Kurt Mehlhorn und Corinna Coupette WiSe 2021/22

Übungen zu Ideen der Informatik

https://www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/winter21/ideen/

Blatt 7: Internet **Abgabeschluss: 13.12.2021**

Aufgabe 1 Navigation *(10 Punkte)*

Betrachten Sie Algorithmus 1, der kürzeste Wege von einem gegebenen Startknoten zu allen anderen Knoten in einem *gerichteten*, gewichteten Graphen berechnen soll (stellen Sie sich ein Straßennetz vor, das nur aus Einbahnstraßen besteht).

Zur Erinnerung: Wir notieren eine gerichtete Kante von u nach v als (u, v) und zeichnen sie als Pfeil von u nach v. Die Anzahl der Elemente in einer Menge M notieren wir mit $|M|$.

Algorithmus 1 : Kürzeste Wege

$G \leftarrow$ Eingabe (Graph);
$s \leftarrow$ Eingabe (Startknoten);
$V \leftarrow$ Knoten von G; `// Menge von Knoten`
$E \leftarrow$ Kanten von G; `// Menge von Kanten der Form` (u, v)
$Q \leftarrow V$; `// Menge von Knoten`
`// Nehmen Sie an, dass die Listen mit den Knotennamen indexiert sind`
d $\leftarrow$ Liste der Länge $|V|$, alle Einträge auf ∞ initialisiert; `// Liste der Distanzen`
p $\leftarrow$ Liste der Länge $|V|$, alle Einträge auf $\perp$ initialisiert; `// Liste der Knoten-Vorgänger`
d$[s] \leftarrow 0$;
while $Q \neq \{\}$ **do**
    $u \leftarrow$ Knoten in Q mit kleinstem d$[u]$ (bei Gleichstand: alphabetisch);
    Entferne u aus Q;
    $N \leftarrow \{v \mid v \in Q \wedge (u, v) \in E\}$;
    for $v \in N$ **do**
        alt $\leftarrow$ d$[u] + w(u, v)$; `//` $w(u, v)$ `ist das Gewicht der Kante von` u `nach` v
        if alt $<$ d[v] **then**
            d$[v] \leftarrow$ alt;
            p$[v] \leftarrow u$;

Betrachten Sie den Graphen in Abbildung 1. Jeder Knoten ist eine Straßenecke. Jede gerichtete Kante ist eine Einbahnstraße, deren Gewicht der Fahrzeit von ihrem Startknoten zu ihrem Zielknoten entspricht.

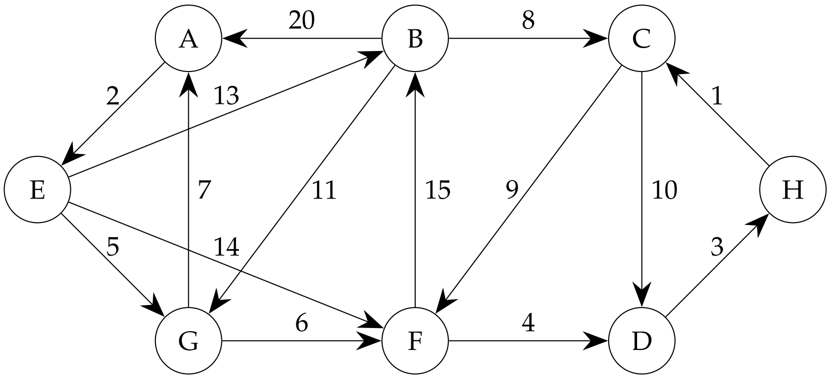


Abbildung 1: Einbahnstraßen

a) Führen Sie Algorithmus 1 mit Startknoten $s = A$ auf diesem Graphen aus und füllen Sie dabei die nachstehende Tabelle aus. Dabei soll die Tabelle in der mit „*Runde* x“ markierten Zeile den Zustand *nach* dem x-ten Durchlauf der *while*-Schleife enthalten; Runde 0 stellt den Zustand vor dem ersten Eintritt in die *while*-Schleife dar. (*4 Punkte*)

	A		B		C		D		E		F		G		H	
Runde	d	p	d	p	d	p	d	p	d	p	d	p	d	p	d	p
0	0	$\perp$	∞	$\perp$	∞	$\perp$	∞	$\perp$	∞	$\perp$	∞	$\perp$	∞	$\perp$	∞	$\perp$
1																
2																
3																
4																
5																
6																
7																
8																

Lösung:
Wir führen hier in jeder Spalte nur die jeweiligen Änderungen auf.

	A		B		C		D		E		F		G		H	
Runde	d	p	d	p	d	p	d	p	d	p	d	p	d	p	d	p
0	0	$\perp$	∞	$\perp$	∞	$\perp$	∞	$\perp$	∞	$\perp$	∞	$\perp$	∞	$\perp$	∞	$\perp$
1									2	A						
2			15	E							16	E	7	E		
3											13	G				
4							17	F								
5					23	B										
6															20	D
7					21	H										
8																

b) Mit Begründung: Was ist der kürzeste Weg von A nach H? (*1 Punkt*)

Lösung:
$A - E - G - F - D - H$, siehe Tabelle.

c) Führen Sie Algorithmus 1 mit Startknoten $s = H$ auf diesem Graphen aus und füllen Sie dabei die nachstehende Tabelle aus. Dabei soll die Tabelle in der mit „*Runde x*“ markierten Zeile den Zustand *nach* dem x-ten Durchlauf der *while*-Schleife enthalten; Runde 0 stellt den Zustand vor dem ersten Eintritt in die *while*-Schleife dar. (*4 Punkte*)

	A		B		C		D		E		F		G		H	
Runde	d	p	d	p	d	p	d	p	d	p	d	p	d	p	d	p
0	∞	$\perp$	∞	$\perp$	∞	$\perp$	∞	$\perp$	∞	$\perp$	∞	$\perp$	∞	$\perp$	0	$\perp$
1																
2																
3																
4																
5																
6																
7																
8																

Lösung:
Wir führen hier in jeder Spalte nur die jeweiligen Änderungen auf.

	A		B		C		D		E		F		G		H	
Runde	d	p	d	p	d	p	d	p	d	p	d	p	d	p	d	p
0	∞	$\perp$	∞	$\perp$	∞	$\perp$	∞	$\perp$	∞	$\perp$	∞	$\perp$	∞	$\perp$	0	$\perp$
1					1	H										
2							11	C			10	C				
3			25	F												
4																
5	45	B											36	B		
6	43	G														
7									45	A						
8																

d) Mit Begründung: Was ist der kürzeste Weg von H nach A? (*1 Punkt*)

Lösung:
$H - C - F - B - G - A$, siehe Tabelle.

Aufgabe 2 Infrastruktur *(10 Punkte)*

Betrachten Sie den Graphen in Abbildung 2. Jeder Knoten ist hier ein Gebäude. Jede Kante ist ein Kabel, das sie zwischen den Gebäuden verlegen könnten, die sie verbindet. Das (immer positive) Kantengewicht entspricht den Kosten des Kabels.

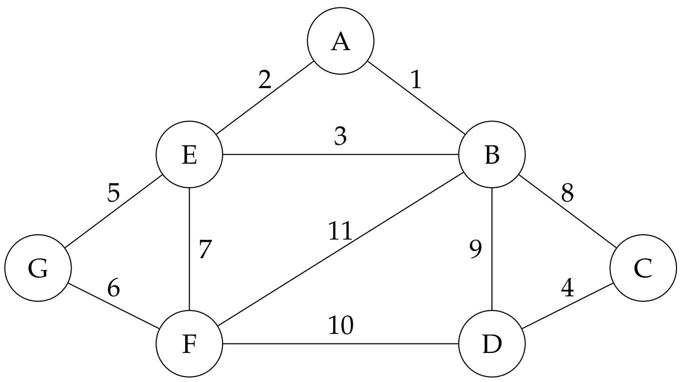


Abbildung 2: Gebäude und mögliche Kabel

Ihre Aufgabe ist es, eine Menge von Kabeln so auszuwählen, dass alle Gebäude (nicht notwendig *direkt*) über Kabel miteinander verbunden sind. Dabei sollen Sie die Kosten so gering wie möglich halten.

a) Geben Sie für den Graphen in Abbildung 2 eine *minimale* Menge K von Kabeln an, die alle Gebäude miteinander verbindet. Die Menge ist *minimal*, wenn sie kein Kabel aus ihr entfernen könnten, ohne die wechselseitige Verbindung aller Gebäude zu zerstören. (*1 Punkt*)

Hinweis: Sie müssen hier nur eine minimale *Menge finden*, nicht *die Menge mit den geringsten Kosten.*

Lösung:
Es gibt viele *minimale* Mengen. Zwei Beispiele:
$K = \{\{A,B\},\{A,E\},\{C,D\},\{E,G\},\{F,G\},\{B,C\}\}$ ist die kostengünstigste Lösung, siehe noch weiter unten.

$K = \{\{A,B\},\{A,E\},\{B,F\},\{F,D\},\{F,G\},\{D,C\}\}$ ist eine Alternative (mit deutlich höheren Kosten).

b) Jeweils mit Begründung: Wie viele Kabel müssen Sie im vorliegenden Fall *mindestens* auswählen? Wie viele Kabel müssen Sie *mindestens* auswählen, wenn Sie n Gebäude miteinander verbinden wollen? (*1 Punkt*)

Lösung:
Der Graph hat 7 Knoten, daher müssen wir mindestens 6 Kabel auswählen, um alle Knoten miteinander zu verbinden. Für einen Graphen mit n Knoten müssen wir aus demselben Grund $n-1$ Kabel auswählen.

c) Jeweils mit Begründung: Wie viele Kabel dürfen Sie im vorliegenden Fall *höchstens* auswählen? Wie viele Kabel dürfen Sie *höchstens* auswählen, wenn Sie n Gebäude miteinander verbinden wollen? (*1 Punkt*)

Lösung:
Im vorliegenden Fall reichen 6 Kabel, um alle Knoten miteinander zu verbinden. Da die Kabel strikt positive Kosten haben, kann eine zulässige Lösung mit minimalen Kosten kein zusätzliches Kabel enthalten. Im allgemeinen Fall reichen immer $n-1$ Kabel, um n Knoten miteinander zu verbinden. In jeder Lösung mit mehr als $n-1$ Kabeln könnte man daher mindestens ein Kabel weglassen, sodass eine solche Lösung nicht die kostengünstigste sein kann. Daher darf man, um n Gebäude möglichst günstig miteinander zu verbinden, höchstens $n-1$ Kabel auswählen.

d) Betrachten Sie Algorithmus 2.

Algorithmus 2 : Kabelwahl

```
1 G ← Eingabe (Graph);
2 E ← Kanten von G;
3 geprueft ← {};                    // Menge von Kanten der Form {knoten1,knoten2}
4 ausgewaehlt ← {};                 // Menge von Kanten der Form {knoten1,knoten2}
5 while geprueft ≠ E do
6 |   {u,v} ← Kante mit dem niedrigsten Gewicht, die noch nicht in geprueft ist;
7 |   ausgewaehlt ← ausgewaehlt ∪ {{u,v}};  // "∪" bildet die Vereinigung von Mengen
8 |   geprueft ← geprueft ∪ {{u,v}};
```

(1) Argumentieren Sie, dass der Algorithmus Ihre Aufgabe *nicht* korrekt löst. (*1 Punkt*)

Lösung:
Der Algorithmus fügt *alle* Kanten zu `ausgewaehlt` hinzu, unabhängig davon, ob sie zur Verbindung der Gebäude benötigt werden.

(2) Um den Algorithmus zu korrigieren, müssen Sie lediglich eine Bedingung (mit *if*) einfügen. Wie muss diese Bedingung formuliert sein und an welcher Stelle müssen Sie diese einfügen (mit Begründung)? (*2 Punkte*)

Hinweis: Sie dürfen annehmen, dass es eine Funktion erreichbar(u,v,K) *gibt, die für Knoten u, v und eine Kantenmenge K die Frage beantwortet, ob u von v über Kanten aus K erreichbar ist.*

Lösung:
Die Bedingung muss nach Zeile 6 hinzugefügt werden, sodass Zeile 7 (und nur diese Zeile) nur ausgeführt wird, wenn die Bedingung erfüllt ist. Sie muss sicherstellen, dass nur Kanten ausgewählt werden, die zur Verbindung der Gebäude tatsächlich benötigt werden. Eine Kante $\{u,v\}$ wird *nicht* benötigt, wenn u bereits über ausgewählte Kanten von v erreichbar ist. Die Bedingung muss daher lauten:
if ¬erreichbar(u, v, `ausgewaehlt`)

(3) Argumentieren Sie, dass der Algorithmus mit der von Ihnen vorgeschlagenen Korrektur Ihre Aufgabe nun *korrekt* löst. (*2 Punkte*)

Lösung:
Die Kanten werden in aufsteigender Reihenfolge ihres Gewichts geprüft und immer dann hinzugefügt, wenn die Knoten, die sie verbinden, noch nicht über andere ausgewählte Kanten voneinander erreichbar sind. Dadurch werden für einen Graphen mit n Knoten genau $n-1$ Kanten ausgewählt, und zwar die $n-1$ Kanten mit dem kleinsten Gewicht, die nicht die schwersten Kanten in einem Kreis (das ist ein Pfad von v nach v) sind.

(4) Benutzen Sie den korrigierten Algorithmus, um für den Graphen in Abbildung 2 eine *kostengünstigste* Menge von Kabeln zu bestimmen, die all Gebäude miteinander verbindet. Geben Sie die Kabel in der Reihenfolge an, in der sie ausgewählt werden, und geben Sie außerdem die Gesamtkosten Ihrer Lösung an. (*2 Punkte*)

Lösung:
$K = \{\{A, B\}, \{A, E\}, \{C, D\}, \{E, G\}, \{F, G\}, \{B, C\}\}$. Die Kosten sind $1 + 2 + 4 + 5 + 6 + 8 = 26$.

Aufgabe 3 Sendefrequenzen *(10+4 Punkte)*

Sie betreiben ein Mobilfunknetz mit mehreren Antennen. Jede Antenne hat einen Senderadius, wobei sich die Radii einiger Antennen überlappen, um einen zuverlässigen Mobilfunkempfang sicherzustellen.

Sie wollen nun jeder Antenne eine *Sendefrequenz* zuordnen. Um *Interferenz* zu vermeiden, müssen Sie dabei sicherstellen, dass Antennen mit überlappenden Senderadii auf *unterschiedlichen* Frequenzen senden.

Nehmen Sie an, dass die verschiedenen Sendefrequenzen durchnummeriert sind, angefangen bei 1, und dass die Antennen *Unique Identifiers* haben, die sich alphabetisch ordnen lassen.

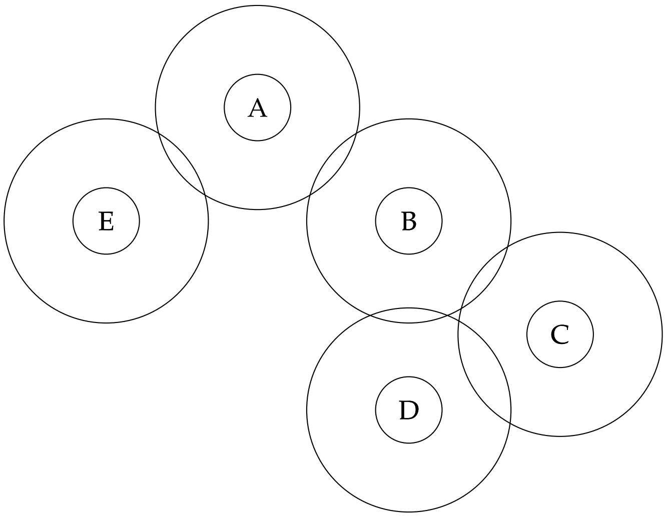
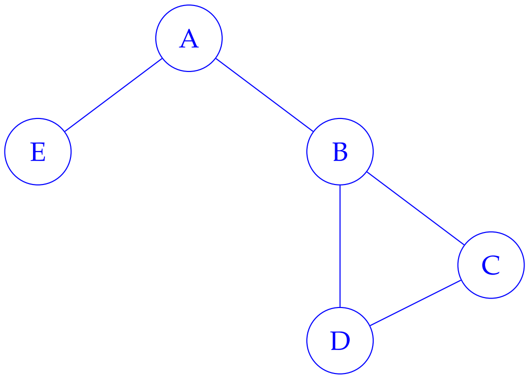


Abbildung 3: Antennen und ihre Senderadii

a) Abbildung 3 zeigt die Antennen mit ihren jeweiligen Senderadii. Stellen Sie diese Situation nach der folgenden Regel als Graph dar: *Jede Antenne ist ein Knoten, und eine Kante zwischen zwei Knoten gibt an, dass die Senderadii zwischen den beteiligten Antennen überlappen.* (*2 Punkte*)

Lösung:



b) Die Menge der Knoten, mit denen ein Knoten v verbunden ist, heißt *Nachbarn* von v, notiert $\Gamma(v)$ (Γ ist das große griechische *Gamma*). Die Anzahl der Nachbarn von v heißt *Grad* von v, notiert $\delta(v) = |\Gamma(v)|$ (δ ist das kleine griechische *Delta*), und der größte Grad eines Knoten in einem Graphen ist $\Delta = \max_{v \in V} \delta(v)$ (dabei ist V die Menge der Knoten des Graphen; Δ ist das große griechische *Delta*). Wer sind $\Gamma(A)$ und $\delta(A)$ in Ihrer Lösung zur vorigen Teilaufgabe, und was ist Δ? (2 *Punkte*)

Lösung:
$\Gamma(A) = \{E, B\}$, $\delta(A) = 2$ und $\Delta = \delta(B) = 3$, siehe Zeichnung zur vorigen Teilaufgabe.

c) Mit Begründung: Wie viele verschiedene Sendefrequenzen müssen Sie *mindestens* zuweisen, um sicherzustellen, dass benachbarte Antennen unterschiedliche Frequenzen erhalten? (1 *Punkt*)

Lösung:
3, denn die Senderadii von B, C und D überlappen sich jeweils.

d) Betrachten Sie Algorithmus 3.

Algorithmus 3 : Zuweisung von Sendefrequenzen

$G \leftarrow$ Eingabe (Graph);
$V \leftarrow$ Knoten von G;
`behandelt` $\leftarrow \{\}$; `// Menge von Knoten`
`frequenzen` $\leftarrow \{\}$; `// Menge von Tupeln der Form (knoten, frequenz)`
while `behandelt` $\neq V$ **do**
  $v \leftarrow$ Knoten mit dem größten Grad, der noch nicht in `behandelt` ist (bei Gleichstand: alphabetisch);
  `nachbarn` $\leftarrow \Gamma(v)$;
  `nachbarfrequenzen` $\leftarrow \{i \mid x \in$ `nachbarn` $\wedge (x, i) \in$ `frequenzen`$\}$;
  $f \leftarrow \min\{i \in \mathbb{N}_{>0} \mid i \notin$ `sendefrequenzen`$\}$; `//` $\mathbb{N}_{>0} = \{1, 2, 3, \dots\}$`, natürliche Zahlen` > 0
  `frequenzen` $\leftarrow$ `frequenzen` $\cup \{(v, f)\}$; `// "∪" bildet die Vereinigung von Mengen`
  `behandelt` $\leftarrow$ `behandelt` $\cup \{v\}$;

(1) Führen Sie den Algorithmus auf dem Graphen in Abbildung 3 aus und geben Sie an, welche Frequenzen welchen Knoten zugeordnet werden. Behalten Sie dabei die Reihenfolge der Zuordnung bei. (2 *Punkte*)

Lösung:
`frequenzen` $= \{(B, 1), (A, 2), (C, 2), (D, 3), (E, 1)\}$.

(2) Mit Begründung: Wie viele verschiedene Frequenzen verwendet der Algorithmus *höchstens* für einen Graphen mit größtem Grad x (d.h. mit $\Delta = x$)? (3 *Punkte*)

Lösung:
Der Algorithmus verwendet höchstens $x + 1$ verschiedene Frequenzen. Sei v ein beliebiger Knoten. Da v höchstens x Nachbarn hat, können höchstens x verschiedene Frequenzen bereits an die Nachbarn vergeben sein, wenn v betrachtet wird (es können weniger sein, da Nachbarn die gleiche Frequenz haben können bzw. noch gar keine Frequenz haben können). Also ist eine der kleinsten $x + 1$ Frequenzen für v frei.

e) (Bonus) Betrachten Sie nun die Antennen in Abbildung 4 mit ihren jeweiligen Senderadii.

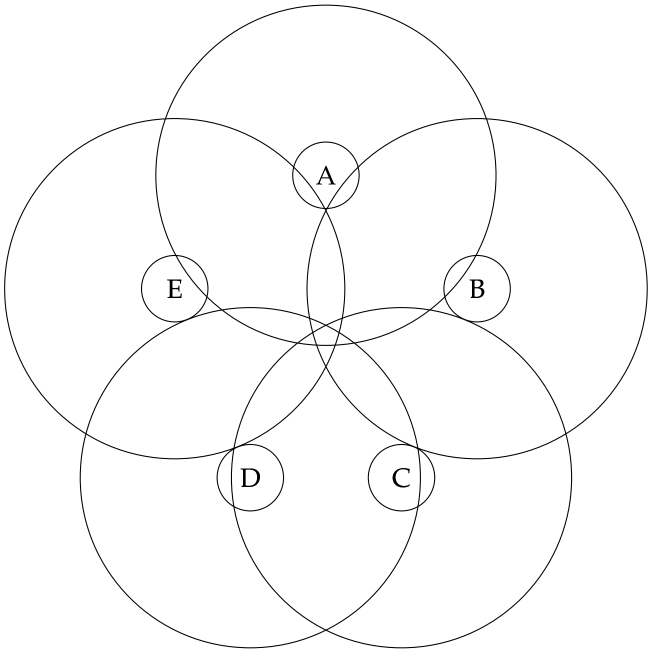


Abbildung 4: Seltsame Antennensituation

(1) Mit Begründung: Wie viele Frequenzen müssen Sie hier *mindestens* zuweisen? Wie häufig kommt jede einzelne Frequenz *höchstens* vor? (*2 Punkte*)

Lösung:
In der Graphdarstellung des Problems ist jeder Knoten mit allen anderen Knoten verbunden. Daher muss jeder Knoten seine eigene Frequenz erhalten. Da es 5 Knoten gibt, gibt es also genau 5 verschiedene Frequenzen, und jede Frequenz kommt genau einmal vor.

(2) Sei c Ihre Antwort auf die erste Frage in der vorigen Teilaufgabe. Wie viele verschiedene Möglichkeiten gibt es, den Antennen in Abbildung 4 die Frequenzen 1 bis c zuzuweisen?
(*2 Punkte*)

Lösung:
Es gibt c Möglichkeiten, für Antenne A eine Frequenz zu wählen, $c - 1$ Möglichkeiten, dies für Antenne B zu tun, usw. Also gibt es insgesamt $c!$ verschiedene zulässige Zuweisungen von Frequenzen zu c Antennen, die jeweils paarweise miteinander verbunden sind.

Ich habe für die Videos, die Nachbereitung und das Übungsblatt etwa ☐ Stunden gebraucht.
(Ann-Sophie fertigt aus diesen Zahlen eine Statistik an. Kurt und Corinna sehen nur diese Statistik. Wir möchten wissen, ob der Schwierigkeitsgrad in etwa richtig ist.)

Internet	war	spannend ☐	okay ☐	langweilig ☐
		schwierig ☐	okay ☐	einfach ☐