

Algorithmische Graphentheorie

10. Übungsblatt

Jasper Gude

Pia Röttgers

29. Juni 2026

14 / 20

Aufgabe 1 – Größte Clique

- ✓ a) Wir prüfen für jede Teilmenge $V' \subseteq V$, ob sie eine Clique ist, d. h. wir prüfen für jeden Knoten in V' ob es eine Kante zu jedem anderen Knoten in V' gibt.

Dabei fangen wir mit der größten Teilmenge an und werden schrittweise kleiner. Wenn wir eine Clique finden ist sie die größte.

Exakt haben wir eine Laufzeit von

$$\sum_{i=1}^n \binom{n}{i} \cdot i^2$$

Es existieren ~~maximal~~ ^{...genau...} 2^n viele Teilmengen. Für jede Teilmenge V' prüfen wir für jedes Knotenpaar, ob sie adjazent sind. Das geht in $\mathcal{O}(n^2)$ Zeit. Also $\mathcal{O}(2^n \cdot n^2)$.

2 / 2

- ✓ b) Wir prüfen diesmal für jede Teilmenge der Nachbarschaft $N(v)$ von v (insbesondere ist v in der $N(v)$) ob sie eine Clique ist. Wir fangen wieder mit den größten Teilmengen an. Wenn wir eine Clique finden, ist sie die größte, die v enthält.

Die exakte Laufzeit ist damit

$$\sum_{i=1}^{\deg(v)} \binom{|N(v)|}{i} \cdot i^2$$

v ist nicht in $N(v)$ so wie wir Nachbarschaft definiert haben. Dafür keine Abzüge, nur der Hinweis dass diese Aussage so nicht stimmt.=)

(Es ist hier auch egal, ob man v mit einbezieht oder nicht. Durch die Definition von $N(v)$ sind alle Kanten zwischen v und Knoten in $N(v)$ ja bereits "geprüft")

Den Knotengrad können wir mit dem Maximalknotengrad Δ abschätzen, ebenso wie die Größe von $N(v)$. Somit erhalten wir

$$\sum_{i=1}^{\Delta} \binom{\Delta}{i} \cdot i^2$$

Das heißt wir bekommen asymptotisch $\mathcal{O}(2^{\Delta} \cdot \Delta^2)$.

2 / 2

- ✓ c) Mit dem Ansatz aus Teilaufgabe b) können wir einen Algorithmus für GRÖSSTE CLIQUE mit Parameter Δ konstruieren.

Dafür iterieren wir über jeden Knoten $v \in V$ und führen für ihn unseren Algorithmus aus.

Das führt zu einer Laufzeit von $\mathcal{O}((2^{\Delta} \cdot \Delta^2) \cdot n)$, was in FPT liegt, da das die Form $\mathcal{O}(f(k) \cdot |I|^c)$ mit Parameter k , Instanz I und Konstante c erfüllt.

2 / 2

Aufgabe 2 – Geradenüberdeckung

a) viele habens nur mit der ersten Aussage versucht, die würde mir nicht reichen. Der zweite Satz alleine wäre mir genug. ;)

- a) Eine Gerade wird hinreichend bestimmt durch zwei Punkte, durch die sie verläuft. Bei $n \geq 2$ können wir jede Gerade die nur einen Punkt überdeckt, durch eine Gerade ersetzen, die durch mindestens zwei Punkte verläuft.

1 / 1

- b) Jede Gerade überdeckt mindestens zwei Punkte. Es gibt n viele Punkte. Also brauchen wir maximal $n / 2$ Geraden.

Folglich besteht ein Dreieck aus 1.5 Geraden. Hm...

0 / 1

- c) Eine Gerade die mehr als k Punkte überdeckt ist in der Geradenüberdeckung.

Angenommen wir haben eine Geradenüberdeckung C bei der jede Gerade höchstens k Punkte überdeckt. Wenn es eine Gerade g gibt, die mehr als k Punkte überdeckt und nicht in C ist, dann wird sie an maximal k Punkten geschnitten, da in C höchstens k Geraden sind. Das bedeutet aber, dass nicht alle Punkte, die auf g sind geschnitten und damit überdeckt werden. Das ist ein Widerspruch zur Annahme. Somit muss g in der Geradenüberdeckung sein.

1 / 1

- d) Unter der Annahme, dass es keine Gerade gibt, die mehr als k Punkte enthält gilt: Für $k < \sqrt{n}$ gibt es keine Geradenüberdeckung.

Angenommen jede Gerade überdeckt genau k Punkte. Dann überdecken wir mit k Geraden maximal k^2 Punkte. Da $k < \sqrt{n}$ sind das $k^2 < n$ viele Punkte und somit keine Geradenüberdeckung.

1 / 1

e) k-GERADENÜBERDECKUNG:

1. Ist $k \geq n/2$, gib true zurück. 1. zu unscharfe Schranke nach b), aber der Punkt ging oben schon drauf. Fall berücksichtigt, ergo gut. =)
2. Erzeuge die Menge G aller Geraden, die von zwei Punkten aufgespannt werden. Das sind n^2 viele.
3. Bestimme für jede Gerade die Anzahl der überdeckten Punkte. Nehme die Geraden, die mehr als k Punkte überdecken in die Geradenüberdeckung C. Wir überprüfen für alle n^2 Geraden, ob sie noch weitere Punkte überdecken. Also $n^2 \cdot n$ Überprüfungen.
4. Ist $C = \emptyset$ und $k < \sqrt{n}$, gib false zurück.
5. Löse den Rest mit Brute Force.
Prüfe dafür für alle $A \in \binom{G \setminus C}{k - |C|}$ ob $A \cup C$ alle Punkte überdeckt. Falls ja, gib true zurück, sonst gib false zurück.

Im Worst Case finden wir keine Geraden die mehr als k Punkte überdecken. Dann iterieren wir über alle $\binom{|G|}{k}$ Mengen von Geraden. Da $k \geq \sqrt{n}$ gilt und $|G| = n^2$ können wir folgende Abschätzung machen.

$$\binom{|G|}{k} = \binom{n^2}{k} = \binom{(\sqrt{n})^4}{k} \leq \binom{k^4}{k}$$

Die Anzahl der Teilmengen A passt, allerdings ist der Aufwand für den markierten Teilschritt aus Schritt 5 nicht berücksichtigt. Hier müssen wir je A worst case für alle k^2 Punkte noch für k Geraden schauen, ob sie den Punkt enthalten oder nicht.

Wir erhalten somit eine Gesamtlaufzeit von

$$\mathcal{O} \left(n^2 + n^3 + \overbrace{\binom{k^4}{k} \cdot n}^{\text{Für jede Auswahl an Geraden alle Punkte prüfen}} \right)$$

Das liegt in FPT.

Es wäre noch zu zeigen, was im Fall $C \neq \emptyset$ und $k < \sqrt{n}$ passiert. Das jedoch bleibt dem Leser als Übungsaufgabe überlassen.

4 / 5

Aufgabe 3 – Mehrgüterfluss

a) Siehe Abbildung 1.

b) Siehe Abbildung 2

raphen. Es sei $c(e) = 1$ für alle e an (mit Begründung) unsswert besitzt.

1 / 2

0 / 3



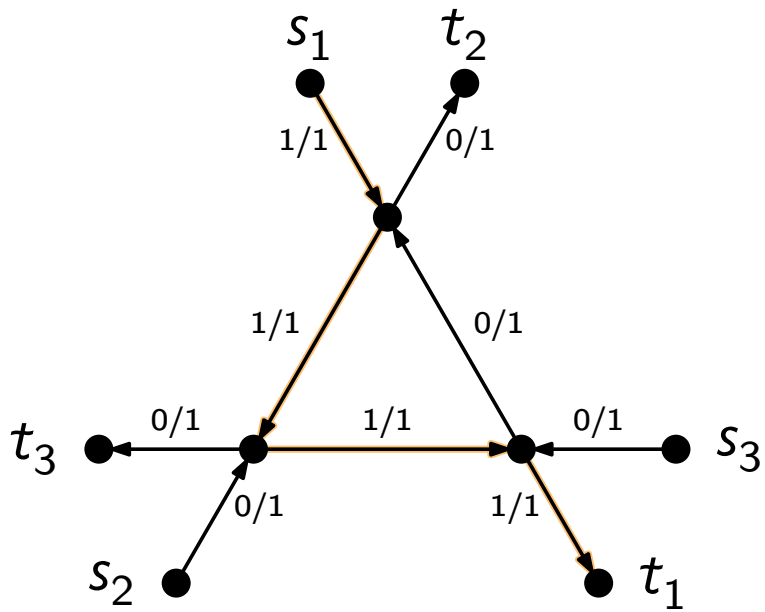


Abbildung 1: Maximaler Fluss mit Gesamtflusswert 1.

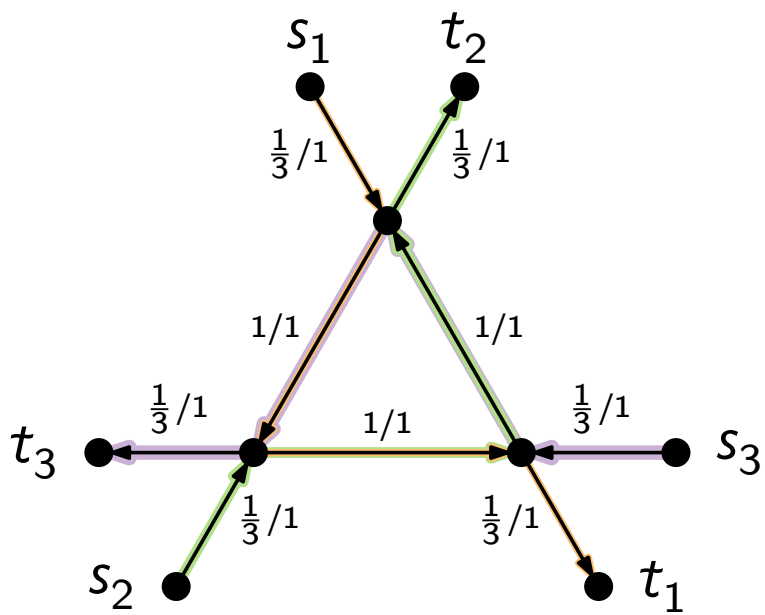


Abbildung 2: Maximaler Fluss mit Gesamtflusswert 1.

Ein Fluss von $\frac{1}{3}$ fließt hier unnötig im Kreis.
Ihr könnt die drei Einzel-Flüsse auf jeweils 0.5
erhöhen ohne Kapazitäten zu sprengen.