Universität des Saarlandes
FR Informatik

Kurt Mehlhorn und Corinna Coupette **WiSe 2020/21**

Übungen zu Ideen der Informatik

https://www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/winter20/ideen/

Blatt 6 Abgabeschluss: 14.12.2020

Aufgabe 1 (*10 Punkte*)

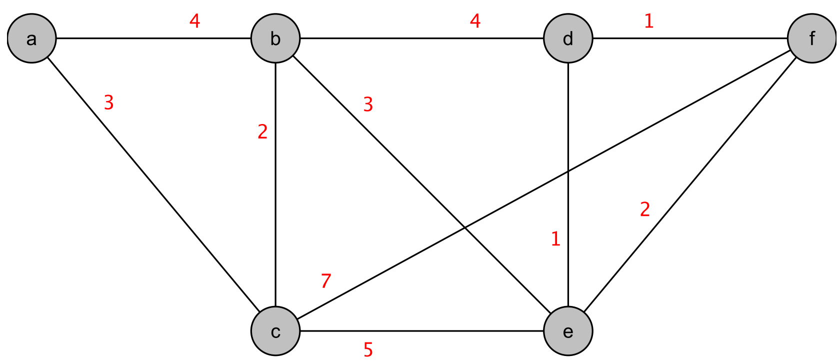

Abbildung 1: Beispielgraph für Aufgabe 1

Betrachten Sie den Graphen in Abbildung 1 mit in Rot annotierten Kantenlängen (auch Kantengewichte genannt). Alle Kanten können in beide Richtungen durchlaufen werden.

a) Finden Sie mithilfe von Dijkstras Algorithmus die kürzesten Wege von Knoten a zu allen anderen Knoten und geben Sie dabei alle Zwischenschritte an. Bei Gleichstand bearbeiten Sie die Knoten in alphabetischer Reihenfolge. (*7 Punkte*)

Lösung:

Runde (neu schwarz)		**a**	**b**	**c**	**d**	**e**	**f**
0 (–)	dist[·]	0	∞	∞	∞	∞	∞
1 (a)	dist[·]	0	4	3	∞	∞	∞
2 (c)	dist[·]	0	4	3	∞	8	10
3 (b)	dist[·]	0	4	3	8	7	10
4 (e)	dist[·]	0	4	3	8	7	9
5 (d)	dist[·]	0	4	3	8	7	9
6 (f)	dist[·]	0	4	3	8	7	9

b) Betrachten Sie noch einmal den Pseudocode von Dijkstras Algorithmus, der in der Vorlesung eingeführt wurde. Der Algorithmus berechnet aktuell die Distanz eines ausgewählten Knotens zu allen anderen Knoten, ohne dass sich die kürzesten Wege selbst unmittelbar aus dem Ergebnis erschließen ließen.

Wie und wo müssten Sie den Algorithmus ergänzen, um am Ende in weniger als n Schritten (für einen Graphen mit n Knoten) den kürzesten Pfad zwischen dem ausgewählten Startknoten und einem ausgewählten Zielknoten ermitteln zu können, wenn Sie nicht genug Platz haben, um alle kürzesten Pfade direkt zu speichern? (*3 Punkte*)

Lösung: Sie führen eine zusätzliche Datenstruktur parent[$\cdot$] ein, in der Sie festhalten, von welchem anderen Knoten aus ein Knoten v erreicht wird, wenn Sie die Länge des kürzesten Weges aktualisieren (im Beispielgraph wäre am Ende z.B. parent[b] = a und parent[e] = b). Wenn Sie dann nach der Distanz zwischen dem Startknoten s und einem Zielknoten t gefragt werden, können Sie den Weg finden, indem Sie beginnend bei t die Einträge in parent[$\cdot$] zurückverfolgen, bis Sie s erreichen. Da der Graph n Knoten hat und ein kürzester Weg keinen Knoten doppelt besucht, müssen Sie höchstens $n-1$ Werte nachschlagen.

Aufgabe 2 (*10 Punkte*)

Wir nehmen nun an, dass es auch negative Kantengewichte geben darf (die nächste Aufgabe gibt ein Beispiel für die Sinnhaftigkeit von negativen Kantengewichten). Ein Weg mit Kanten der Länge 5, -5, 1 hat Gesamtlänge $5 + (-5) + 1 = 1$. Im folgenden Beispielgraphen kann man die Kanten in der oberen Reihe nur in Pfeilrichtung und die anderen Kanten in beide Richtungen benutzen.

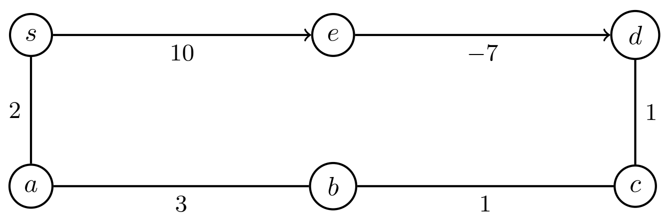

Abbildung 2: Beispielgraph für Aufgabe 2

a) Geben Sie die Entfernungen von s zu allen Knoten (einschließlich s) an. (*3 Punkte*)

Lösung:

$s \to s:\ 0$

$s \to a:\ 2$

$s \to b:\ 5$

$s \to c:\ 4$

$s \to d:\ 3$

$s \to e:\ 10$

b) Führen Sie (nach denselben Bedingungen wie in Aufgabe 1) Dijkstras Algorithmus auf dem Beispielgraphen aus und geben Sie das Ergebnis an (diesmal müssen Sie die Zwischenschritte nicht aufführen). (*3 Punkte*)

Lösung: Mit Dijkstras Algorithmus ermittelte Distanzen von s zu allen Knoten (einschließlich s):

$s \to s:\ 0$

$s \to a:\ 2$

$s \to b:\ 5$

$s \to c:\ 6$

$s \to d:\ 3$

$s \to e:\ 10$

c) Was schließen Sie hinsichtlich der Korrektheit von Dijkstras Algorithmus auf diesem Graphen und was könnte die Ursache für das beobachtete Verhalten sein? (*2 Punkte*)

Lösung: Dijkstras Algorithmus funktioniert auf dem Beispielgraphen nicht korrekt. Der Grund ist, dass der Algorithmus jeden Knoten nur einmal abarbeitet. Das funktioniert bei nicht-negativen Kantengewichten, weil wir dann wissen (s. Vorlesung), dass dist[y] der kürzeste Weg von s nach y ist, wenn wir den Knoten y schwarz färben. Wenn Kantengewichte negativ sein können, kann dies nicht mehr garantiert werden (wie im Beispielgraphen, für den wir d mit Distanz 7 schwarz markieren, obwohl der kürzeste Weg von s zu d die Länge 3 hat).

d) Funktioniert der erste in der Vorlesung vorgestellte Algorithmus (angepasst so, dass er auf Graphen mit Kanten, die in beide Richtungen durchlaufen werden können und nicht-negative Kantengewichte haben, grundsätzlich funktioniert) auf diesem Graphen? Warum bzw. warum nicht? (*2 Punkte*)

Lösung: Der erste in der Vorlesung vorgestellte Algorithmus funktioniert noch. Er würde nämlich nicht aufhören, sondern bemerken, dass er die Distanz $s \to c$ auf 4 verringern kann.

Aufgabe 3 (*10 Punkte*)

Sie betreiben ein Regionalverkehrsunternehmen in einem Schienennetz. Bei dem Betrieb entstehen Ihnen für jeden Streckenabschnitt zwischen zwei Stationen x und y bestimmte Kosten c_{xy}; außerdem erhalten Sie für das Bedienen eines Streckenabschnitts ein bestimmtes, von Ihren Kosten unabhängig festgelegtes Entgelt r_{xy}. Die Situation ist in Abbildung 3 dargestellt: Jeder Knoten ist eine Station, jede Kante ist ein Streckenabschnitt und mit Ihren Kosten (Ihrem Entgelt) in Rot (Blau) beschriftet.

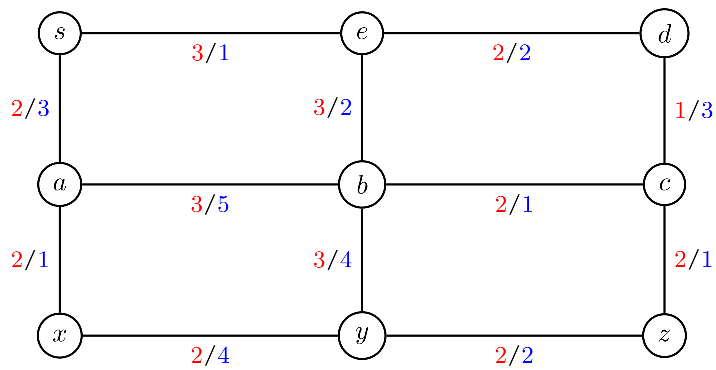

Abbildung 3: Beispielgraph für Aufgabe 3

Ihr Unternehmen ist aktuell in Knoten s beheimatet und kann vorerst nur einen Zug betreiben, der eine Rundtour fährt: eine Tour, die in Ihrem Heimatknoten beginnt und wieder endet und jeden Streckenabschnitt maximal einmal überquert.

a) Sie wollen wissen, ob es eine Rundtour gibt, auf der das Unternehmen Geld verdient. Welche Voraussetzung in Bezug auf die Kosten und Entgelte der Kanten muss erfüllt sein, damit dies der Fall ist? Unter welcher Voraussetzung ist eine Tour ein Verlustgeschäft – und wann ist die Tour wirtschaftlich neutral? (*3 Punkte*)

Lösung: Sei E die Menge der auf einer Tour überquerten Kanten. Eine Kante hat Kosten c_e und Entgelt r_e. Die Summe S der Entgelte abzüglich Kosten ist also $S = \sum_{e \in E}(r_e - c_e) = \sum_{e \in E} r_e - \sum_{e \in E} c_e$. Voraussetzung für die Profitabilität einer Tour ist $S > 0$. Ist $S < 0$, machen Sie Verluste; für $S = 0$ ist die Tour wirtschaftlich neutral.

b) Geben Sie eine Rundtour von s aus an, die profitabel ist, und belegen Sie die Profitabilität. (*2 Punkte*)

Lösung: $s \to e \to d \to c \to b \to a \to s$.

Die Kosten sind $C = 3 + 2 + 1 + 2 + 3 + 2 = 13$ und die Entgelte sind $R = 1 + 2 + 3 + 1 + 5 + 3 = 15$, ihr Profit is also $S = R - C = 15 - 13 = 2$.

c) Um die Profitabilität Ihres Unternehmens zu erhöhen, erwägen Sie, Ihren Unternehmenssitz von s weg zu einem anderen Knoten hin zu verlegen. Angenommen, dass Sie weiterhin nur eine Rundtour bedienen können: Welches Kriterium muss Ihr neuer Standort erfüllen, wenn Sie Ihre Profitabilität maximieren wollen? (*2 Punkte*)

Lösung: Der Standort sollte Teil einer Rundtour mit für das gegebene Schienennetz maximalem Profit sein. Sollte es mehrere Touren mit gleichem maximalen Profit geben, könnten Sie als zusätzliches Entscheidungskriterium versuchen, den durchschnittlichen Profit pro Streckenabschnitt zu maximieren (dann würden kürzere Touren bevorzugt) oder die Netzabdeckung zu maximieren (dann würden längere Touren bevorzugt).

d) Geben Sie einen geeigneten neuen Standort an und begründen Sie Ihre Wahl. (*3 Punkte*)

Lösung: Die Standorte s und z sind ungeeignet, da Sie bei der An- und Abreise gezwungenermaßen unnötige Verluste machen und die Standorte daher nicht Teil der profitabelsten Rundtour sein können. Eine Rundtour mit maximalem Profit ist $b \to e \to d \to c \to b \to y \to x \to a \to b$ (oder eine strukturgleiche Variante dieser Tour) mit $S = R - C = (2+2+3+1+4+4+1+5) - (3+2+1+2+3+2+2+3) = 22 - 18 = 4$; eine *minimale* Rundtour mit gleichem Profit wäre $b \to y \to x \to a \to b$. Alle Standorte bis auf s und z sind daher grundsätzlich geeignet, wobei Sie zwischen den Standorten $\{a, b, x, y\}$ wählen sollten, wenn Sie zugleich darauf achten wollen, Ihren durchschnittlichen Profit pro Streckenabschnitt zu maximieren. Aus strategischen Erwägungen böte sich natürlich Standort b an, weil dieser von allen in Betracht kommenden Standorten insgesamt am besten vernetzt ist.

Zusatzaufgabe (*10 Punkte*)

Die Punkte, die Sie mit dieser Aufgabe erlangen können, sind Zusatzpunkte. Sie beeinflussen nicht die Mindestpunktzahl, die Sie benötigen, um zur Klausur zugelassen zu werden. Die von Ihnen erreichten Punkte werden aber mit Ihren anderen Punkten zusammengerechnet, um festzustellen, ob Sie die notwendige Mindestpunktzahl erreicht haben. Zusatzaufgaben erfordern in der Regel mehr eigenständiges Denken als normale Aufgaben. Auch wenn Sie keine Zusatzpunkte benötigen, um zur Klausur zugelassen zu werden, kann sich die Bearbeitung der Zusatzaufgabe daher für Sie (intellektuell) lohnen.

Nehmen Sie an, ein Ganove hat Ihre Festplatte verschlüsselt und den Code zur Entschlüsselung irgendwo in den Straßen oder an einer Straßenecke Ihrer Heimatstadt eindeutig erkennbar auf den Boden geschrieben. Der Ganove gibt Ihnen einen Tag Zeit, um ohne fremde Hilfe den Code zu finden und die Festplatte wieder zu entschlüsseln – scheitern Sie, wird er all Ihre Daten löschen.[1] Ihre Heimatstadt hat moderate Größe, sodass Sie alle Straßenabschnitte zweimal ganz ablaufen könnten, bevor der Tag endet. Sie ist aber zu kompliziert strukturiert, als dass Sie ihr gesamtes Straßennetz im Kopf behalten könnten. In Ihrer Tasche befindet sich eine ausreichend große Menge roter und blauer Kreide, mit der Sie in Straßen und an Straßenecken Markierungen anbringen können.

a) Überlegen Sie sich ein Verfahren, mit dem Sie mit Sicherheit den Code finden, bevor Ihre Daten gelöscht werden (Sie können dafür das Straßennetz als einen Graphen mit den Straßenecken als Knoten und den Straßenabschnitten als Kanten auffassen). (*4 Punkte*)

Lösung: Wir geben ein Verfahren an, wie wir systematisch die Kanten (Straßenabschnitte) und Knoten (Straßenecken) ablaufen können, indem wir sie unter bestimmten Bedingungen einfärben und unseren jeweils nächsten Schritt von den in unserem Umfeld vorgefundenen Markierungen abhängig machen.[2]

Am Anfang sind alle Knoten und Kanten des Graphen mit Ausnahme des Startknotens ungefärbt; den Startknoten färben wir blau.

Wenn wir uns auf einem blau gefärbten Knoten x befinden und es eine ungefärbte Kante $\{x, y\}$ gibt, betreten wir diese Kante und färben sie rot. Wenn der Knoten y ungefärbt ist, betreten wir im nächsten

[1] Eigentlich kein Problem, denn Sie haben ja ein aktuelles Backup – oder?

[2] Das beschriebene Verfahren nennt sich Tiefensuche (*depth-first search*, DFS).

Schritt y, markieren ihn blau und bringen am Straßenabschnitt $\{x, y\}$ eine Markierung an, die uns daran erinnert, dass wir y von x aus zum ersten Mal betreten haben, anderenfalls gehen wir zurück zu x.

Wenn wir uns auf einem blau gefärbten Knoten x befinden und es keine ungefärbte Kante $\{x, y\}$ gibt, färben wir den Knoten von blau zu rot um, und wenn x nicht der Startknoten ist, bewegen wir uns anschließend zurück zu dem Knoten, von dem aus wir x das erste Mal betreten haben.

Das Verfahren endet spätestens, wenn wir alle Straßen und Straßenecken rot eingefärbt haben und zum Startknoten zurückgekehrt sind, und bis dahin sollten wir den Code gefunden haben.

b) Erproben Sie Ihr Verfahren an dem Graphen von Aufgabe 3 (ignorieren Sie dafür die Kantenbeschriftungen) und geben Sie dabei alle Einzelschritte an. Sie starten im Knoten s. Der Code findet sich auf dem Straßenabschnitt zwischen y und z. (*2 Punkte*)

Lösung: Wir geben die eingefärbten Elemente in der Reihenfolge ihrer Einfärbung an (eine Kante $\{x, y\}$ repräsentieren wir kurz mit xy). Wenn wir zwischen mehreren Kanten auswählen können, wählen wir die Kante, deren Endpunkt im Alphabet am weitesten vorne steht.

$sa, ab, bc, cd, de, eb, es, e, d, cz, zy \rightarrow$ Code gefunden

c) Begründen Sie, warum Ihr Verfahren im Allgemeinen funktioniert und Sie alle Straßenabschnitte nicht mehr als zweimal komplett ablaufen. (*4 Punkte*)

Lösung: Zur Korrektheit des Verfahrens: Das Verfahren funktioniert, wenn wir alle Knoten und alle Kanten besuchen. Dies ist dann sichergestellt, wenn wir alle Knoten blau und alle Kanten rot färben. Das Verfahren selbst stellt sicher, dass wir alle Kanten rot färben, die von den Knoten ausgehen, die wir blau färben. Wir müssen also nur zeigen, dass jeder Knoten irgendwann blau gefärbt wird. Das geht am einfachsten mittels vollständiger Induktion.[3]

Induktionshypothese: Jeder Knoten wird irgendwann blau gefärbt.

Induktionsanfang (Knoten in Distanz 0 vom Startknoten):

Den Knoten in Distanz 0 vom Startknoten haben wir bereits blau gefärbt.

Induktionsschritt (Knoten in Distanz $d - 1 \rightarrow$ Knoten in Distanz d vom Startknoten):

Angenommen, wir haben bereits gezeigt, dass alle Knoten in Distanz $d-1$ vom Startknoten irgendwann blau gefärbt werden. Für jeden Knoten v in Distanz d vom Startknoten gibt es mindestens einen Nachbarn w, der in Distanz $d - 1$ vom Startknoten liegt. Nach der Induktionshypothese wird w irgendwann blau gefärbt. Das Verfahren selbst stellt sicher, dass wir alle Kanten, die von w ausgehen, irgendwann rot färben, und dass wir bei Erkundung der Kante $\{w, v\}$ zu v laufen, falls v noch nicht blau gefärbt ist. Also wird v irgendwann blau gefärbt.

Zur Laufzeit des Verfahrens: Wir überqueren jede Kante $\{x, y\}$ maximal zweimal: einmal, wenn wir sie rot färben, und einmal, nachdem wir y rot gefärbt haben, falls wir y zuerst von x betreten haben. Die Laufzeitanforderungen sind also erfüllt.

Ich habe für die Videos, die Nachbereitung und das Übungsblatt etwa ☐ Stunden gebraucht.

(Angelina fertigt aus diesen Zahlen eine Statistik an. Kurt und Corinna sehen nur diese Statistik. Wir möchten wissen, ob der Schwierigkeitsgrad in etwa richtig ist.)

Kürzeste Wege war	spannend ☐ okay ☐ langweilig ☐
	schwierig ☐ okay ☐ einfach ☐

[3]Siehe `https://de.wikipedia.org/wiki/Vollständige_Induktion`.