Technische Universität München
Fakultät für Informatik, I-16
Dr. Stefanie Demirci
Mai Bui

SoSe 2018
Lösungsblatt 8
18. Juni 2018

Algorithmen und Datenstrukturen

Aufgabe 1 Dijkstra (Beispiellösung)

a) Um den Dijkstra-Algorithmus auf einen gewichteten Graphen $G = (V,E), w : E \to \mathbb{R}$ anwenden zu können, müssen alle Kantengewichte positiv sein. D.h.

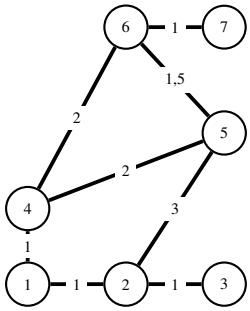
$$\forall e \in E : w(e) \geq 0$$

Diese Bedingung ist bei dem gegebenen Graphen erfüllt.

b)

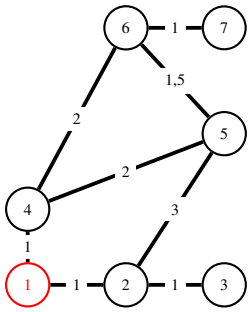
- Initialisiere Startknoten 1.

	1	2	3	4	5	6	7
pred							
dist	0	∞	∞	∞	∞	∞	∞



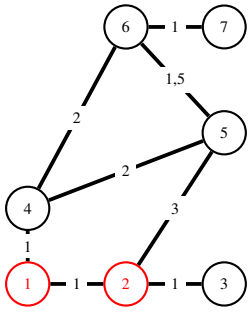
- Besuche Knoten 1. Die Auswahl erfolgt stets wegen minimaler Distanz.

	1	2	3	4	5	6	7
pred		1		1			
dist	0	1	∞	1	∞	∞	∞



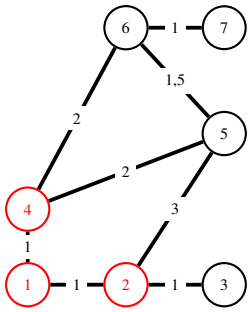
- Besuche Knoten 2.

	1	2	3	4	5	6	7
pred		1	2	1	2		
dist	0	1	2	1	4	∞	∞



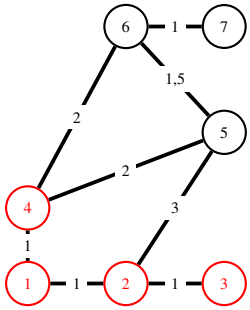
- Besuche Knoten 4. Dabei erfolgt Update für Knoten 5!

	1	2	3	4	5	6	7
pred		1	2	1	4	4	
dist	0	1	2	1	3	3	∞



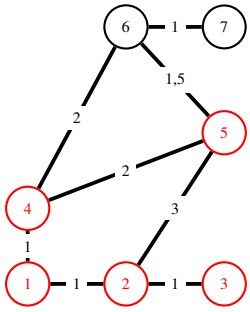
- Besuche Knoten 3. Dieser hat keine unbesuchten Nachbarn, also keine Änderungen!

	1	2	3	4	5	6	7
pred		1	2	1	4	4	
dist	0	1	2	1	3	3	∞



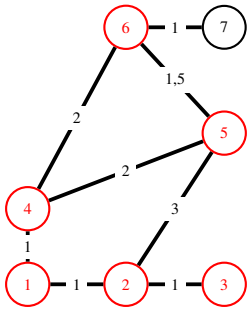
- Besuche Knoten 5. Der Pfad zu Knoten 6 über 5 ist schlechter als der bereits bekannte, also keine Änderungen!

	1	2	3	4	5	6	7
pred		1	2	1	4	4	
dist	0	1	2	1	3	3	∞



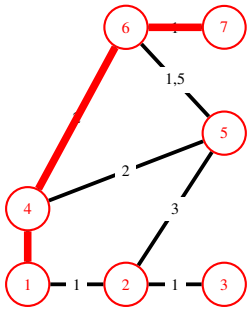
- Besuche Knoten 6.

	1	2	3	4	5	6	7
pred		1	2	1	4	4	6
dist	0	1	2	1	3	3	4



- Besuche Knoten 7. Wir sind am Ziel, der kürzeste Pfad ist also $1 \to 4 \to 6 \to 7$ mit Kosten 4. Dieser Pfad kann aus der Tabelle ausgelesen werden, indem man von Knoten 7 aus die jeweiligen Vorgänger *pred* verfolgt.

	1	2	3	4	5	6	7
pred		1	2	1	4	4	6
dist	0	1	2	1	3	3	4



c) Der Dijkstra-Algorithmus löst das *SSSP*-Problem.

d) Man kann den Algorithmus abbrechen, sobald der Endknoten abgearbeitet wurde. Ein solcher Abbruch problemlos möglich ist, da Dijkstra **nicht** „greedy“ ist, sondern stets in monotoner Steigerung immer länger werdende Pfade verfolgt werden. Wird also der Zielknoten besucht[1], gibt es keine kürzeren verfolgbaren Pfade mehr, und man kann abbrechen!

   Insofern beginnt man bei der Lösung des *SPSP*-Problems mit einer *SSSP*-Problem-Lösung und bricht allenfalls früher ab.

Aufgabe 2 Dijkstra und A* (Beispiellösung)

Die Heuristik wird ausschließlich in der Prioritätsliste verwendet, man würde also die Operation *decreaseKey* statt mit den akkumulierten Kosten *d[v]* mit $d[v] + h[v]$ durchführen. Insbesondere wird das Array der Pfadkosten *d* nicht verändert!

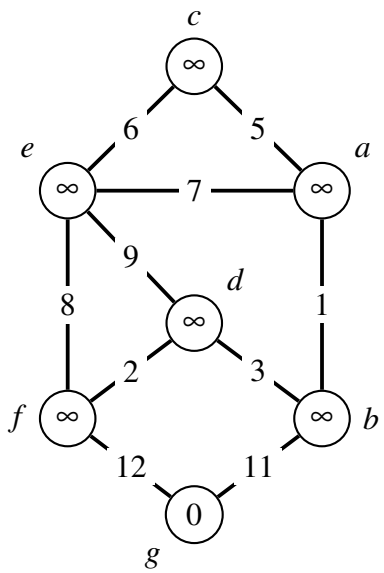
[1] Es ist wichtig, dass der Knoten wirklich besucht wird. Es genügt **nicht**, nur den ersten Vorgänger zu finden!

Aufgabe 3 Prim-Algorithmus (Beispiellösung)

Zum Verständnis werden die Kanten zu $pred[\cdot]$ grün dargestellt, während Kanten, welche dem MST hinzugefügt werde, rot gekennzeichnet sind. Ebenso werden abgearbeitete Knoten in rot dargestellt.

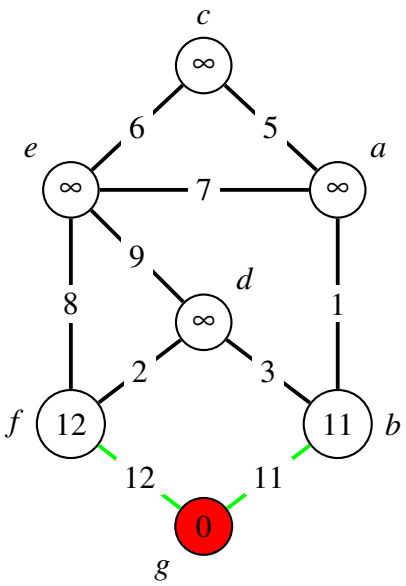
- Initialisiere Startknoten g.

$$Q = \{(g,0),(a,\infty),(b,\infty),(c,\infty),(d,\infty),(e,\infty),(f,\infty)\}$$



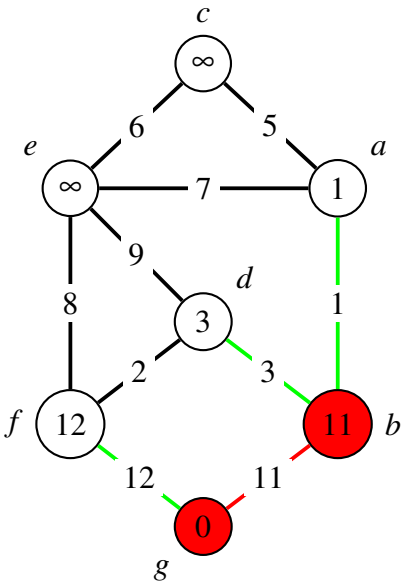
- Entferne das erstes Element aus der Priority Queue g und führe decreaseKey auf den Nachbarknoten aus.

$$Q = \{(b,11),(f,12),(a,\infty),(c,\infty),(d,\infty),(e,\infty)\}$$



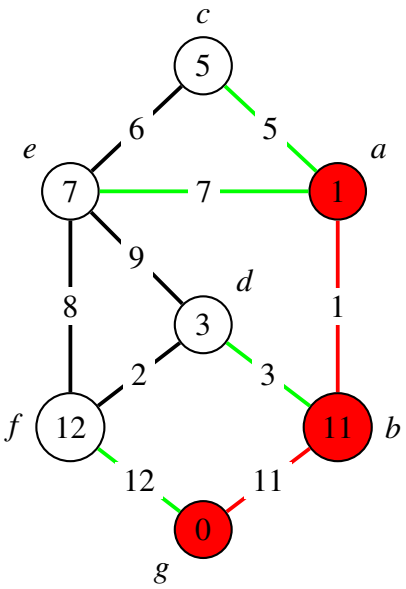
- Entferne das erstes Element aus der Priority Queue b, markiere die entsprechend Kante und führe decreaseKey auf den Nachbarknoten aus.

$Q = \{(a,1),(d,3),(f,12),(c,\infty),(e,\infty)\}$



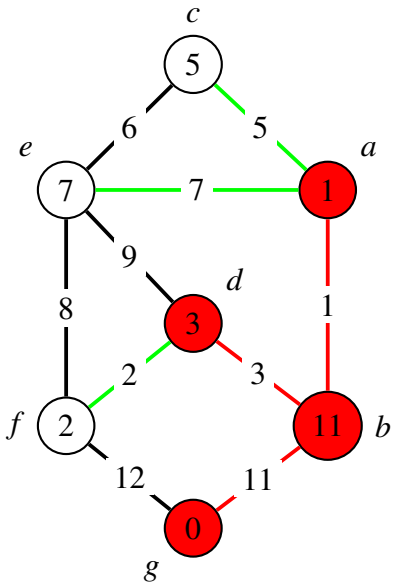
- Entferne das erstes Element aus der Priority Queue a, markiere die entsprechend Kante und führe decreaseKey auf den Nachbarknoten aus.

$Q = \{(d,3),(c,5),(e,7),(f,12)\}$



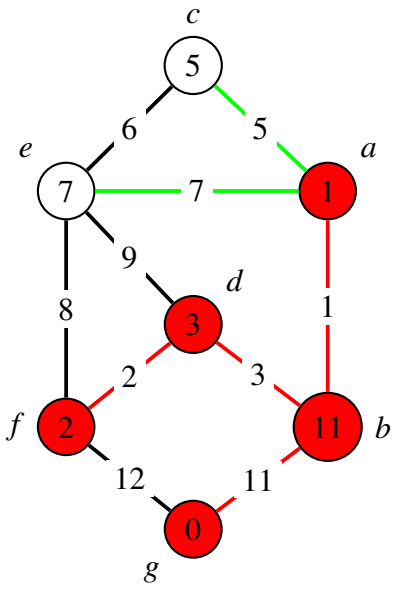
- Entferne das erstes Element aus der Priority Queue d, markiere die entsprechend Kante und führe decreaseKey auf den Nachbarknoten aus.

$Q = \{(f,2),(c,5),(e,7)\}$



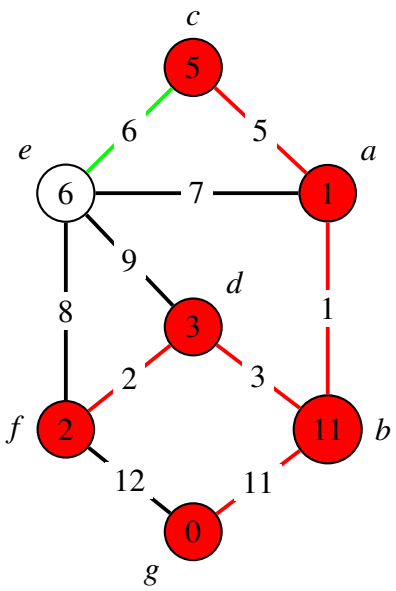
- Entferne das erstes Element aus der Priority Queue f, markiere die entsprechend Kante und führe decreaseKey auf den Nachbarknoten aus.

$Q = \{(c,5),(e,7)\}$



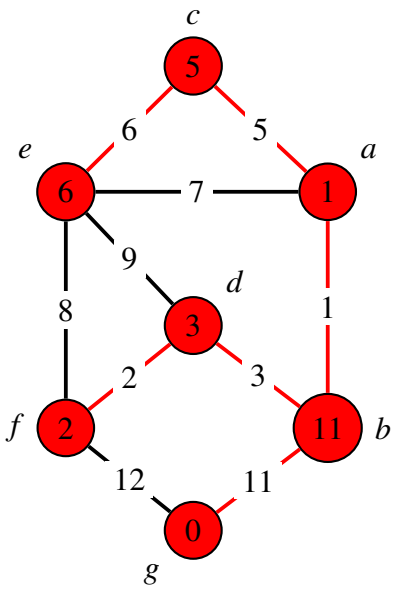
- Entferne das erstes Element aus der Priority Queue c, markiere die entsprechend Kante und führe decreaseKey auf den Nachbarknoten aus.

$Q = \{(e,6)\}$



- Entferne das erstes Element aus der Priority Queue e, markiere die entsprechend Kante und führe decreaseKey auf den Nachbarknoten aus.

$Q = \{\}$



Aufgabe 4 Strassen-Algorithmus (Beispiellösung)

Entsprechend der Vorlesung, beginnen wir mit dem Divide-Schritt (hier mittels entsprechender Klammerung dargestellt):

$$\underbrace{\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}}_{\mathbf{A}} \cdot \underbrace{\begin{pmatrix} 16 & 15 & 14 & 13 \\ 12 & 11 & 10 & 9 \\ 8 & 7 & 6 & 5 \\ 4 & 3 & 2 & 1 \end{pmatrix}}_{\mathbf{B}}$$

$$= \begin{pmatrix} \underbrace{\begin{pmatrix} 1 & 2 \\ 5 & 6 \end{pmatrix}}_{\mathbf{a_{11}}} & \underbrace{\begin{pmatrix} 3 & 4 \\ 7 & 8 \end{pmatrix}}_{\mathbf{a_{12}}} \\ \underbrace{\begin{pmatrix} 9 & 10 \\ 13 & 14 \end{pmatrix}}_{\mathbf{a_{21}}} & \underbrace{\begin{pmatrix} 11 & 12 \\ 15 & 16 \end{pmatrix}}_{\mathbf{a_{22}}} \end{pmatrix} \cdot \begin{pmatrix} \underbrace{\begin{pmatrix} 16 & 15 \\ 12 & 11 \end{pmatrix}}_{\mathbf{b_{11}}} & \underbrace{\begin{pmatrix} 14 & 13 \\ 10 & 9 \end{pmatrix}}_{\mathbf{b_{12}}} \\ \underbrace{\begin{pmatrix} 8 & 7 \\ 4 & 3 \end{pmatrix}}_{\mathbf{b_{21}}} & \underbrace{\begin{pmatrix} 6 & 5 \\ 2 & 1 \end{pmatrix}}_{\mathbf{b_{22}}} \end{pmatrix}$$

Nun der Conquer-Schritt:

$$\mathbf{q_1} = \underbrace{(\mathbf{a_{11}} + \mathbf{a_{22}})}_{\text{4-Additionen}} \cdot \underbrace{(\mathbf{b_{11}} + \mathbf{b_{22}})}_{\text{4-Additionen}} = \underbrace{\begin{pmatrix} 12 & 14 \\ 20 & 22 \end{pmatrix} \cdot \begin{pmatrix} 22 & 20 \\ 14 & 12 \end{pmatrix}}_{\text{8-Multiplikation, 4-Additionen}} = \begin{pmatrix} 460 & 408 \\ 748 & 664 \end{pmatrix}$$

$$\mathbf{q_2} = \underbrace{(\mathbf{a_{21}} + \mathbf{a_{22}})}_{\text{4-Additionen}} \cdot \mathbf{b_{11}} = \underbrace{\begin{pmatrix} 20 & 22 \\ 28 & 30 \end{pmatrix} \cdot \begin{pmatrix} 16 & 15 \\ 12 & 11 \end{pmatrix}}_{\text{8-Multiplikation, 4-Additionen}} = \begin{pmatrix} 584 & 542 \\ 808 & 750 \end{pmatrix}$$

$$\mathbf{q_3} = \mathbf{a_{11}} \cdot \underbrace{(\mathbf{b_{12}} - \mathbf{b_{22}})}_{\text{4-Subtraktionen}} = \underbrace{\begin{pmatrix} 1 & 2 \\ 5 & 6 \end{pmatrix} \cdot \begin{pmatrix} 8 & 8 \\ 8 & 8 \end{pmatrix}}_{\text{8-Multiplikation, 4-Additionen}} = \begin{pmatrix} 24 & 24 \\ 88 & 88 \end{pmatrix}$$

$$\mathbf{q_4} = \mathbf{a_{22}} \cdot \underbrace{(\mathbf{b_{21}} - \mathbf{b_{11}})}_{\text{4-Subtraktionen}} = \underbrace{\begin{pmatrix} 11 & 12 \\ 15 & 16 \end{pmatrix} \cdot \begin{pmatrix} -8 & -8 \\ -8 & -8 \end{pmatrix}}_{\text{8-Multiplikation, 4-Additionen}} = \begin{pmatrix} -184 & -184 \\ -248 & -248 \end{pmatrix}$$

$$\mathbf{q_5} = \underbrace{(\mathbf{a_{11}} + \mathbf{a_{12}})}_{\text{4-Additionen}} \cdot \mathbf{b_{22}} = \underbrace{\begin{pmatrix} 4 & 6 \\ 12 & 14 \end{pmatrix} \cdot \begin{pmatrix} 6 & 5 \\ 2 & 1 \end{pmatrix}}_{\text{8-Multiplikation, 4-Additionen}} = \begin{pmatrix} 36 & 26 \\ 100 & 74 \end{pmatrix}$$

$$\mathbf{q_6} = \underbrace{(\mathbf{a_{21}} - \mathbf{a_{11}})}_{\text{4-Subtraktionen}} \cdot \underbrace{(\mathbf{b_{11}} + \mathbf{b_{12}})}_{\text{4-Additionen}} = \underbrace{\begin{pmatrix} 8 & 8 \\ 8 & 8 \end{pmatrix} \cdot \begin{pmatrix} 30 & 28 \\ 22 & 20 \end{pmatrix}}_{\text{8-Multiplikation, 4-Additionen}} = \begin{pmatrix} 416 & 384 \\ 416 & 384 \end{pmatrix}$$

$$\mathbf{q_7} = \underbrace{(\mathbf{a_{12}} - \mathbf{a_{22}})}_{\text{4-Subtraktionen}} \cdot \underbrace{(\mathbf{b_{21}} + \mathbf{b_{22}})}_{\text{4-Additionen}} = \underbrace{\begin{pmatrix} -8 & -8 \\ -8 & -8 \end{pmatrix} \cdot \begin{pmatrix} 14 & 12 \\ 6 & 4 \end{pmatrix}}_{\text{8-Multiplikation, 4-Additionen}} = \begin{pmatrix} -160 & -128 \\ -160 & -128 \end{pmatrix}$$

$$\mathbf{A} \cdot \mathbf{B} = \underbrace{\begin{pmatrix} \mathbf{q_1} + \mathbf{q_4} - \mathbf{q_5} + \mathbf{q_7} & \mathbf{q_3} + \mathbf{q_5} \\ \mathbf{q_2} + \mathbf{q_4} & \mathbf{q_1} + \mathbf{q_3} - \mathbf{q_2} + \mathbf{q_6} \end{pmatrix}}_{\text{24-Additionen, 8-Subtraktionen}} = \begin{pmatrix} 80 & 70 & 60 & 50 \\ 240 & 214 & 188 & 162 \\ 400 & 358 & 316 & 274 \\ 560 & 502 & 444 & 386 \end{pmatrix}$$

Im Falle der „normalen“ Implementierung, erhalten wir $4*4*4 = 64$ Multiplikationen und $4*4*3 = 48$ Additionen. Für den Strassen-Algorithmus ergeben sich (s.o.) $7*8 = 56$ Multiplikationen,

$4*7+4*6+24=76$ Additionen und $4*4+8=24$ Subtraktionen. Wir sehen dass die Anzahl der teueren Multiplikationen reduziert wird und dafür die Anzahl der billigen Operationen „+,-“ steigt.

Hinweis: Bei der vollständigen Ausführung des Strassen-Algorithmus würden wir die Matrixprodukte innerhalb der Berechnung der $\mathbf{q_i}$'s ein weiteres Mal aufteilen. Dies würde die Anzahl an Multiplikationen weiter senken.

Aufgabe 5 Lineare Gleichungssysteme (Beispiellösung)

a) Wir verwenden die Beziehung zwischen der Adjunkten ($\text{adj}(A)$) und der Inversen der Matrix.

$$A^{-1} = \frac{1}{\det(A)}\text{adj}(A) = \frac{1}{6}\begin{pmatrix} 2 & -2 \\ 2 & 1 \end{pmatrix} \qquad \mathbf{x} = \begin{pmatrix} 8 \\ 5 \end{pmatrix}$$

Hinweis: Leider gibt es immer wieder Verwechslungen zwischen der Adjunkten und der Adjungierten der Matrix. Erste, welche wir hier verwendet haben, bezeichnet die Transponierte der Kofaktormatrix. Während zweitere im Falle einer Matrix $A \in \mathbb{R}^{n\times m}$ ($A \in \mathbb{C}^{n\times m}$) die normale Transponierte (Hermitsche) der Matrix bezeichnet.

b) Lineare Abhängigkeit von Zeilen bzw. Gleichungen I und III führt zu Null-Zeile während der Gauss-Elimination. Also $\text{rank}(A) = 2 < 3$ und $\det(A) = 0$, und es existiert keine Inverse!

c) Wir erweitern die Matrix um die Identitätsmatrix und führen die Gauss-Jordan Elimination durch:

$$\left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 2 & 5 & 3 & 0 & 1 & 0 \\ 1 & 0 & 8 & 0 & 0 & 1 \end{array}\right] \xrightarrow[R_2=R_2-2\cdot R_1]{R_3=R_3-R_1} \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & -2 & 5 & -1 & 0 & 1 \end{array}\right]$$

$$\xrightarrow{R_3=-R_3-2\cdot R_2} \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right] \xrightarrow[R_2=R_2+3\cdot R_3]{R_1=R_1-3\cdot R_3} \left[\begin{array}{ccc|ccc} 1 & 2 & 0 & -14 & 6 & 3 \\ 0 & 1 & 0 & 13 & -5 & -3 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right]$$

$$\xrightarrow{R_1=R_1-2\cdot R_2} \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & -40 & 16 & 9 \\ 0 & 1 & 0 & 13 & -5 & -3 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right]$$

Damit folgt nun:

$$A^{-1} = \begin{pmatrix} -40 & 16 & 9 \\ 13 & -5 & -3 \\ 5 & -2 & -1 \end{pmatrix}$$

Aufgabe 6 Matrix-Zerlegungen (Beispiellösung)

a) Gültig, $P \cdot A = L \cdot U$.

b) Ungültig, $A = L \cdot L^T$, aber L nicht untere Dreiecksmatrix mit strikt positiven Diagonaleinträgen.

c) Ungültig, $A = Q \cdot R$, aber $Q^{-1} \neq Q^T$, also nicht orthogonal.

d) Ungültig, $A = U\Sigma V^T$, aber die Singulärwerte sind nicht absteigend sortier.