Matthias Krause
Alexander Moch

Mannheim, den 23.11.2020

CS 307 Algorithmen und Datenstrukturen, Herbstsemester 2020

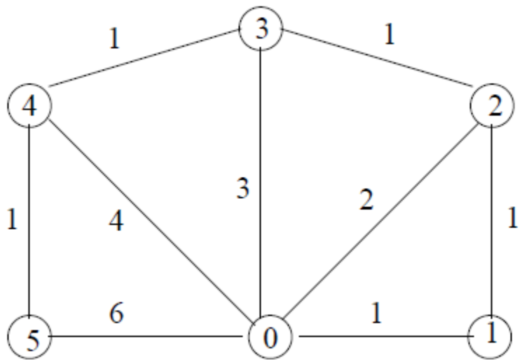
Übungsblatt 9

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?

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.

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?

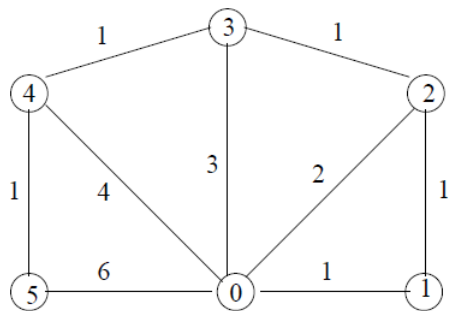
b) Gegeben sei nun der geringfügig modifizierte Graph $G' = (V', E', w')$ mit $V' = V$, $E' = E$ und

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

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.

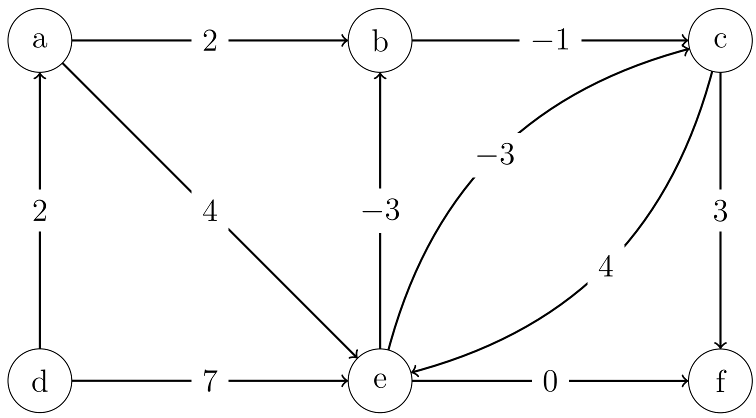
AUFGABE 9.4^K:
Gegeben ist der folgende (ungerichtete) Graph G.



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.

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.

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:
$((a, b), (a, e), (b, c), (c, e), (c, f), (d, a), (d, e), (e, b), (e, c), (e, f))$.