Technische Universität Braunschweig

Algorithmen und Datenstrukturen
Übung 1

Arne Schmidt

14.11.2019

Organisation

Homepage

Aktuelle Informationen, Hausaufgaben, Slides auf:
https://www.ibr.cs.tu-bs.de/courses/ws1920/aud/webpages/index.html

Semesterplan AuD WS19/20

Woche (KW)	Woche (Datum)	Vorlesung (Di.+Mi.)	Gr. Übung (Do.)	Kl. Übung (Mo.-Fr.)	HA Ausgabe (Mo.)	HA Abgabe (Mo. bis 10:00 Uhr)	HA Rückgabe (in kl. Übung)
43	21.10.	-,-					
44	28.10.	0,1	Reformationstag				
45	04.11.	2,3			HA0a*,HA1		
46	11.11.	4,5	1	1			HA0a
47	18.11.	6,7			HA2	HA1	
48	25.11.	8,9	2	2			HA1
49	02.12.	10,11			HA3	HA2	
50	09.12.	12,13	3	3			HA2
51	16.12.	14,15			HA4	HA3	
52	23.12.	Weihnachtsferien					
1	30.12.						
2	06.01.	16,17	4	4			HA3
3	13.01.	18,19			HA5	HA4	
4	20.01.	20,21	5	5			HA4
5	27.01.	22,23				HA5	
6	03.02.	24,25	6	6			HA5
*: Mündliche Aufgaben, Bearbeitung in ersten kleinen Übungen; keine Bewertung							

Anmeldung zu den kleinen Übungen: **Mo., 28.10.18 (0:00 Uhr) - Mi., 06.11.18 (20:00 Uhr)**

Stundenplan

Semesterplan AuD WS19/20

Woche (KW)	Woche (Datum)	Vorlesung (Di.+Mi.)	Gr. Übung (Do.)	Kl. Übung (Mo.-Fr.)	HA Ausgabe (Mo.)	HA Abgabe (Mo. bis 10:00 Uhr)	HA Rückgabe (in kl. Übung)
43	21.10.	-,-					
44	28.10.	0,1	Reformationstag				
45	04.11.	2,3			HA0a*,HA1		
46	11.11.	4,5	1	1			HA0a
47	18.11.	6,7			HA2	HA1	
48	25.11.	8,9	2	2			HA1
49	02.12.	10,11			HA3	HA2	
50	09.12.	12,13	3	3			HA2
51	16.12.	14,15			HA4	HA3	
52	23.12.	Weihnachtsferien					
1	30.12.						
2	06.01.	16,17	4	4			HA3
3	13.01.	18,19			HA5	HA4	
4	20.01.	20,21	5	5			HA4
5	27.01.	22,23				HA5	
6	03.02.	24,25	6	6			HA5

*: Mündliche Aufgaben, Bearbeitung in ersten kleinen Übungen; keine Bewertung

Große Übungen

- Aufarbeitung des Inhalts aus der Vorlesung
- Weiterführende Inhalte
- Beantwortung von Fragen
- Interaktion!

Semesterplan AuD WS19/20

Woche (KW)	Woche (Datum)	Vorlesung (Di.+Mi.)	Gr. Übung (Do.)	Kl. Übung (Mo.-Fr.)	HA Ausgabe (Mo.)	HA Abgabe (Mo. bis 10:00 Uhr)	HA Rückgabe (in kl. Übung)
43	21.10.	-,-					
44	28.10.	0,1	Reformationstag				
45	04.11.	2,3			HA0a*,HA1		
46	11.11.	4,5	1	1			HA0a
47	18.11.	6,7			HA2	HA1	
48	25.11.	8,9	2	2			HA1
49	02.12.	10,11			HA3	HA2	
50	09.12.	12,13	3	3			HA2
51	16.12.	14,15			HA4	HA3	
52	23.12.	Weihnachtsferien					
1	30.12.						
2	06.01.	16,17	4	4			HA3
3	13.01.	18,19			HA5	HA4	
4	20.01.	20,21	5	5			HA4
5	27.01.	22,23				HA5	
6	03.02.	24,25	6	6			HA5

*: Mündliche Aufgaben, Bearbeitung in ersten kleinen Übungen; keine Bewertung

Anmeldung zu den kleinen Übungen: **Mo., 28.10.18 (0:00 Uhr) - Mi., 06.11.18 (20:00 Uhr)**

Fragen und/oder Wünsche einfach per Mail an mich.

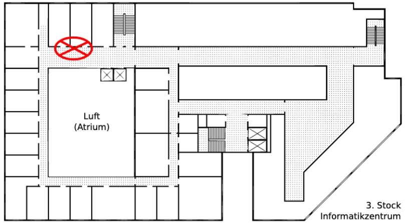
Hausaufgaben

Algorithmen und Datenstrukturen
Prof. Dr. Sándor P. Fekete
Arne Schmidt

Winter 2019/20
Abgabe: 18.11.2019
Rückgabe: 25.–29.11.2019

Übungsblatt 1

Abgabe der Lösungen bis zum 18.11.2019 um 10:00 Uhr im Hausaufgabenschrank bei Raum IZ 337 (siehe Skizze rechts). Es werden nur mit einem dokumentenechten Stift (kein Rot!) geschriebene Lösungen gewertet. **Bitte die Blätter zusammenheften und vorne deutlich mit eigenem Namen, Matrikel- und Gruppennummer, sowie Studiengang versehen!**



- 6 Blätter
  - 1 unbewertet (Blatt 0a)
  - 5 bewertet (Blätter 1 bis 5)
- Je 20 Punkte ➔ Insgesamt 100 Punkte
- Studienleistung: 50% aller Punkte, also 50 Punkte
  - Studienleistung ist **keine** Voraussetzung, um an der Prüfung teilzunehmen.
  - Studienleistung ist **eine** Voraussetzung, um das Modul abzuschließen.
  - Studienleistung ist nicht benotet und fließt nicht in die Prüfung ein.

Hausaufgaben

- Zu späte Abgaben: 0 Punkte
- Falscher Abgabeschrank: 0 Punkte
- Mit Bleistift oder Rot geschriebene Teile werden nicht gewertet.
- Zusammen überlegen, **ABER:** einzeln aufschreiben und abgeben.

Wichtig: Die Hausaufgaben dienen Euch (und nicht uns)
zur Vorbereitung für die Klausur. Abschreiben bringt nichts!

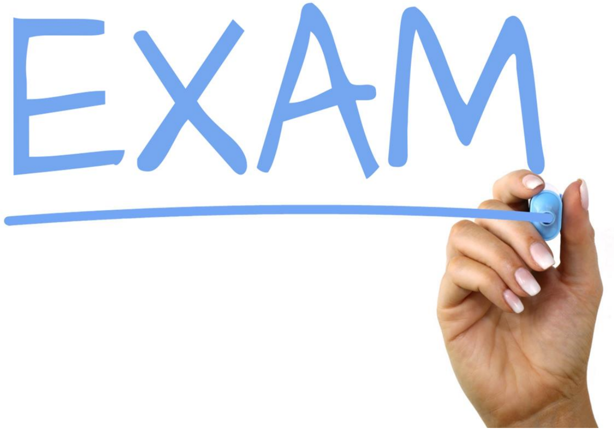
Hausaufgaben

Hausaufgabenschrank

Klausur

Findet statt am 28.02.2020, zwischen 17:00 Uhr und 20:00 Uhr.
Nähere Informationen wie
- Raumaufteilung
- Beginn der Klausur

werden wenige Tage vorher bekanntgegeben



Technische Universität Braunschweig Wintersemester 2019/2020
Institut für Betriebssysteme und Rechnerverbund
Abteilung Algorithmik

Prof. Dr. Sándor P. Fekete
Arne Schmidt

Klausur
Algorithmen und Datenstrukturen
28.02.2020

Name: ....

Vorname: ....

Matr.-Nr.: ....

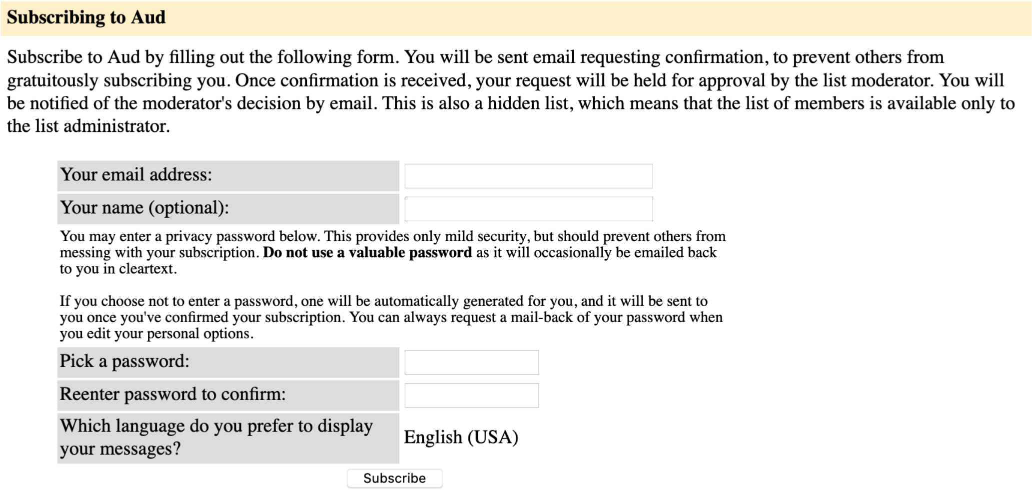
Studiengang: ....

☐ Bachelor ☐ Master ☐ Andere

Klausurcode:
Dieser wird benötigt, um das Ergebnis der Klausur abzurufen.

Mailingliste

- Anmeldung über Homepage
- Für Informationen wie
    - Raumänderungen,
    - Ausfälle,
    - Etc.
- Möglichkeit für Fragen
    - Diskussion mit anderen Teilnehmern



Subscribing to Aud

Subscribe to Aud by filling out the following form. You will be sent email requesting confirmation, to prevent others from gratuitously subscribing you. Once confirmation is received, your request will be held for approval by the list moderator. You will be notified of the moderator's decision by email. This is also a hidden list, which means that the list of members is available only to the list administrator.

Your email address:	
Your name (optional):	

You may enter a privacy password below. This provides only mild security, but should prevent others from messing with your subscription. **Do not use a valuable password** as it will occasionally be emailed back to you in cleartext.

If you choose not to enter a password, one will be automatically generated for you, and it will be sent to you once you've confirmed your subscription. You can always request a mail-back of your password when you edit your personal options.

Pick a password:	
Reenter password to confirm:	
Which language do you prefer to display your messages?	English (USA)

Subscribe

Fragen

Stellt Eure Fragen

- Über Mailingliste
- Euren Tutoren
- Sprechstunde von mir (Mo. 9:45 – 10:30)
- Per Mail an mich (aschmidt@ibr.cs.tu-bs.de)
- Sprechstunde von Prof. Fekete (Mi. 13:15 – 14:00)

Fragen?


And Now For Something
Completely Different

Beweise

https://www.ibr.cs.tu-bs.de/courses/ws1920/aud/uebungen/Merkzettel-Beweise.pdf

Mathematische Aussagen

Mathematische Aussagen sind entweder wahr oder falsch.

1. 13 ist eine gerade Zahl.
2. Die Summe der ersten n natürlichen Zahlen ist $\frac{n(n+1)}{2}$.
3. Die größte Primzahl ist $2^{82\,589\,933} - 1$
4. Zusammenhängende, einfache Graphen mit $n \geq 2$ Knoten und $n-1$ Kanten besitzen mindestens zwei Knoten vom Grad 1.

Mathematische Aussagen

Mathematische Aussagen sind entweder wahr oder falsch.

1. $\boxed{\text{Es gibt}}$ eine Zahl $k \in \mathbb{N}$, sodass $2k = 13$
2. $\boxed{\text{Für jedes}}$ $n \in \mathbb{N}$ gilt: $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$.
3. $\boxed{\text{Für jede}}$ Primzahl p gilt: $\mathrm{p} \leq 2^{82\,589\,933} - 1$
4. $\boxed{\text{Für alle}}$ zusammenhängende, einfache Graphen mit $n \geq 2$ Knoten und $n - 1$ Kanten gilt: es gibt mindestens zwei Knoten vom Grad 1.

Existenz- vs. Allaussagen

	Existenzaussage	Allaussage
Zeigen	Beispiel	Beweis
Widerlegen	Beweis	Beispiel

Negation einer Existanzaussage wird zu einer Allaussage.
Negation einer Allaussage wird zu einer Existenzaussage.

Logische Verknüfungen

Negation („Nicht“, $\neg$)

Konjunktion („Und“, $\wedge$)

Disjunktion („Oder“, $\vee$)

Implikation („wenn…dann“, $\Rightarrow$)

Äquivalenz („genau dann wenn“, $\Leftrightarrow$)

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \Rightarrow B$	$A \Leftrightarrow B$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

Beweise – Direkter Beweis

Aussagen oft in der Form $A \Rightarrow B$. Unterscheide zwischen **Voraussetzung** (A) und **Schlussfolgerung** (B)

Wird die Schlussfolgerung durch eine logische Folgerungskette aus den Voraussetzungen hergeleitet, so spricht man von einem **direkten Beweis**.

Beispiel:

Wenn $x, y \in \mathbb{R}$ und $x, y \geq 0$, dann gilt $\sqrt{xy} \leq \frac{x+y}{2}$

Beweis:

$$\begin{aligned} x, y \in \mathbb{R} \text{ und } x, y \geq 0 &\Rightarrow (x-y)^2 = x^2 - 2xy + y^2 \geq 0 \\ &\Rightarrow x^2 + 2xy + y^2 \geq 4xy \\ &\Rightarrow \frac{(x+y)^2}{4} \geq xy \Rightarrow \frac{x+y}{2} \geq \sqrt{xy} \end{aligned}$$

Monotonie der Wurzelfunktion

Beweise – Kontraposition

Ein direkter Beweis kann schwierig sein, sodass sich die **Kontraposition** anbietet.
Es gilt

$$(A \Rightarrow B) \Leftrightarrow (\neg B \Rightarrow \neg A)$$

Wir können also Annehmen, dass B nicht gilt und folgern daraus, dass auch A nicht gelten kann.

Beispiel: Sei $n \in \mathbb{Z}$. „Wenn n^2 ungerade ist, ist n ungerade“ wird zu „Wenn n gerade ist, ist n^2 gerade.“

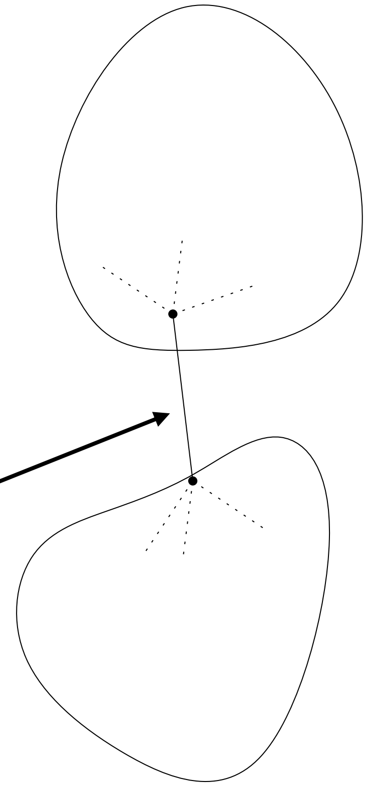
Beweis: n gerade $\Rightarrow$ Es ex. $k \in \mathbb{Z}$ mit $2k = n$
$\Rightarrow n^2 = 4k^2 = 2(2k^2)$, also eine gerade Zahl

Beweise – Widerspruch

Mathematische Aussagen sind entweder wahr oder falsch. Um eine wahre Aussage zu beweisen, können wir zeigen, dass die Negation nicht gilt.
Diese Beweistechnik nennt man **Widerspruchsbeweis**.

Bei einer Implikation $(A \Rightarrow B)$ zeigt man also, dass A und $\neg B$ nicht gleichzeitig gelten können.

Beispiel: Besitzt ein zusammenhängender Graph G eine ***Brücke***, so besitzt G Knoten mit ungeradem Grad.



Beweise – Widerspruch

Beispiel: Besitzt ein zusammenhängender Graph G eine ***Brücke***, so besitzt G Knoten mit ungeradem Grad.

Beweis: Angenommen, G besitzt nur Knoten geraden Grades.
Dann existiert in G eine Eulertour.

Da e auf der Eulertour liegt und sein Entfernen G in zwei Komponenten teilt, können wir eine Komponente verlassen, aber nicht dorthin zurückkehren. Wir können also keine Eulertour konstruieren (ansonsten wäre e keine Brücke).

Also muss G mindestens ein Knoten mit ungeradem Grad besitzen.

Beweise – Äquivalenzen

Aussagen der Form $A \Leftrightarrow B$ werden bewiesen, indem sowohl $A \Rightarrow B$ **und** $B \Rightarrow A$ gezeigt werden

Um eine solche Aussage zu widerlegen, reicht es $A \Rightarrow B$ **oder** $B \Rightarrow A$ zu widerlegen.

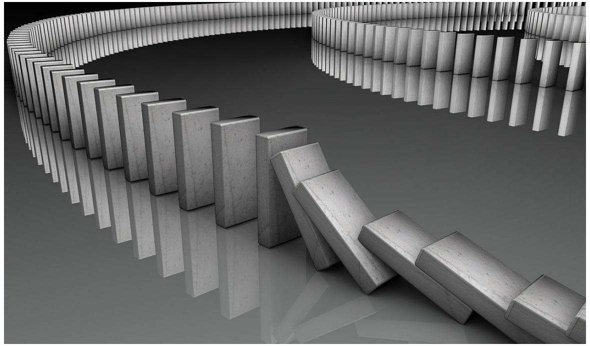
Beweise – Vollständige Induktion

Beweise – Vollständige Induktion

Bei der **vollständigen Induktion** beweist man Aussagen der Form
„Für alle $n \in \mathbb{N}, n \geq n_0$ gilt $P(n)$“ indem man

1. $P(n_0)$ und
2. „Für alle $x \in \mathbb{N}, x \geq n_0$ gilt $P(x) \Rightarrow P(x+1)$“

zeigt.



Vollständige Induktion lässt sich auch oft auf Graphenstrukturen anwenden!

Beweise – Vollständige Induktion

Zeige: Für alle $n \in \mathbb{N}$ gilt $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$.

Induktionanfang:

Für $n = 1$ ist $\sum_{i=1}^{n} i = 1 = \frac{2}{2} = \frac{1\cdot(1+1)}{2}$. Diese Aussage stimmt also.

Induktionsvoraussetzung:

Es gelte $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ mit n beliebig, aber fest.

Induktionsschritt: (n → n+1)

$$\sum_{i=1}^{n+1} i = (n+1) + \sum_{i=1}^{n} i$$

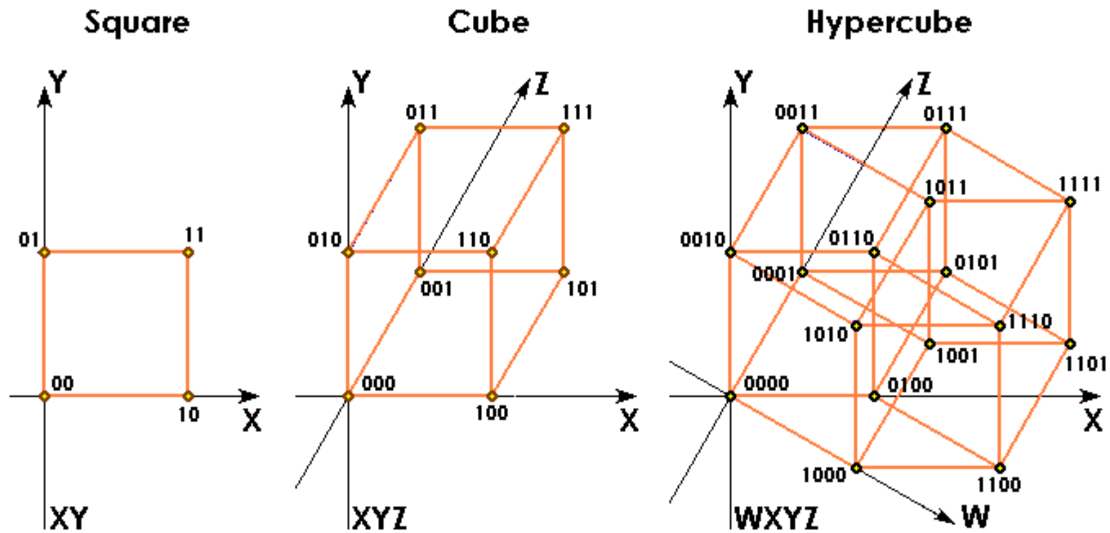
$$\overset{IV}{\triangleq} (n+1) + \frac{n(n+1)}{2}$$

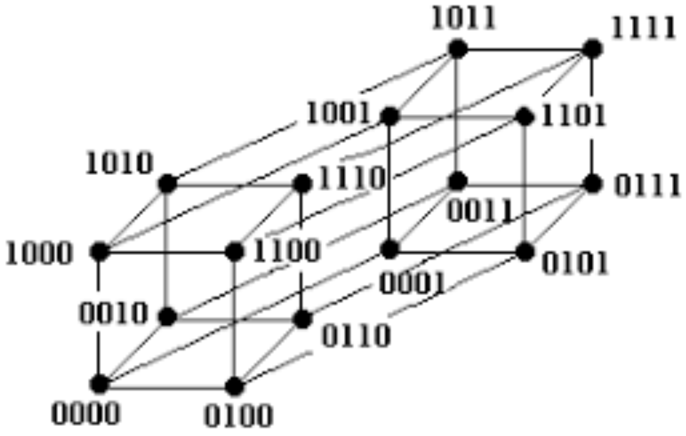
$$= \frac{2(n+1) + n(n+1)}{2} = \frac{(n+1)(n+2)}{2}$$

Beweise – Vollständige Induktion

n-Dimensionale Würfel

Zeige: n-Dimensionale Würfel sind hamiltonsch.





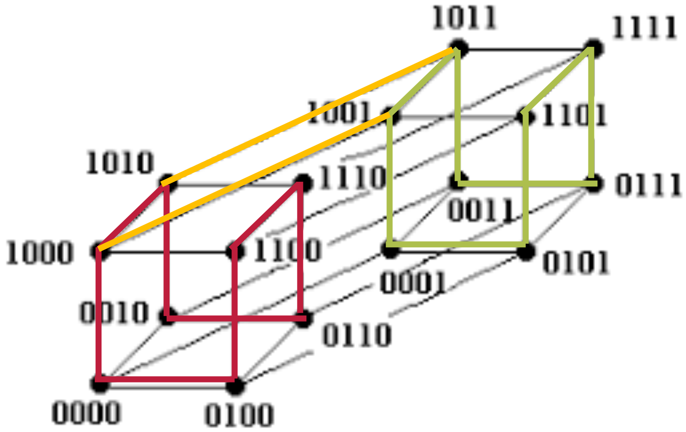
Beweise – Vollständige Induktion

I.A.: Der 2-Dimensionale Würfel ist hamiltonsch (der Kreis selbst ist die Tour).

I.V.: Der n-Dimensionale Würfel ist hamiltonsch für bel. aber festes n.

I.S.:

- Betrachte $(n+1)$-Dimensionalen Würfel.
- Nach IV: Zwei Subgraphen hamiltonsch.
- Verschmelze die Kreise.



Beweise – Vollständige Induktion

I.A.: Der 2-Dimensionale Würfel ist hamiltonsch (der Kreis selbst ist die Tour).

I.V.: Der n-Dimensionale Würfel ist hamiltonsch für bel. aber festes n.

I.S.: Betrachte einen $(n+1)$-dimenisonalen Würfel W und die Aufteilung in zwei n-dimensionale Würfel $W_1 \coloneqq \{v_1, \dots, v_{2^n}\}, W_2 \coloneqq \{v_1', \dots, v_{2^n}'\}$, wobei v_i und v_i' in W verbunden sind.

W_1 und W_2 sind beide hamiltonsch. Betrachte einen Hamiltonkreis, der in beiden Würfeln die gleiche Reihenfolge benutzt. O.B.d.A. sei $\mathrm{v}_1\mathrm{v}_2$ die letzte Kante des Kreises K_1 in W_1 und $\mathrm{v}_1'\mathrm{v}_2'$ die letzte Kante des Kreises K_2 in W_2.

Ein Hamiltonkreis in W ist: $\mathrm{v}_1 \overbrace{\mathrm{v}_1' \dots \mathrm{v}_2'}^{K_2} \overbrace{\mathrm{v}_2 \dots \mathrm{v}_1}^{K_1}$

Beweise – Vollständige Induktion

Beliebte Fehler:

- Build-Up-Error:

  Füge etwas hinzu, um den Induktionsschritt zu zeigen.
  So werden ggf. nicht alle Strukturen berücksichtigt!
- Kein Induktionsanfang oder -schluss.

  IA und IS müssen beiden vorhanden sein!
- Zu wenig Induktionsanfänge
- Induktionsvoraussetzung falsch angewendet.

  Behauptung: $a^n = b^n$ für alle $a, b \in \mathbb{R}, n \geq 0$
  IA: $n = 0: a^0 = 1 = b^0$
  IV: $a^x = b^x$ gilt für alle $x \leq n$ mit n bel., aber fest.
  IS: $a^{n+1} = a^n a \neq b^n b = b^{n+1}$

Beispiele für Beweise

„Handshake-Lemma“

Satz 2.5: Für jeden beliebigen einfachen Graphen ist die Zahl der Knoten mit ungeradem Grad gerade.

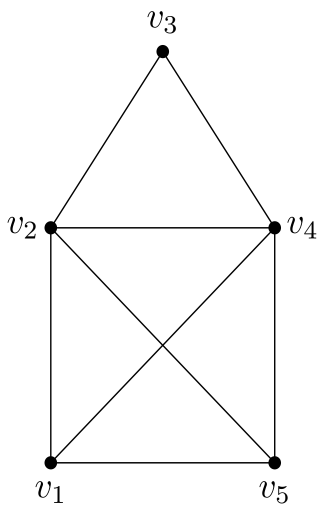
Beweis:
Betrachte Summe der Knotengrade

$$\sum_{i=1}^{n} \delta(v_i)$$

Jede Kante wird doppelt betrachtet, also

$$\sum_{i=1}^{n} \delta(v_i) = 2m$$

Das ist eine gerade Zahl!
Es kann also nur gerade viele Knoten ungeraden Grades geben.
Gäbe es ungerade viele ungerade Grade, wäre auch die Gradsumme ungerade



$$\sum_{i=1}^{n} \delta(v_i) = 3 + 4 + 2 + 4 + 3 = 16$$

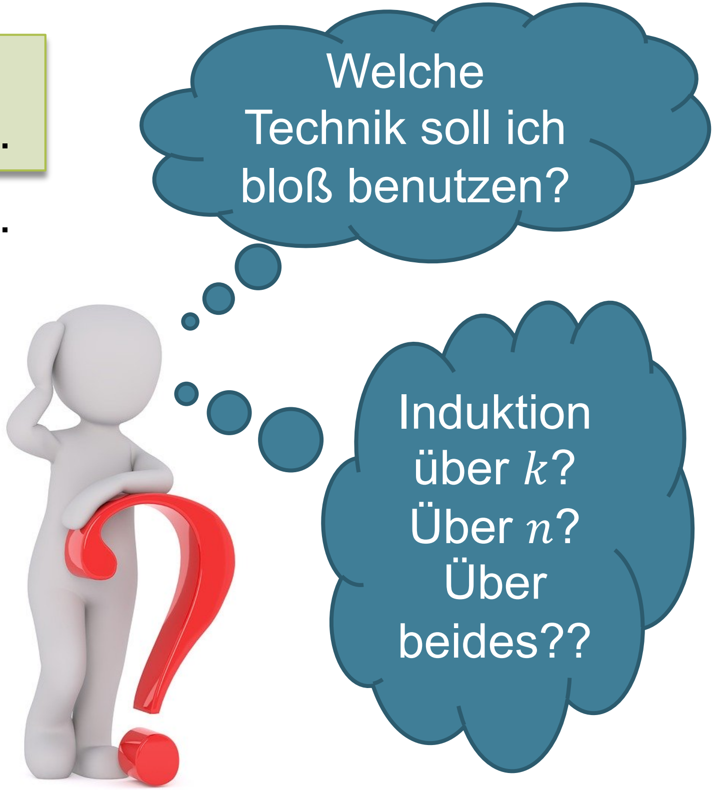
Kreisfreie Graphen

Satz: Jeder kreisfreie Graph mit n Knoten und k Zusammenhangskomponenten besitzt $n-k$ Kanten.

Beweis per Induktion über k mit n fest, aber beliebig.
IA: Für n Komponenten besitzt der Graph keine Kanten, also: $0 = n - n = n - k$ Kanten.

IV: Jeder kreisfreie Graph mit k ZHKs besitzt $n - k$ Kanten.

IS: $k \to k-1$
Sei G ein Graph mit $k-1$ ZHKs und sei $e \in E$.
G ohne e besitzt k ZHKs, also $n-k$ Kanten.
$\Rightarrow G$ besitzt $n-(k-1)$ Kanten.



Kreisfreie Graphen

Satz: Jeder zusammenhänge kreisfreie Graph besitzt mindestens zwei Knoten mit Grad 1.

Beweis per Widerspruch.
Annahme: Es gibt maximal einen Knoten von Grad 1.
Dann ist:

$$2m = \sum_{i=1}^{n} \delta(v_i) \geq 1 + 2(n-1) > 2(n-1)$$

Graph hat also mindestens n Kanten.
$\Rightarrow$ Graph ist nicht kreisfrei.