mathcal{O}(\lvert V\rvert^2 \log \lvert V\rvert + \lvert V\rvert \cdot \lvert E\rvert)$ *

* amortisiert

Johnson (dieses Jahr nicht erklärt) ist besser als Floyd-Warshall nur für dünn besetzte Graphen $(\lvert E\rvert \approx \Theta(\lvert V\rvert))$.

$E = \{ \{u,v\} \ldots \}$

$\{ (u,v), (v,u) \ldots \}$



Fragen/Unklarheiten?

6. Live [code]expert

Live [code]expert

'Lazy Deletion' auf [code]expert

7. In-Class-Exercise (theoretisch)

In-Class-Exercises: Längster Pfad in DAGs

Das Kürzeste-Pfad-Problem hat einfache Lösungen (BFS, Dijkstra, Bellman-Ford). Das Längste-Pfad-Problem hingegen ist sehr schwierig! Für gerichtete Graphen gibt es vermutlich keinen schnellen Algorithmus, um Pfade der Länge $\gg \log^2 n$ zu finden.

In-Class-Exercises: Längster Pfad in DAGs

Das Kürzeste-Pfad-Problem hat einfache Lösungen (BFS, Dijkstra, Bellman-Ford). Das Längste-Pfad-Problem hingegen ist sehr schwierig! Für gerichtete Graphen gibt es vermutlich keinen schnellen Algorithmus, um Pfade der Länge $\gg \log^2 n$ zu finden.

Aufgabe:

Gegeben sei ein gerichteter, **kreisfreier** Graph (DAG) $G = (V, E)$.

Entwerfen Sie einen $\mathcal{O}(|V| + |E|)$-Laufzeit Algorithmus, um den *längsten Pfad* zu finden.

In-Class-Exercises: Längster Pfad in DAGs

Das Kürzeste-Pfad-Problem hat einfache Lösungen (BFS, Dijkstra, Bellman-Ford). Das Längste-Pfad-Problem hingegen ist sehr schwierig! Für gerichtete Graphen gibt es vermutlich keinen schnellen Algorithmus, um Pfade der Länge $\gg \log^2 n$ zu finden.

Aufgabe:

Gegeben sei ein gerichteter, **kreisfreier** Graph (DAG) $G = (V, E)$.

Entwerfen Sie einen $\mathcal{O}(|V| + |E|)$-Laufzeit Algorithmus, um den *längsten Pfad* zu finden.

Tipp: G ist kreisfrei, Sie können also zuerst topologisch sortieren.

In-Class-Exercises: Längster Pfad in DAGs

Lösung:

1. Topologisch Sortieren. Laufzeit: $\mathcal{O}(|V| + |E|)$.

In-Class-Exercises: Längster Pfad in DAGs

Lösung:

1. Topologisch Sortieren. Laufzeit: $\mathcal{O}(|V| + |E|)$.
2. Berechne für jeden Knoten alle eingehenden Kanten: $\mathcal{O}(|V| + |E|)$.

In-Class-Exercises: Längster Pfad in DAGs

Lösung:

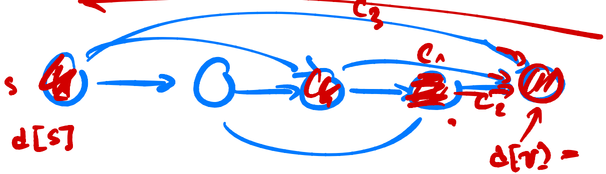
1. Topologisch Sortieren. Laufzeit: $\mathcal{O}(|V| + |E|)$.
2. Berechne für jeden Knoten alle eingehenden Kanten: $\mathcal{O}(|V| + |E|)$.
3. Besuche jeden Knoten v in Reihenfolge der topologischen Sortierung und betrachte die Eingangs-Kanten: $\mathcal{O}(|V| + |E|)$.

In-Class-Exercises: Längster Pfad in DAGs

Lösung:

1. Topologisch Sortieren. Laufzeit: $\mathcal{O}(|V| + |E|)$.
2. Berechne für jeden Knoten alle eingehenden Kanten: $\mathcal{O}(|V| + |E|)$.
3. Besuche jeden Knoten v in Reihenfolge der topologischen Sortierung und betrachte die Eingangs-Kanten: $\mathcal{O}(|V| + |E|)$.

$$\mathtt{dist}[v] = \begin{cases} 0 & \text{keine Kanten,} \\ \max\limits_{(u,v) \in E} \{\mathtt{dist}[u] + c(u,v)\} & \text{sonst.} \end{cases}$$



In-Class-Exercises: Längster Pfad in DAGs

Lösung:

1. Topologisch Sortieren. Laufzeit: $\mathcal{O}(|V| + |E|)$.
2. Berechne für jeden Knoten alle eingehenden Kanten: $\mathcal{O}(|V| + |E|)$.
3. Besuche jeden Knoten v in Reihenfolge der topologischen Sortierung und betrachte die Eingangs-Kanten: $\mathcal{O}(|V| + |E|)$.

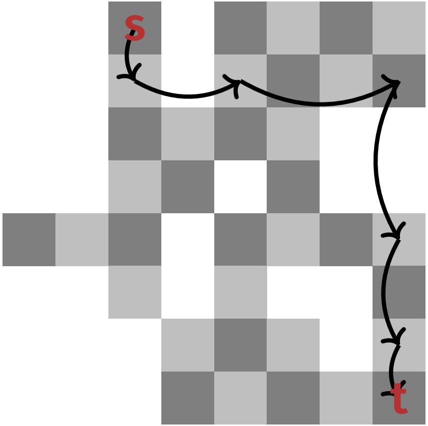
   $$\texttt{dist}[v] = \begin{cases} 0 & \text{keine Kanten,} \\ \max\limits_{(u,v) \in E} \{\texttt{dist}[u] + c(u, v)\} & \text{sonst.} \end{cases}$$

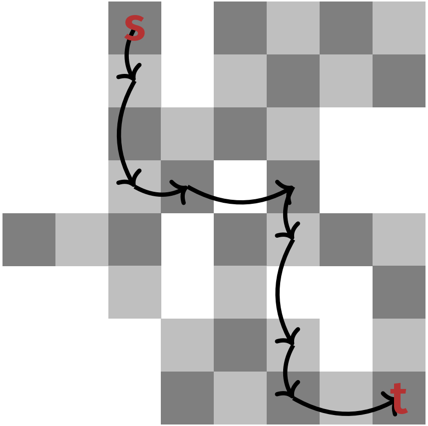
   Vorgänger merken!

Fragen/Unklarheiten?

8. Alte Prüfungsfragen (26.1.2018)

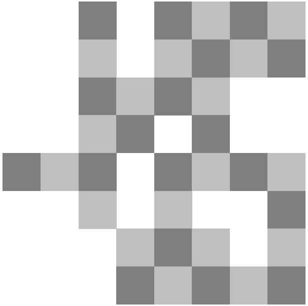
Kürzeste Wege Frage

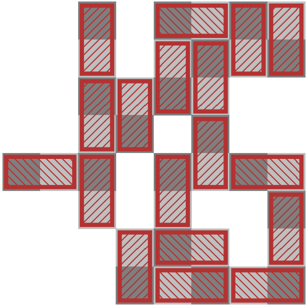




ist der dazugehörige Zustandsraum?

Max Flow Question





Wie bildet man das auf ein Max-Flow (Matching) Problem ab?

9. Outro

Allgemeine Fragen?

bei Fragen zu [code]expert $\longrightarrow$ Mail ($+$ Geduld)

Bis zum nächsten Mal

Schönes Wochenende!