

Musterlösung – Probeklausur 1

Analyse von Algorithmen und Komplexität, SS 2026

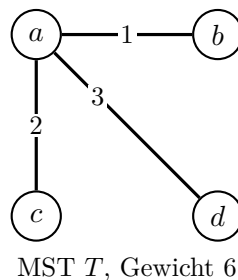
Aufgabe 1 ANWENDUNG – CHRISTOFIDES' ALGORITHMUS (6 + 4 Punkte)

Kantengewichte: $ab = 1$, $ac = 2$, $ad = 3$, $bc = 3$, $bd = 4$, $cd = 4$.

Lösung.

a) Anwendung von Christofides (Δ TSP2) ab Startknoten a .

Schritt 1 – Minimaler Spannbaum T . Kruskal (aufsteigend ab 1, ac 2, ad 3, bc 3, bd 4, cd 4): Wähle ab (1), ac (2); ad (3) verbindet d . Die nächstteure Kante bc (3) würde einen Kreis schließen (*No-Op*: wird geprüft, aber verworfen); ebenso bd (4), cd (4). $T = \{ab, ac, ad\}$ ist ein Stern um a mit Gewicht $1 + 2 + 3 = 6$ und eindeutig.



Schritt 2 – Knoten mit ungeradem Grad. Grade in T : $\deg(a) = 3$, $\deg(b) = \deg(c) = \deg(d) = 1$. Alle vier Grade sind ungerade (jeder Grad wird geprüft), also $X = \{a, b, c, d\}$.

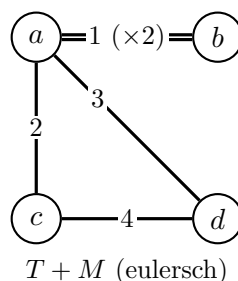
Schritt 3 – Minimales perfektes Matching M auf X . Die drei möglichen perfekten Matchings auf $\{a, b, c, d\}$:

- $\{ab, cd\} = 1 + 4 = 5$
- $\{ac, bd\} = 2 + 4 = 6$
- $\{ad, bc\} = 3 + 3 = 6$

Minimum eindeutig: $M = \{ab, cd\}$ mit Kosten 5.

Schritt 4 – Multigraph $T + M$ und Eulerkreis. Die Kante ab kommt sowohl aus T als auch aus M , ist also doppelt. Kanten: ab (doppelt), ac , ad , cd . Grade: $\deg(a) = 4$, $\deg(b) = 2$, $\deg(c) = 2$, $\deg(d) = 2$ – alle gerade, der Multigraph ist eulersch. Eulerkreis ab a :

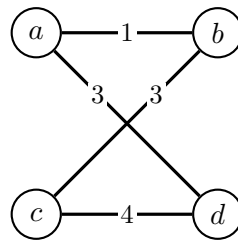
$[a, b, a, c, d, a]$.



Schritt 5 – Abkürzen (shortcut). Durchlaufe den Eulerkreis $[a, b, a, c, d, a]$ und überspringe bereits besuchte Knoten: a (neu, behalten – *No-Op*), b (neu, behalten – *No-Op*), a (*bereits besucht* $\Rightarrow$ überspringen), c (neu), d (neu), a (Schluss). Resultierende Tour:

$[a, b, c, d, a]$.

Tourlänge = $ab + bc + cd + da = 1 + 3 + 4 + 3 = 11$.



Tour $[a, b, c, d, a]$, Länge 11

Zum Vergleich (Brute-Force über alle 3 Touren): $[a, b, c, d, a] = 11$, $[a, b, d, c, a] = 1 + 4 + 4 + 2 = 11$, $[a, c, b, d, a] = 2 + 3 + 4 + 3 = 12$. Also $\text{OPT} = 11$; hier liefert Christofides sogar das Optimum, die Güte $\frac{3}{2}$ ist eingehalten ($\frac{11}{11} = 1 \leq \frac{3}{2}$).

b) Bedeutung, Zusatzeigenschaften, Güte.

- **Was berechnet Christofides?** Eine Rundreise (Hamiltonkreis) durch alle Knoten, also eine zulässige Näherungslösung für das metrische Travelling-Salesperson-Problem.
- **Zusatzeigenschaft 1 – Symmetrie:** $w(u, v) = w(v, u)$ für alle Kanten (die Kostenfunktion ist ungerichtet). *Verletzende Eingabe:* ein gerichteter Graph mit $w(a, b) = 1$, aber $w(b, a) = 5$. Dann sind MST und Matching nicht wohldefiniert bzw. die Analyse bricht.
- **Zusatzeigenschaft 2 – Dreiecksungleichung:** $w(u, v) \leq w(u, x) + w(x, v)$ für alle Knoten u, x, v . *Verletzende Eingabe:* drei Knoten a, b, c mit $w(a, b) = 1$, $w(b, c) = 1$, $w(a, c) = 10$; hier ist $w(a, c) = 10 > 1 + 1 = w(a, b) + w(b, c)$. Ohne Dreiecksungleichung liefert das Abkürzen keine Abschätzung der Tourlänge.

(Eine einzige Eingabe, die *beide* Eigenschaften verletzt, wäre ebenfalls zulässig.) **Güte:** $\frac{3}{2}$. $\square$

Aufgabe 2 VORLESUNGSBEWEIS – DEFINITION NP (2 + 8 Punkte)

Zu zeigen: Existiert ein polynomieller Verifizierer für L , so existiert eine NDTM, die L in nicht-deterministischer Polynomialzeit entscheidet.

Lösung.

a) Analoge NDTM zum gegebenen VertexCover-Verifizierer. Der Verifizierer rät nichts, sondern *prüft* ein von außen gegebenes Zertifikat C . Die analoge NDTM ersetzt dieses Zertifikat durch eine *nichtdeterministische Ratephase* und führt danach dieselben Prüfungen aus:

$M(G = (V, E), k)$

1. Rate nichtdeterministisch eine Menge von Knoten $C \subseteq V$.
2. Falls $|C| > k$, verwurfe.
3. Für alle $\{u, v\} \in E$: Falls $u \notin C$ und $v \notin C$, verwurfe.
4. Akzeptiere.

Korrektheit: M hat genau dann einen akzeptierenden Rechenweg, wenn ein $C \subseteq V$ existiert, das die Prüfungen 2–3 übersteht, d.h. ein Vertex Cover der Größe $\leq k$ – genau dann akzeptiert auch V ein Zertifikat. Also akzeptiert M dieselbe Sprache L .

Laufzeit: Raten von C in $O(|V|)$, Größe prüfen $O(|V|)$, Kanten prüfen $O(|E|)$ – jeder Rechenweg läuft in $O(|V| + |E|)$, also polynomiell.

b) Allgemeingültiger Beweis. Sei L eine Sprache mit polynomielltem Verifizierer M der Laufzeit $p(|x|) = O(|x|^d)$: Für $x \in L$ existiert ein Zertifikat c mit $M(x, c) = 1$; für $x \notin L$ akzeptiert M kein Zertifikat. In $p(|x|)$ Schritten liest M höchstens die ersten $p(|x|)$ Zeichen des Zertifikats.

Konstruktion der NDTM N . Bei Eingabe x arbeitet N in zwei Phasen:

- *Raten:* Schreibe mit nichtdeterministischen Entscheidungen einen String u mit $|u| \leq p(|x|)$ auf ein Arbeitsband.
- *Verifizieren:* Simuliere M auf (x, u) und übernehme dessen Antwort (akzeptiere/verwirf).

Korrektheit:

- Sei $x \in L$. Dann existiert ein Zertifikat c mit $M(x, c) = 1$; davon liest M höchstens die ersten $p(|x|)$ Zeichen. Der Rechenweg, der genau diesen Anfang als u rät, akzeptiert – also akzeptiert N .
- Sei $x \notin L$. Dann verwirft M jedes Zertifikat, insbesondere jedes geratene u . Also verwerfen alle Rechenwege von N , und N verwirft.

Somit akzeptiert N genau die Sprache L .

Laufzeit: Phase 1 braucht $\leq p(|x|)$ Rateschritte, Phase 2 läuft in $O(|x|^d)$ – insgesamt $O(|x|^d)$, also polynomiell. Damit entscheidet die NDTM N die Sprache L in nicht-deterministischer Polynomialzeit $T = O(|x|^d)$. $\square$

Aufgabe 3 NP-VOLLSTÄNDIGKEIT

(10 Punkte)

Zu zeigen: k -CLIQUEMEMBER ist NP-vollständig.

Lösung.

∈ **NP**: Eingabe: Graph $G = (V, E)$, Knoten v , Zahl k und Zertifikat $C \subseteq V$. Der Verifizierer arbeitet wie folgt:

- Falls $|C| < k$ oder $v \notin C$, lehne ab.
- Prüfe für jedes Paar $\{x, y\} \subseteq C$, ob $\{x, y\} \in E$; lehne sonst ab.
- Sonst akzeptiere.

Existiert eine Clique C mit $|C| \geq k$ und $v \in C$, so ist sie ein akzeptiertes Zertifikat; sonst scheitert jedes Zertifikat an einer Prüfung. Laufzeit: Größe und Mitgliedschaft prüfen $O(|V|)$, alle Paare testen $O(|V|^2)$ – gesamt $O(|V|^2)$. Also gilt k -CLIQUEMEMBER ∈ **NP**.

NP-schwer: Zeige die NP-Schwere durch die Reduktion

$$\text{CLIQUE} \leq_p k\text{-CLIQUEMEMBER}.$$

CLIQUE ist bereits als NP-vollständig bekannt.

Konstruktion: Sei $(G = (V, E), k)$ eine CLIQUE-Instanz. Wir konstruieren daraus eine k -CLIQUEMEMBER-Instanz $(G' = (V', E'), v, k')$ durch:

- $V' = V \cup \{v\}$ mit einem neuen Knoten $v \notin V$
- $E' = E \cup \{\{v, w\} \mid w \in V\}$ (v ist mit allen Knoten aus V verbunden, also universell)
- $k' = k + 1$, ausgezeichneter Knoten v

Beweis hin:

- Sei C eine Clique in G mit $|C| \geq k$.
- Da v mit allen Knoten von $V \supseteq C$ verbunden ist, sind alle Paare in $C \cup \{v\}$ Kanten in E' ; also ist $C \cup \{v\}$ eine Clique in G' .
- Wegen $v \notin V \supseteq C$ gilt $|C \cup \{v\}| = |C| + 1 \geq k + 1 = k'$, und $v \in C \cup \{v\}$.
- Also ist (G', v, k') eine Ja-Instanz.

Beweis zurück:

- Sei C' eine Clique in G' mit $|C'| \geq k'$ und $v \in C'$.
- Setze $C = C' \setminus \{v\}$; wegen $v \in C'$ gilt $|C| = |C'| - 1 \geq k' - 1 = k$, und $C \subseteq V$.
- Alle Kanten zwischen Knoten von C liegen in E , denn neu in E' sind nur die Kanten mit Endpunkt v , und $v \notin C$.
- Also ist C eine Clique in G mit $|C| \geq k$.

(Beide Richtungen zusammen: (G', v, k') ist Ja-Instanz $\iff (G, k)$ ist Ja-Instanz. Der Randfall $k \geq 1$ ist gewährleistet, da $k' = k + 1 \geq 2$ und v stets in der Clique liegen muss.)

Laufzeit: Einen Knoten und $|V|$ Kanten anlegen – $O(|V| + |E|)$, polynomiell.

Da k -CLIQUEMEMBER ∈ **NP** und NP-schwer ist, folgt: k -CLIQUEMEMBER ist NP-vollständig. $\square$

Lösung.

a) **Parameter der VertexCover-Instanz.** Es gilt $V' = V$, also

$$|V'| = |V|, \quad |E'| = \binom{|V|}{2} - |E| = \frac{1}{2} |V| (|V| - 1) - |E| \leq |V|^2.$$

(Die Komplementkantenmenge E' enthält alle $\binom{|V|}{2}$ möglichen Kanten außer den $|E|$ Kanten von G .)

b) **ETH-Schranken für VertexCover.** Die Reduktion $\text{CLIQUE} \leq_p \text{VERTEXCOVER}$ läuft in Polynomialzeit.

Bezüglich $|V'|$:

- Angenommen, VERTEXCOVER wäre in $2^{o(|V'|)} \cdot |I|^{O(1)}$ lösbar.
- Da die Reduktion polynomiell ist und $|V'| = |V|$ gilt, ergäbe die Komposition (Reduktion, dann Algorithmus) einen $2^{o(|V|)} \cdot |I|^{O(1)}$ -Algorithmus für CLIQUE.
- Widerspruch zur ETH ($\text{CLIQUE} \notin 2^{o(|V|)} \cdot |I|^{O(1)}$).
- Also besitzt VERTEXCOVER keinen $2^{o(|V'|)} \cdot |I|^{O(1)}$ -Algorithmus.

Bezüglich $|E'|$:

- Angenommen, VERTEXCOVER wäre in $2^{o(\sqrt{|E'|})} \cdot |I|^{O(1)}$ lösbar.
- Wegen $|E'| \leq \frac{1}{2} |V| (|V| - 1) \leq |V|^2$ ist $\sqrt{|E'|} \leq |V|$, also $o(\sqrt{|E'|}) \subseteq o(|V|)$.
- Damit wäre VERTEXCOVER in $2^{o(|V|)} \cdot |I|^{O(1)}$ lösbar, und mit $|V'| = |V|$ (Reduktion polynomiell) folgt ein $2^{o(|V|)} \cdot |I|^{O(1)}$ -Algorithmus für CLIQUE.
- Widerspruch zur ETH.
- Also besitzt VERTEXCOVER keinen $2^{o(\sqrt{|E'|})} \cdot |I|^{O(1)}$ -Algorithmus.

Ergebnis: kein $2^{o(|V'|)}$ und kein $2^{o(\sqrt{|E'|})}$.

c) **Parameter der HittingSet-Instanz.** Die Grundmenge besteht aus den $2n$ Literalen, und es gibt n Paar-Mengen sowie m Klausel-Mengen:

$$|U| = 2n, \quad r = n + m.$$

d) **ETH-Schranken für HittingSet.** Die Reduktion $3\text{-SAT} \leq_p \text{HITTINGSET}$ läuft in Polynomialzeit. (Sie ist korrekt, weil die n disjunkten Paar-Mengen bei $k = n$ erzwingen, dass H aus jedem Paar $\{x_i, \bar{x}_i\}$ genau ein Literal enthält – also eine Belegung kodiert –, und das Treffen der Klausel-Mengen genau der Erfüllung der Klauseln entspricht.)

Bezüglich $|U|$:

- Angenommen, HITTINGSET wäre in $2^{o(|U|)} \cdot |I|^{O(1)}$ lösbar.

- Wegen $|U| = 2n$ ist $o(|U|) = o(n)$; die Komposition ergäbe einen $2^{o(n)} \cdot |I|^{O(1)}$ -Algorithmus für 3-SAT.
- Direkter Widerspruch zur ETH.
- Also kein $2^{o(|U|)} \cdot |I|^{O(1)}$ -Algorithmus.

Bezüglich r :

- Angenommen, HITTINGSET wäre in $2^{o(r)} \cdot |I|^{O(1)}$ lösbar.
- Wegen $r = n + m$ und $n \leq 3m$ gilt $r \leq 4m = O(m)$, also $o(r) \subseteq o(m)$; die Komposition ergäbe einen $2^{o(m)} \cdot |I|^{O(1)}$ -Algorithmus für 3-SAT.
- Widerspruch zur ETH nach dem Sparsification-Lemma ($3\text{-SAT} \notin 2^{o(m)} \cdot |I|^{O(1)}$).
- Also kein $2^{o(r)} \cdot |I|^{O(1)}$ -Algorithmus.

Ergebnis: kein $2^{o(|U|)}$ und kein $2^{o(r)}$.

□

Aufgabe 5 APPROXIMATIVE ALGORITHMEN – MAX-3-SAT (2 + 5 + 3 + (10) Punkte)

Lösung.

a) **Anwendung auf φ .** Dabei setzt β_0 alle Variablen auf falsch und β_1 alle auf wahr. Klauselauswertung:

Klausel	unter β_0	unter β_1
$C_1 = (x_1 \vee x_2 \vee x_3)$	falsch (alle Lit. falsch)	wahr
$C_2 = (\neg x_1 \vee \neg x_2 \vee \neg x_3)$	wahr ($\neg x_1$)	falsch
$C_3 = (x_1 \vee \neg x_2 \vee x_3)$	wahr ($\neg x_2$)	wahr
$C_4 = (\neg x_1 \vee x_2 \vee \neg x_3)$	wahr ($\neg x_1$)	wahr

Also $v(\beta_0) = 3$ und $v(\beta_1) = 3$. Der Algorithmus gibt (bei Gleichstand β_0) den Wert $v(A(\varphi)) = \max\{3, 3\} = 3$ aus.

Optimum (Brute-Force über alle 8 Belegungen): Die Belegung $(x_1, x_2, x_3) = (\text{falsch}, \text{falsch}, \text{wahr})$ erfüllt C_1 (durch x_3), C_2 (durch $\neg x_1$), C_3 (durch $\neg x_2$) und C_4 (durch $\neg x_1$), also alle vier. Somit $v(\text{OPT}(\varphi)) = 4$. (Verhältnis $\frac{\text{OPT}}{A} = \frac{4}{3}$.)

b) **Güte 2.** Zu zeigen: $v(A(\varphi)) \geq \frac{1}{2} v(\text{OPT}(\varphi))$. Sei φ eine Formel mit m Klauseln.

Jede Klausel zählt mindestens einmal. Unter β_0 ist jedes *negative* Literal wahr, unter β_1 jedes *positive*. Da jede Klausel mindestens ein Literal enthält und jedes Literal positiv oder negativ ist, wird jede Klausel von β_0 oder von β_1 (oder beiden) erfüllt. Somit zählt jede Klausel in der Summe mindestens einmal:

$$v(\beta_0) + v(\beta_1) \geq m.$$

Obere Schranke für das Optimum. Höchstens alle m Klauseln können erfüllt sein, also $v(\text{OPT}(\varphi)) \leq m$.

Kombination. A wählt die bessere der beiden Belegungen, daher

$$v(A(\varphi)) = \max\{v(\beta_0), v(\beta_1)\} \geq \frac{v(\beta_0) + v(\beta_1)}{2} \geq \frac{m}{2} \geq \frac{v(\text{OPT}(\varphi))}{2}.$$

Also hat A die Güte 2.

c) **Scharfe Instanz.** Wähle

$$\varphi^* = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3).$$

- Unter β_0 (alles falsch) ist die erste Klausel unerfüllt (alle Literale positiv, also falsch) und die zweite erfüllt: $v(\beta_0) = 1$.
- Unter β_1 (alles wahr) ist die zweite Klausel unerfüllt und die erste erfüllt: $v(\beta_1) = 1$.
- Also $v(A(\varphi^*)) = \max\{1, 1\} = 1$.
- Die Belegung $(x_1, x_2, x_3) = (\text{wahr}, \text{falsch}, \text{falsch})$ erfüllt beide Klauseln (x_1 bzw. $\neg x_2$), also $v(\text{OPT}(\varphi^*)) = 2$.

Damit gilt $v(A(\varphi^*)) = 1 = \frac{1}{2} \cdot 2 = \frac{1}{2} v(\text{OPT}(\varphi^*))$: die Schranke aus b) wird mit Gleichheit angenommen, die Güte 2 ist scharf.

d) BONUS – MAX- k -SAT. *Behauptung:* Für jedes $k \geq 2$ garantiert derselbe Algorithmus A ebenfalls die Güte 2, und diese ist scharf.

Güte 2 (Beweis). Das Argument aus b) benutzt nur, dass jede Klausel mindestens ein Literal enthält und jedes Literal positiv oder negativ ist – es ist von der Klausellänge unabhängig. Auch bei genau k Literalen pro Klausel gilt: eine Klausel wird von β_0 nur dann nicht erfüllt, wenn all ihre k Literale positiv sind; von β_1 nur dann nicht, wenn alle k negativ sind. Beides zugleich ist bei $k \geq 1$ unmöglich. Also wird jede Klausel von β_0 oder β_1 erfüllt, mithin $v(\beta_0) + v(\beta_1) \geq m$, und wie in b)

$$v(A(\varphi)) = \max\{v(\beta_0), v(\beta_1)\} \geq \frac{m}{2} \geq \frac{v(\text{OPT}(\varphi))}{2}.$$

Die garantierte Güte ist also 2, unabhängig von k .

Schärfe (für jedes k). Betrachte über den Variablen $x_1, \dots, x_k$ die beiden Klauseln

$$\varphi_k^* = (x_1 \vee \dots \vee x_k) \wedge (\neg x_1 \vee \dots \vee \neg x_k).$$

Jede Klausel besteht aus genau k Literalen mit paarweise verschiedenen Variablen. β_0 erfüllt nur die zweite, β_1 nur die erste, also $v(A(\varphi_k^*)) = 1$. Die Belegung $x_1 = \text{wahr}$, $x_2 = \dots = x_k = \text{falsch}$ erfüllt beide Klauseln, also $v(\text{OPT}(\varphi_k^*)) = 2$. Damit $v(A(\varphi_k^*)) = \frac{1}{2} v(\text{OPT}(\varphi_k^*))$, die Güte 2 wird für jedes k mit Gleichheit angenommen und ist somit scharf. $\square$