ETH Zürich FS 2023
Institute of Theoretical Computer Science
Prof. Rasmus Kyng, Prof. Angelika Steger, Prof. Emo Welzl
Marc Kaufmann, Ulysse Schaller

Algorithmen und Wahrscheinlichkeit
Warmup-Exercises — Lösung

Aufgabe 1 – *Pfade, Wege, Kreise*

1. Der Pfad $\langle a, b, c, f, e\rangle$ ist der einzige Pfad der Länge 4 von a nach e.
2. Es gibt 12 solche Wege: $\langle a, b, c, f, e\rangle$, $\langle a, b, c, b, e\rangle$, $\langle a, b, e, d, e\rangle$, $\langle a, b, e, f, e\rangle$, $\langle a, b, e, b, e\rangle$, $\langle a, b, a, b, e\rangle$, $\langle a, b, a, d, e\rangle$, $\langle a, d, a, d, e\rangle$, $\langle a, d, a, b, e\rangle$, $\langle a, d, e, d, e\rangle$, $\langle a, d, e, b, e\rangle$, $\langle a, d, e, f, e\rangle$.
3. Es gibt drei wesentlich verschiedenen Kreise $\langle a, b, d, e, a\rangle$, $\langle b, c, f, e, b\rangle$ und $\langle a, b, c, f, e, d, a\rangle$. Ausserdem können wir für jeden Kreis noch den Startpunkt auswählen (vier Möglichkeiten in den ersten zwei Fällen, 6 Möglichkeiten im dritten Fall), und die Durchlaufrichtung. Insgesamt gibt es also $8 + 8 + 12 = 28$ verschiedene Kreise. In der Vorlesung werden wir oft Startpunkt und Durchlaufrichtung ignorieren – daher werden wir im Einzelfall spezifizieren, ob sie berücksichtigt werden sollen.
4. Es gibt unendlich viele Zykeln in G, z.B. $\langle a, b, a\rangle$, $\langle a, b, a, b, a\rangle$, $\langle a, b, a, b, a, b, a\rangle, \ldots$.

Aufgabe 2 – *Asymptotisches Wachstum.*

Zur besseren Lesbarkeit kürzen wir $f = o(g)$ durch $f \ll g$ und $f = \Theta(g)$ durch $f \equiv g$ ab.

$$2^{32} \ll \log(n^2) \equiv \ln n \equiv \log n \ll e^{\sqrt{\log n}} \ll n^{1/4} \ll \frac{n}{\log n} \ll n \equiv n + \sqrt{n} \ll 0.01n^2 \ll 2^n \ll e^n \ll n!$$

Anmerkungen

- Die meisten Vergleiche können durch Theorem 1.1 aus *Algorithmen und Datenstrukturen* ermittelt werden. Es besagt unter anderem, dass
$$\lim_{n\to\infty} \frac{g(n)}{f(n)} = \infty \qquad \Longrightarrow \qquad f = o(g)$$
und
$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = C \text{ für ein } C \in \mathbb{R}^+ \qquad \Longrightarrow \qquad f = \Theta(g).$$
Zur Auswertung des Limes ist oft der Satz von L'Hôpital hilfreich.
- 2^{32} ist ein Konstante.
- $\log(n^2) = 2 \cdot \log n$ und $\log n = \frac{\ln n}{\ln 2} = 1.442\ldots \cdot \ln n$.
- $\frac{e^{\sqrt{\log n}}}{\log n} = \frac{e^{\sqrt{\log n}}}{e^{\log \log n}} = e^{\sqrt{\log n} - \log \log n} \to \infty$, weil $\sqrt{x} - \log x \to \infty$ für $x \to \infty$. (Hier mit $x = \log n$.)
- $\frac{n^{1/4}}{e^{\sqrt{\log n}}} = \frac{e^{1/4 \cdot \log n}}{e^{\sqrt{\log n}}} = e^{1/4 \cdot \log n - \sqrt{\log n}} \to \infty$, weil $\frac{1}{4}x - \sqrt{x} \to \infty$ für $x \to \infty$. (Wieder mit $x = \log n$.)
- In $n!$ stecken $n - 3$ Faktoren, die alle mindestens Grösse 4 haben. Daher ist $n! \geq 4^{n-3} = \frac{1}{64} \cdot 4^n = \omega(e^n)$.

Aufgabe 3 – *Induktion*

(a) **Induktionsanfang:** Für $n = 1$ sind beide Seiten $1/2$, daher stimmt die Behauptung.

Induktionsvoraussetzung: $\frac{1}{1\cdot 2} + \frac{1}{2\cdot 3} + \frac{1}{3\cdot 4} + \ldots + \frac{1}{n(n+1)} = \frac{n}{n+1}$.

Induktionsbehauptung: $\frac{1}{1\cdot 2} + \frac{1}{2\cdot 3} + \ldots + \frac{1}{n\cdot(n+1)} + \frac{1}{(n+1)(n+2)} = \frac{n+1}{n+2}$.

Induktionsschritt: Es gilt

$$\begin{aligned}\frac{1}{1\cdot 2} + \frac{1}{2\cdot 3} + \ldots + \frac{1}{n(n+1)} + \frac{1}{(n+1)\cdot(n+2)} &\overset{I.V.}{=} \frac{n}{n+1} + \frac{1}{(n+1)(n+2)} \\ &= \frac{n^2+2n+1}{(n+1)(n+2)} = \frac{(n+1)^2}{(n+1)(n+2)} \\ &= \frac{n+1}{n+2}.\end{aligned}$$

(b) Sei $h \geq -1$ fest vorgegeben. Wir verwenden Induktion über n.

Induktionsanfang: Für $n = 1$ sind beide Seiten gleich $1+h$, daher stimmt die Behauptung.

Induktionsvoraussetzung: $1 + nh \leq (1+h)^n$.

Induktionsbehauptung: $1 + (n+1)h \leq (1+h)^{n+1}$.

Induktionsschritt: Es gilt

$$\begin{aligned}(1+h)^{n+1} &= (1+h)^n \cdot (1+h) \overset{\text{I.V., und } (1+h)\geq 0}{\geq} (1+nh)\cdot(1+h) = 1 + (n+1)h + nh^2 \\ &\geq 1 + (n+1)h.\end{aligned}$$

Aufgabe 4 – *Eine generelle Eigenschaft von Graphen*

Wir zeigen die Aussage mit dem Schubfachprinzip. Nehmen Sie an, dass der Graph n Knoten hat. Jeder dieser Knoten ist mit entweder $0, 1, 2, \ldots, n-2$ oder $n-1$ anderen Knoten verbunden. Falls es einen Knoten mit Grad $n-1$ gibt, so ist dieser mit allen anderen verbunden und es kann keinen Knoten mit Grad 0 geben. Es ist also unmöglich einen Graphen mit n Knoten zu haben indem es sowohl einen Knoten mit Grad 0 als auch einen Knoten mit Grad $n-1$ gibt. Die Knoten können also höchstens $n-1$ verschiedene Grade haben, da es n Knoten gibt haben nach dem Schubfachprinzip wenigstens zwei Knoten den gleichen Grad.

Aufgabe 5 – *Algorithmus*

Es gibt viele Arten dieses Problem zu lösen. Hier betrachten wir einen Algorithmus der auf dem Statement aus 6 (b) basiert.

Algorithmus: Als erstes prüfen wir ob $|E| = |V| - 1$. Dies tun wir in dem wir durch die Kanten gehen und abbrechen, sobald wir $|V|$ Kanten gesehen haben. Falls $|E| \neq |V| - 1$, geben wir sofort "Nein" aus. Sonst wählen wir einen Knoten v und führen eine Tiefensuche beginnend bei v durch. Dann prüfen wir, ob alle Knoten als besucht markiert sind. Falls sie das sind, geben wir "Ja" aus, ansonsten geben wir "Nein" aus.

Korrektheitsbeweis: Falls wir mehr als $|V|$ Kanten haben gilt sicher $|E| > |V| - 1$ somit reicht es die ersten $|V|$ Kanten zu betrachten, um zu entscheiden ob $|E| \neq |V| - 1$. Wir wissen, dass die Tiefensuche, die in einem Knoten v beginnt genau die Knoten besucht, die in der Zusammenhangskomponente von v sind. Daher gibt der Algorithmus "Ja" aus, genau dann wenn $|E| = |V| - 1$ und der Graph zusammenhängend ist. Nach 6(b), ist dies äquivalent dazu, dass der Graph ein Baum ist.

Laufzeitanalyse: Um zu entscheiden, ob $|E| = |V| - 1$ oder nicht, gehen wir die Kanten eine nach der anderen durch (in beliebiger Reihenfolge). Da wir abbrechen, wenn wir $|V|$ Kanten gesehen haben können wir dies in $O(|V|)$ tun. (Man beachte, das wir an dieser Stelle das wir $O(|V| + |E|)$ Zeit bräuchten um den ganzen Graphen einzulesen)

Die Tiefensuche durchzuführen braucht normalerweise $O(|V| + |E|)$ Zeit. Da wir diese nur ausführen falls $|E| = |V| - 1$, läuft diese in $O(|V| + |V| - 1) = O(|V|)$. Zu prüfen ob alle Knoten von der Tiefensuche besucht wurden können wir in $O(|V|)$ durchführen.

Also ist die Gesamtlaufzeit unseres Algorithmus $O(|V|)$.

Aufgabe 6 – *Charakterisierung von Bäumen (Challenge-Aufgabe)*

Siehe Satz 1.6 im Skript.