Matthias Krause
Alexander Moch

Mannheim, den 23.11.2020

CS 307 Algorithmen und Datenstrukturen, Herbstsemester 2020

Lösungsskizze zum Übungsblatt 9

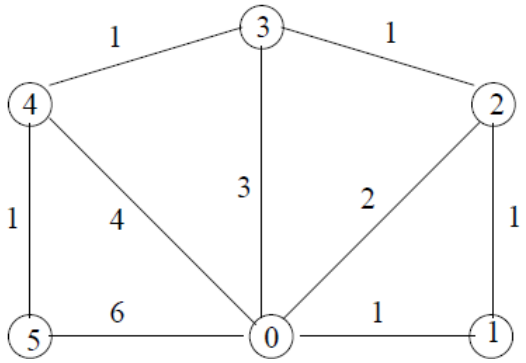
AUFGABE 9.1:

Welche Laufzeit können Sie für den Algorithmus von Kruskal garantieren, wenn bekannt ist, dass die Kantengewichte aus der Menge $\{1, \ldots, |V|^2\}$ (V ist die Knotenmenge) stammen?

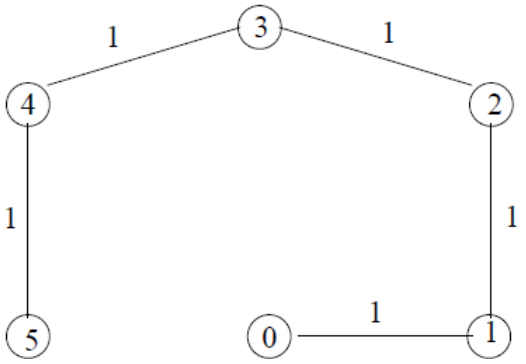
Man kann die Kanten mit Hilfe von Radixsort in Zeit und Platz $O(|V| + |E|)$ sortieren. Sei der Graph zusammenhängend. Der Rest des Algorithmus braucht (bei Anwendung einer geeigneten UNION-FIND-Datenstruktur, speziell mit Pfadkompression, siehe Folien) Zeit $O(|E| \cdot \alpha(|V|))$, also ist die Gesamtzeit in $O(|E| \cdot \alpha(|V|))$.

AUFGABE 9.2$^{\mathbf{K}}$:

Gegeben ist der folgende (ungerichtete) Graph G.



Wenden Sie auf G den Algorithmus von Prim (beginnend mit dem Knoten 0) an. Geben Sie jede Veränderung des Feldes *key* (oder d) und den berechneten Spannbaum an.
Zuerst werden alle key-Werte auf ∞ gesetzt. Dann gibt es folgende Veränderungen:
key[0]=0, *0:* *key*[1]=1, *key*[2]=2, *key*[3]=3, *key*[4]=4, *key*[5]=6, *1:* *key*[2]=1 *2:* *key*[3]=1 *3:* *key*[4]=1 *4:* *key*[5]=1, *5:*
Der minimale Spannbaum:



AUFGABE 9.3:

Gegeben sei der gewichtete gerichtete Graph $G = (V, E, w)$, mit

$$V = \{s, u, v, x\};$$
$$E = \{(s,u), (u,v), (v,x), (x,u), (s,v), (s,x)\};$$
$$w(s,u) = 1,\ w(u,v) = -3,\ w(v,x) = w(x,u) = w(s,v) = 2,\ w(s,x) = -1.$$

a) Wie sieht $G_\pi = (V_\pi, E_\pi)$ nach Ausführung der Operationen Initialize(G, w, s), Relax(s, u, w), Relax(u, v, w), Relax(v, x, w), Relax(s, v, w), Relax(s, x, w) aus?

Lösung: $V_\pi = V$, $E_\pi = \{(s,u), (u,v), (s,x)\}$

b) Gegeben sei nun der geringfügig modifizierte Graph $G' = (V', E', w')$ mit $V' = V$, $E' = E$ und

$$w'(e) = \begin{cases} -7 & \text{für } e = (u,v) \\ w(e) & \text{sonst} \end{cases}.$$

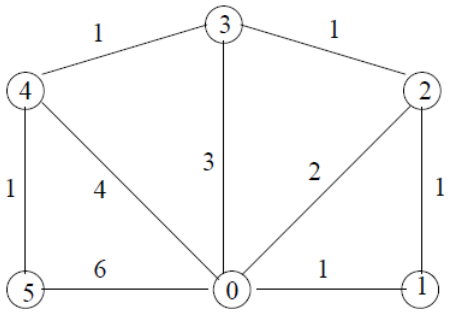
Sprich: G' entsteht aus G durch ändern des Gewichts der Kante (u, v) von $w(u, v) = -3$ (bei G) zu $w'(u, v) = -7$ (bei G').

Man gebe nun eine möglichst kurze Folge von Relaxoperationen an, deren Ausführung auf dem Ergebnis von Initialize(G', w', s) einen Graphen $G'_\pi = (V'_\pi, E'_\pi)$ erzeugt, der kein Baum mit Wurzel s ist.

Lösung: bspw.
Initialize(G', w', s), *Relax*(s, u, w'), *Relax*(u, v, w'), *Relax*(v, x, w'), *Relax*(x, u, w').
Der zugehörige Graph $G'_\pi = (V'_\pi, E'_\pi)$, $E'_\pi = \{(u,v), (v,x), (x,u)\}$, *weist (in Form von ganz E) offenbar einen Kreis auf und ist somit per Definition kein Baum.*

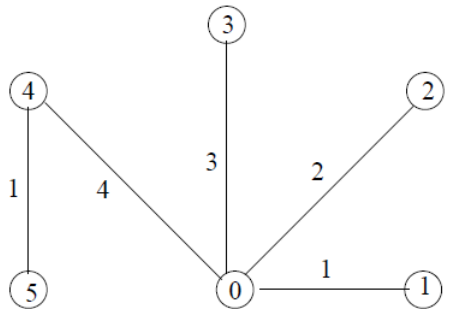
AUFGABE 9.4$^{\mathbf{K}}$:

Gegeben ist der folgende (ungerichtete) Graph G.



Wenden Sie auf G den Algorithmus von Dijkstra (beginnend mit dem Knoten 0) an. Geben Sie jede Veränderung des Feldes d und den vollständigen Baum der kürzesten Wege an.

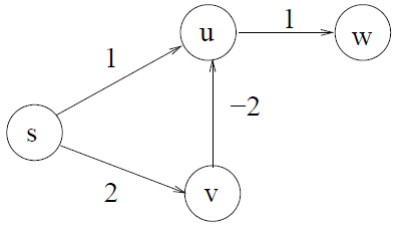
Zuerst werden alle d-Werte auf ∞ *gesetzt. Dann gibt es folgende Veränderungen:*
$d[0]=0$, *0:* $d[1]=1$, $d[2]=2$, $d[3]=3$, $d[4]=4$, $d[5]=6$, *1: 2: 3: 4:* $d[5]=5$, *5:*
Der Baum der kürzesten Wege:



AUFGABE 9.5:

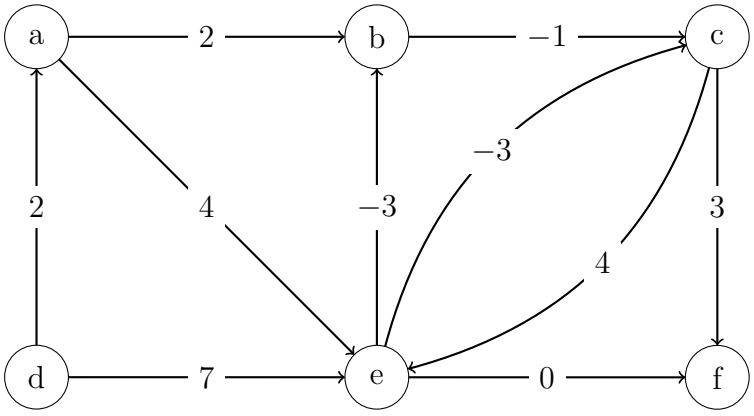
Beweisen Sie oder finden Sie ein Gegenbeispiel: Der Algorithmus von Dijkstra liefert auch dann immer korrekte Ergebnisse, wenn negative Kantengewichte, aber keine Kreise mit negativem Gesamtgewicht existieren.

Die Aussage ist falsch, betrachte das Beispiel ($d[w]$ *sollte* 1 *sein, wird aber auf* 2 *gesetzt*)*:*



AUFGABE 9.6:

Betrachten Sie den folgenden Graphen:



Benutzen Sie den Bellman-Ford-Algorithmus, um den kürzesten Weg vom Knoten d zu Knoten f zu finden. Die Kantenreihenfolge hierbei ist die kanonische:
$((a,b),(a,e),(b,c),(c,e),(c,f),(d,a),(d,e),(e,b),(e,c),(e,f))$.

Die untenstehende Tabelle gibt jeweils den Stand am Ende einer Runde (bei der für alle Kanten in obiger Reihenfolge die `RELAX`*-Operation ausgeführt wurde) an. Die fett gedruckten Werte haben sich im Vergleich zur Vorrunde verändert. Da nach der dritten Runde keine änderung mehr eintritt, wurden Runden 4 und 5 weggelassen.*
Also ist der kürzeste Pfad mit Länge 5*:*

	Init		Runde 1		Runde 2		Runde 3	
v	$v.d$	$v.\pi$	$v.d$	$v.\pi$	$v.d$	$v.\pi$	$v.d$	$v.\pi$
a	∞	**-**	**2**	**d**	2	d	2	d
b	∞	**-**	**4**	**e**	**3**	e	3	e
c	∞	**-**	**4**	**e**	**3**	**b**	**2**	b
d	**0**	**-**	0	-	0	-	0	-
e	∞	**-**	**7**	**d**	**6**	**a**	6	a
f	∞	**-**	**7**	**e**	**6**	**c**	**5**	c