Datenstrukturen und Algorithmen
Heimübung 11

Eli Kogan-Wang (7251030)
David Noah Stamm (7249709)
Daniel Heins (7213874)
Tim Wolf (7269381)

26. Juni 2022

Aufgabe 1

Sei $G = (V, E)$ ein ungerichteter Graph.
Zu zeigen: Nach DFS(G, s) für ein $s \in V$ ist jede von s erreichbare Kankte $e \in E_{\text{connected } s}$ eine Baumkante des DFS-Baums, eine Rückwärts/Vorwärtskante.
Beweis: Sei $e \in E_{\text{connected } s}$ eine von s erreichbare Kante. Das heißt, dass für $e = v_1, v_2$ mit $v_1 \neq v_2$, der Startknoten v_1 im DFS-Baum liegt, und da der Baum bzgl. der Zusammenhangskomponente von s vollständig ist, auch v_2 im DFS-Baum liegt.
Nun lässt sich eine Fallunterscheidung durchführen:
Wenn e im DFS-Baum liegt, dann ist e eine Baumkante.
Wenn e nicht im DFS-Baum liegt, dann ist e eine Rückwärts/Vorwärtskante. Da e zwei Knoten aus dem DFS-Baum verbindet. Betrachtet man e als gerichtet, so kann man betrachten, ob v_1 vor oder nach v_2 in der topologischen Sortierung des Baumes vorkommt.

Aufgabe 2

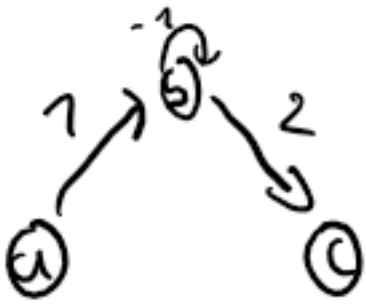
Wir beginnen mit dem Ergebnis des Algorithmus Tran: E^*. $O(|V|^2 + |V| \cdot |E|)$
In $O(|V|^2)$ reduzieren wir E^* zu $E^{*\prime} = \{(u, v) | \text{Es existiert ein Weg in } G \text{ von } u \text{ nach } v \text{ und von } v \text{ nach } u\}$.
Und reduzieren dann $E^{*\prime}$ zur ungerichteten unterliegenden Kantenmenge $E^{*\prime\prime} = \{\{u, v\} | \text{Es existiert ein Weg in } G \text{ von } u \text{ nach } v \text{ und von } v \text{ nach } u\}$ auch in $O(|V|^2)$.
Nun führen wir Suchen (Tiefen oder Breitensuche) maximal $|V|$ mal in $G'' = (V, E^{*\prime\prime})$ durch, um die einzelnen Zusammenhangskomponenten zu entfernen.
$|V|$ mal, da es maximal $|V|$ Zusammenhangskomponenten (eine für jeden Knoten) gibt.
Das geschieht in einer Laufzeit von $O((|V| + |E|) \cdot |V|) = O(|V|^2 + |V| \cdot |E|)$.

Der Algorithmus beruht auf der Korrektheit von Tran und extrahiert mithilfe von korrekten Suchen starke Zusammenhangskomponenten auf. Die Starken zusammenhangskomponenten bilden innerhalb der doppelt transitiven ungerichteten Hülle die regulären starken Zusammenhangskomponenten, die klassisch mit Suchen entdeckt werden können.

Aufgabe 3

a) Der Algorithmus funktioniert wie zuvor.
Sei (u, v) eine Kante mit Gewichtung 0.
Nachdem u aus dem Heap entfernt wurde und zur Menge S der entdeckten Knoten hinzugefügt wurde, wird $d(v)$ auf $d(u) + 0$ gesetzt und v darauffolgend entfernt (mit Knoten mit derselben Distanz).
Damit wird v Zeitlich richtig im Algorithmus entfernet.

b) Wir betrachten diesen Graphen:



Dijkstra von a aus:

$S = \{a\}$

$V \backslash S$	b	c
$d(a)$	1	∞
$\pi(a)$	a	∞

$S = \{a, b\}$ $d(b) = 1$

$V \backslash S$	c
$d(a)$	2
$\pi(a)$	b

Nach Dijkstra ist der kürzeste Weg: $a \to b \to c$ mit Gesamtlänge 3.
$\lightning$ kürzester Weg ist $a \to b \to b \ldots b \to c$ mit Gesamtlänge $-\infty$.