Karlsruher Institut für Technologie
Institut für Theoretische Informatik

Prof. Dr. Dennis Hofheinz
Lukas Barth, Lisa Kohl

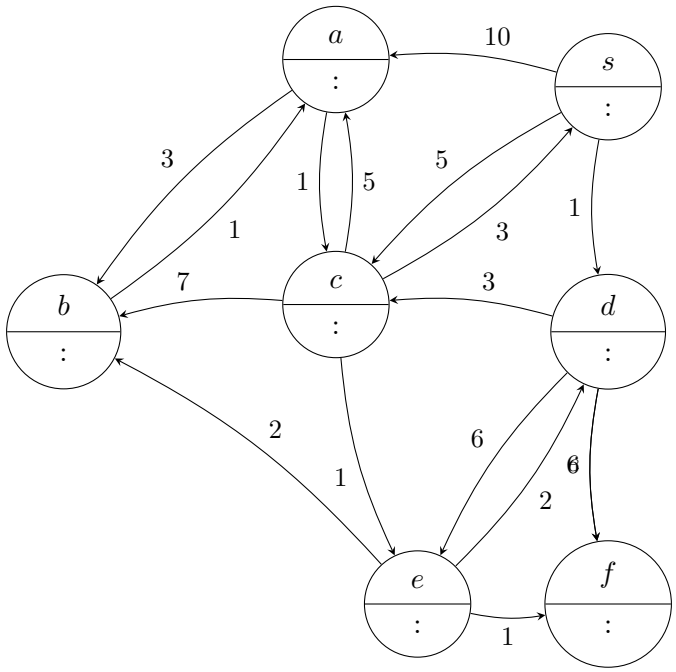
9. Übungsblatt zu Algorithmen I im SoSe 2016

https://crypto.iti.kit.edu/index.php?id=algo-sose16
{lisa.kohl,lukas.barth}@kit.edu

Aufgabe 1 (*Dijkstra,* 4 *Punkte*)

Wir beschäftigen uns mit dem Algorithmus von Dijkstra aus der Vorlesung. Sei ein Graph $G = (V, E)$ wie unten dargestellt. Führen Sie Dijkstras Algorithmus auf G aus. Sie können die Darstellung von G benutzen, um die Ergebnisse nach der Ausführung darin einzutragen. Schreiben Sie dazu in jeden Knoten $v \in V$ rechts vom Doppelpunkt die Länge des kürzesten Pfads vom Startknoten s aus gesehen und links vom Doppelpunkt den Vorgänger auf diesem kürzesten Pfad. Zeichnen Sie den Baum der kürzesten Wege von s zu allen anderen Knoten aus V in G ein. Geben Sie die Reihenfolge der Knoten an, in der sie als *scanned* markiert werden. Geben Sie außerdem für jeden Knoten $v \in V$ alle Zwischenwerte $d[v]$ und parent$[v]$ an, die während Ausführung des Algorithmus angenommen werden.

Darstellung von G:



Aufgabe 2 (*Unzuverlässige Kommunikation,* 4 *Punkte*)

Sie sind Administrator eines Computernetzwerks bestehend aus n Rechnern und Netzwerkverbindungen zwischen diesen. Leider sind die Netzwerkverbindungen nicht absolut zuverlässig: Jede Verbindung zwischen zwei Rechnern a und b hat eine Wahrscheinlichkeit $0 < p_{(a,b)} < 1$, dass ein auf dieser Verbindung gesendetes Paket nicht ankommt. Die Verbindungen sind gerichtet, das heißt es kann sein, dass $p_{(a,b)} \neq p_{(b,a)}$ gilt.
Ihre Aufgabe ist es nun, zwischen zwei gegebenen Rechnern einen möglichst zuverlässigen Pfad zu finden. Geben Sie hierfür einen möglichst effizienten Algorithmus an, und begründen Sie kurz, warum dieser funktioniert.

Aufgabe 3 (*Verbesserter Bellman-Ford-Algorithmus,* $1 + 4$ *Punkte*)

Gegeben sei folgende Verbesserung des Bellman-Ford-Algorithmus: Im gerichteten Graphen $G = (V, E)$ weisen wir zunächst den Knoten eine beliebige aber feste Ordnung zu. Die Knoten seien nun $V = \langle v_1, v_2, v_3, \dots v_n \rangle$. Nun können wir E in zwei Teile $E_f \cup E_b = E$ aufteilen, sodass $E_f := \{(v_i, v_j) \in E \mid i < j\}$ und $E_f := \{(v_i, v_j) \in E \mid i > j\}$ gilt. Intuitiv enthält E_f also alle Kanten, die von einem *kleineren* auf einen *größeren* Knoten zeigen (E_b entsprechend umgekehrt).
Weiterhin sei folgende Änderung des Bellman-Ford-Algorithmus gegeben: Anstatt in jedem der n Durchläufe alle Kanten in irgendeiner Reihenfolge zu relaxieren, gehen wir in jedem Durchlauf wie folgt vor: Wir betrachten die Knoten in der Reihenfolge $v_1, v_2, v_3, \dots v_n$ und relaxieren dabei für jeden Knoten alle Kanten in E_f, die den Knoten verlassen. Danach betrachten wir alle Knoten in der Reihenfolge $v_n, v_{n-1}, \dots v_1$ und relaxieren dabei für jeden Knoten alle Kanten in E_b, die den Knoten verlassen.

a) Zeigen Sie: $G_f = (V, E_f)$ und $G_b = (V, E_b)$ sind jeweils DAGs.

b) Zeigen Sie: Die veränderte Version des Bellman-Ford-Algorithmus berechnet bereits nach $\lceil n/2 \rceil$ Schritten das Ergebnis, d.h. einen kürzesten $s - t$-Pfad oder (mindestens) einen negativen Kreis. *Tipp*: Überlegen Sie sich, was der Algorithmus auf den folgenden zwei Graphen tut. **Diese beiden Graphen sind nur als Hilfestellung angegeben. Zur Lösung dieser Aufgabe ist es nicht erforderlich, dass Sie den Algorithmus auf ihnen ausführen.** Es hilft Ihnen aber hoffentlich.

   (a) Ein einfacher Pfadgraph: $V = \{1, 2, \dots n\}, E = \{(i, i+1) \mid i \in \{1, 2 \dots n-1\}\}$

   (b) Zwei aufeinander zulaufende Pfadgraphen:

$$V = \{1, 2, \dots n\}$$
$$E = \{(i, i+1) \mid i \in \{1, 2 \dots \lceil \frac{n}{2} - 1 \rceil\}\} \cup \{(i+1, i) \mid i \in \{\lceil \frac{n}{2} - 1 \rceil \dots n-1, n\}\}$$

Aufgabe 4 (*Doktor Meta,* 6 *Punkte*)

Als der ebenso orientierungsschwache wie unermüdliche Untersuperbösewicht Røbøt endlich in Doktor Metas geheimer Basis ankommt, ist Doktor Meta über die Verspätung und die übrige Ausbeute von 1-Cent-Stücken alles andere als erfreut. Um ihn in bessere Laune zu versetzen, erzählt Røbøt von den Begegnungen mit verschiedenen Bösewichten auf seinen Irrwegen durch das Tunnelsystem.
Ein Superbösewicht hat genug von unzuverlässigen elektronischen Gehilfen und bietet deshalb eine große Menge von identischen Bauteilen vom Typ A für eine Preis von $1,25$ Euro pro Bauteil zum Verkauf an. Ein Mittelbösewicht hat den Plan ein Roboterheer aufzubauen noch nicht aufgegeben hat und ist verzweifelt auf der Suche nach Bauteilen von Typ A. Er ist bereit je zwei Bauteile vom Typ A für je drei Bauteile vom Typ B, die er selbst noch übrig hat, zu tauschen. Ein vom gleichen Bausatz stammende und ähnlich fehlkonstruierte Unterbösewicht hat sich ebenso mit einer Menge Geld im Gepäck verlaufen. Sein Plan ist es einen Unterbösewichtsgehilfen zu bauen, der den Weg für ihn findet. Dazu möchte er Bauteile vom Typ B zum Preis von $0,875$ Euro pro Bauteil erwerben.
Plötzlich ist Doktor Meta nicht mehr schlecht gelaunt, er rechnet nach, dass er so einen Euro ganz einfach in $1,05$ Euro verwandeln kann. Damit Røbøt beim nächsten Mal eine solche Gelegenheit von selbst erkennt, möchte Doktor Meta einen geeigneten Algorithmus entwerfen.
Seien dazu Allgemein eine Menge von angebotenen Objekten $a_1, \dots, a_n$ (im vorliegenden Fall Euros, Bauteile vom Typ A und Bauteile vom Typ B) gegeben und weiter eine Tabelle T, wobei $T[i, j]$ jeweils angibt wieviele Einheiten von Objekt a_j für ein Objekt a_i angeboten werden.
Im vorliegenden Fall ergibt sich also $T[1, 2] = 0,8$, $T[2, 3] = 1,5$ und $T[3, 1] = 0,875$ und damit $T[1, 2] \cdot T[2, 3] \cdot T[3, 1] = 1,05 > 1$.
Modellieren Sie das Problem mit Hilfe von einem geeigneten Graphen und konstruieren Sie einen effizienten Algorithmus, der entscheidet ob es eine Folge von Objekten $a_{i_1}, \dots, a_{i_k}$ gibt, sodass $T[i_1, i_2] \cdot T[i_2, i_3] \cdots T[i_k, i_1] > 1$. Analysieren Sie die Laufzeit Ihres Algorithmus.

Ausgabe: Mittwoch, 22.06.2016
Abgabe: Freitag, 01.07.2016, 12:45 im Briefkasten im Untergeschoss von Gebäude 50.34

Deckblatt Übungsblatt 9
Algorithmen I

A1	**A2**	**A3**	**A4**	$\sum$

Tutoriumsnummer:

Name	**Matrikelnummer**	**Unterschrift**

Mit unseren Unterschriften bestätigen wir, dass die Aufgaben von den Unterzeichnern eigenständig gelöst worden sind.

Hinweis: Das Übungsblatt darf in Gruppen von bis zu zwei Personen bearbeitet werden. Beide Personen müssen demselben Tutorium zugeteilt sein. Möchte jemand seine Abgaben-Gruppe innerhalb des Semesters wechseln, so ist dies im Voraus mit dem Tutor abzusprechen. **Bitte tragen Sie in das obere Quadrat *groß* die Nummer Ihres Tutoriums ein.** Die Lösung des Übungsblattes ist in jedem Fall mit diesem Deckblatt abzugeben.