Institut für Informatik
Prof. Dr. J. Rothe

Wintersemester 2023/24
Übungsblatt 12

Lösungsvorschläge
Kryptokomplexität 1

Bearbeitungszeit: 9. Januar bis 24.-26. Januar
Verantwortlich: Roman Zorn

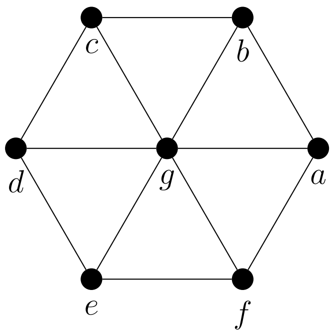
Aufgabe 1 : VertexCover und SetCover

Sei $G = (V, E)$ ein ungerichteter Graph. Eine Teilmenge $C \subseteq V$ heißt *Knotenüberdeckung* (engl. *vertex cover*) von G, falls für alle Kanten $\{x, y\} \in E$ gilt $\{x, y\} \cap C \neq \emptyset$. Das entsprechende Entscheidungsproblem VertexCover ist wie folgt definiert.

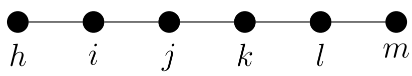
VertexCover	
Gegeben:	Ein Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$.
Frage:	Besitzt G eine Knotenüberdeckung $C \subseteq V$ der Größe höchstens k?

(a) Betrachten Sie die VertexCover-Instanzen

(i) $I = (G, 4)$ mit $G = (V, E)$ wie folgt:



(ii) und $I' = (H, 2)$ mit $H = (V', E')$ wie folgt:



▶ Entscheiden Sie, ob I und I' ja-Instanzen für VertexCover sind und begründen Sie Ihre Antworten.

(b) Sei $I = (G, k)$ eine VertexCover-Instanz mit $G = (V, E)$. Betrachten Sie die folgende SetCover-Instanz (siehe Blatt 11, Aufgabe 4) $I' = (E, \mathcal{S}, k)$ mit

$$\mathcal{S} = \{S_i \mid v_i \in V\} \text{ und } S_i = \{e \in E \mid e \cap \{v_i\} \neq \emptyset\}.$$

▶ Zeigen Sie: $I \in$ VertexCover *gilt genau dann, wenn* $I' \in$ SetCover.

Lösungsvorschlag:

(a) I ist eine JA-Instanz für VERTEXCOVER, da z.B. $\{b, d, f, g\}$ eine Knotenüberdeckung der Größe höchstens 4 für G bildet.

I' ist eine NEIN-Instanz für VERTEXCOVER, da alle Knoten höchstens zwei Nachbarn haben. Ein Knoten kann folglich höchstens zwei Kanten abdecken. Damit kann eine Knotenüberdeckung der Größe höchstens 2 maximal 4 der 5 Kanten von H abdecken.

(b) Wir beweisen beide Richtungen einzeln:

"$\Rightarrow$" Wir nehmen an, dass $I \in$ VERTEXCOVER gilt. Dann existiert eine Knotenüberdeckung $C \subseteq V$ mit $|C| \leq k$ für G. Nun fügen wir für jeden Knoten $v_i \in C$ die Menge S_i zu $\mathcal{S}'$ hinzu. Offensichtlich gilt dann $|\mathcal{S}'| \leq k$. Es bleibt zu argumentieren, dass $\mathcal{S}'$ eine Überdeckung für E ist. Sei dazu $e \in E$ eine beliebige Kante. Da C eine Knotenüberdeckung ist, existiert mindestens ein $v_i \in C$, so dass v_i mit e inzident ist, d.h. es gilt $e \cap \{v_i\} \neq \emptyset$. Gem. der Konstruktion von I' gilt also auch $e \in S_i$ und gem. der Konstruktion von $\mathcal{S}'$ auch $S_i \in \mathcal{S}'$. Damit folgt, dass $\mathcal{S}'$ eine Überdeckung für E ist, also $I' \in$ SETCOVER gilt.

"$\Leftarrow$" Wir nehmen an, dass $I' \in$ SETCOVER gilt. Dann existiert eine Überdeckung $\mathcal{S}'$ für E mit $|\mathcal{S}'| \leq k$. Für jede Menge $S_i \in \mathcal{S}'$ fügen wir nun den entsprechenden Knoten v_i zu C hinzu. Offensichtlich gilt dann $|C| \leq k$. Es bleibt zu argumentieren, dass C eine Knotenüberdeckung für G ist. Sei dazu $e \in E$ eine beliebige Kante in G. Da $\mathcal{S}'$ eine Überdeckung für E ist, existiert eine Menge $S_i \in \mathcal{S}'$ mit $e \in S_i$. Gem. der Konstruktion von S_i gilt für den Knoten v_i, dass dieser inzident zu e ist, d.h. $e \cap \{v_i\} \neq \emptyset$. Gem. der Konstruktion von C gilt weiter $v_i \in C$. Damit ist C eine Knotenüberdeckung für E und es gilt $I \in$ VERTEXCOVER.

Aufgabe 2 : Separationen von P, NP und coNP

Sei $\text{coNP} = \left\{\overline{L} \mid L \in \text{NP}\right\}$ die Menge der Komplemente der Sprachen aus NP.

► Zeigen Sie:

(i) $\text{P} \subseteq \text{coNP}$ *und*

(ii) falls $\text{NP} \neq \text{coNP}$ *gilt, so gilt auch* $\text{P} \neq \text{NP}$.

Lösungsvorschlag:

(i) Wir wissen, dass $\text{P} \subseteq \text{NP}$ gilt und die Klasse P abgeschlossen bzgl. Komplementbildung ist, d.h. es gilt $\text{P} = \text{coP}$. Sei nun $\Pi \in \text{P}$. Dann gilt $\Pi \in \text{NP}$ und damit $\overline{\Pi} \in$ coNP. Damit folgt coP $\subseteq$ coNP und wegen P = coP gilt P $\subseteq$ coNP.

(ii) Allgemein gilt

$$A \Rightarrow B \quad \equiv \quad \neg A \vee B \quad \equiv \quad B \vee \neg A \quad \equiv \quad \neg B \Rightarrow \neg A,$$

so dass es genügt zu zeigen, dass P = NP also NP = coNP impliziert, um die ursprüngliche Aussage zu beweisen.

Wir nehmen nun an, dass P = NP gilt. Dann folgt mit (i) sofort NP $\subseteq$ coNP. Sei nun $\Pi \in$ coNP, dann gilt $\overline{\Pi} \in$ NP und wegen NP $\subseteq$ coNP auch $\overline{\Pi} \in$ coNP, also $\Pi \in$ NP. Damit folgt coNP $\subseteq$ NP und so insgesamt NP = coNP.

Aufgabe 3 : IndependentSet und SetPacking

Sei $G = (V, E)$ ein ungerichteter Graph. Eine Teilmenge $I \subseteq V$ der Knoten von G heißt *unabhängige Menge* in G, falls für alle Knoten $x, y \in I$ mit $x \neq y$ gilt $\{x, y\} \notin E$. Das entsprechende Entscheidungsproblem IndependentSet ist wie folgt definiert.

IndependentSet	
Gegeben:	Graph $G = (V, E)$ und eine natürliche Zahl $k \leq \lvert V \rvert$.
Frage:	Besitzt G eine unabhängige Menge der Größe mindestens k?

Weiter ist das Entscheidungsproblem SetPacking wie folgt definiert.

SetPacking	
Gegeben:	Endliche Menge U, eine Menge $S \subseteq 2^U$ von Teilmengen von U und eine natürliche Zahl $k \leq \lvert S \rvert$.
Frage:	Gibt es eine Teilmenge $S' \subseteq S$ mit $\lvert S' \rvert \geq k$, so dass alle Mengen in S' paarweise disjunkt sind?

(a) ▶ Entscheiden und begründen Sie, ob folgende Instanzen ja- oder nein-Instanzen für IndependentSet sind.

(i) $I_1 = (G_1, k_1)$ mit $G_1 = (V_1, E_1)$ und $k_1 = 4$ sowie

$$V_1 = \{1, 2, 3, 4, 5, 6\} \text{ und}$$
$$E_1 = \{\{1, 2\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \{4, 5\}, \{4, 6\}\},$$

(ii) $I_2 = (G_2, k_2)$ mit $G_2 = (V_2, E_2)$ und $k_2 = 3$ sowie

$$V_2 = \{1, 2, 3, 4, 5\} \text{ und}$$
$$E_2 = \{\{1, 2\}, \{1, 4\}, \{1, 5\}, \{2, 3\}, \{2, 5\}, \{3, 4\}, \{3, 5\}, \{4, 5\}\}.$$

(b) ► Entscheiden und begründen Sie, ob folgende Instanzen JA- oder NEIN-Instanzen für SETPACKING sind.

(i) $I_1 = (U_1, S_1, k_1)$ mit $U_1 = \{1, \dots, 11\}$,

$S_1 = \{\{1, 7, 11\}, \{3, 6, 9\}, \{2, 5, 11\}, \{2, 4, 6, 8, 10\}, \{1, 2, 5, 7\}, \{2, 5, 8\}, \{3, 4, 11\}\}$

und $k_1 = 3$.

(ii) $I_2 = (U_1, S_1, k_2)$ mit $k_2 = 4$.

(c) ► Zeigen Sie, dass SETPACKING $\in$ NP gilt.

(d) ► Sei I_1 eine INDEPENDENTSET Instanz. Geben Sie eine bzgl. der Eingabe in polynomieller Zeit berechenbare Funktion f an, so dass $f(I_1)$ eine SETPACKING Instanz ist und

$$I_1 \in \text{INDEPENDENTSET} \iff f(I_1) \in \text{SETPACKING}$$

gilt. Beweisen Sie die obige Äquivalenz und erläutern Sie warum f in polynomieller Zeit berechenbar ist.

(e) ► Wenden Sie f auf die Instanzen aus (a) an und entscheiden Sie, ob die Ergebnisse JA- oder NEIN-Instanzen für SETPACKING sind.

Lösungsvorschlag:

(a) (i) I_1 ist eine JA-Instanz, da bspw. $I = \{1, 3, 5, 6\}$ eine unabhängige Menge in G_1 mit mindestens 4 Knoten ist.

(ii) I_2 ist eine NEIN-Instanz, da es keine unabhängige Menge mit mehr als 2 Knoten in G_2 gibt.

(b) (i) I_1 ist eine JA-Instanz. Wir wählen die Teilmengen $\{1, 7, 11\}, \{3, 6, 9\}$ und $\{2, 5, 8\}$ zu S' hinzu. Dann gilt $|S'| \geq 3$ und alle Mengen in S' sind paarweise disjunkt.

(ii) I_2 ist eine NEIN-Instanz, da es keine 4 paarweise disjunkten Mengen in S_1 gibt.

(c) Wir wollen zeigen, dass SETPACKING $\in$ NP gilt. Dazu genügt es ein nichtdeterministisches Verfahren anzugeben, dass SETPACKING in polynomieller Zeit löst. Wir verfahren wie folgt: Im ersten Schritt "raten" wir in Zeit in $\mathcal{O}(k)$ eine Teilmenge $S' \subseteq S$ der Größe k. Anschließend überprüfen wir deterministisch in Zeit in $\mathcal{O}(nk^2)$ ob alle k Elemente in S' paarweise disjunkt sind. Damit benötigen wir insgesamt eine Zeit in $\mathcal{O}(k+nk^2)$ und es folgt, dass SETPACKING $\in$ NP gilt.

(d) Wir beschreiben zuerst die Funktionsweise der Funktion f. Sei dazu $I = (G, k)$ eine INDEPENDENTSET Instanz mit $G = (V, E)$ und $V = \{v_1, \dots, v_n\}$. Dann konstruiert f aus I wie folgt eine SETPACKING Instanz $I' = (U, S, k')$: Es gilt

$U = E$, $k' = k$ und $S = \{S_i \mid v_i \in V\}$ mit

$$S_i = \{e \in E \mid e \cap \{v_i\} \neq \emptyset\}.$$

Offensichtlich kann $f(I)$ in Zeit polynomiell in $|I|$ berechnet werden: Für U und k müssen wir nur Kopieren. Für S konstruieren wir n Mal eine Menge S_i, wobei wir jedes Mal alle $|E|$ Kanten durchlaufen. Damit brauchen wir insgesamt eine Zeit in $\mathcal{O}(|E| + |V||E|) \subseteq \mathcal{O}(|V|^3) \subseteq \mathcal{O}(|I|^2)$.

Wir argumentieren für jede Richtung der Äquivalenz einzeln:

"$\Rightarrow$" Sei I eine JA-Instanz für INDEPENDENTSET. Dann existiert eine unabhängige Menge $W \subseteq V$ in G mit $|W| \geq k$. Füge nun zu S' genau die Mengen S_i aus S hinzu, für die $v_i \in W$ gilt. Offensichtlich gilt dann $|S'| \geq k$. Angenommen es gibt S_i, S_j in S' mit $S_i \cap S_j \neq \emptyset$. Dann existiert eine Kante $\{v_i, v_j\} \in E$. Das ist aber ein Widerspruch dazu, dass $v_i, v_j \in W$ gelten und W eine unabhängige Menge ist. Folglich müssen alle $S_i \in S'$ paarweise disjunkt sein. Damit gilt also, dass $f(I)$ auch eine JA-Instanz für SETPACKING ist.

"$\Leftarrow$" Sei $I = (G, k)$ eine INDEPENDENTSET Instanz und $f(I) = (U, S, k)$ eine JA-Instanz für SETPACKING. Dann existiert eine Menge $S' \subseteq S$ mit $|S'| \geq k$ und alle Elemente S_i in S' sind paarweise disjunkt. Für jedes Element $S_i \in S'$ fügen wir v_i zu W hinzu. Offensichtlich gilt dann $W \subseteq V$ sowie $|W| \geq k$. Für $v_i, v_j \in W$ mit $v_i \neq v_j$ muss $\{v_i, v_j\} \notin E$ gelten, da S_i und S_j paarweise disjunkt sind. Demgemäß ist W eine unabhängige Menge in G und I folglich eine JA-Instanz.

(e) (i) Wir wenden f aus (d) auf $I_1 = (G_1, k_1)$ aus (a) an und erhalten $f(I_1) = (U_1, S_1, k'_1)$ mit $U_1 = E_1$, $k'_1 = k_1$ sowie

$$\begin{aligned} S_1 = \{&\{\{1,4\},\{1,2\}\},\{\{1,2\},\{2,3\},\{2,4\}\},\{\{2,3\},\{3,4\}\},\\ &\{\{1,4\},\{2,4\},\{3,4\},\{4,5\},\{4,6\}\},\{\{4,5\}\},\{\{4,6\}\}\}. \end{aligned}$$

Da I_1 gem. (a) eine JA-Instanz ist, folgt mit (d), dass auch $f(I_1)$ eine JA-Instanz ist. Um dies zu überprüfen, betrachten wir

$$S'_1 = \{\{\{1,4\},\{1,2\}\},\{\{2,3\},\{3,4\}\},\{\{4,5\}\},\{\{4,6\}\}\} \subseteq S_1$$

und stellen fest, dass $|S'_1| \geq 4 = k'_1$ gilt und alle Elemente in S'_1 paarweise disjunkt sind, $f(I_1)$ also eine JA-Instanz ist.

(ii) Wir wenden f aus (d) auf $I_2 = (G_2, k_2)$ aus (a) an und erhalten $f(I_2) = (U_2, S_2, k'_2)$ mit $U_2 = E_2$, $k'_2 = k_2$ sowie

$$\begin{aligned} S_2 = \{&\{\{1,2\},\{1,4\},\{1,5\}\},\{\{1,2\},\{2,3\},\{2,5\}\},\{\{2,3\},\{3,4\},\{3,5\}\},\\ &\{\{1,4\},\{3,4\},\{4,5\}\},\{\{1,5\},\{2,5\},\{3,5\},\{4,5\}\}\}. \end{aligned}$$

Da I_2 gem. (a) eine NEIN-Instanz ist, folgt mit (d), dass auch $f(I_2)$ eine NEIN-Instanz ist. Überprüfen ergibt, dass es tatsächlich keine Teilmenge $S_2' \subseteq S_2$ mit mehr als 2 Element gibt, für die alle Elemente paarweise disjunkt sind.

Aufgabe 4 : PARTITION und MAXCUT

Die Entscheidungsprobleme PARTITION und WEIGHTED MAXCUT sind wie folgt definiert:

	PARTITION
Gegeben:	Eine Liste $A = (a_1, \dots, a_n)$ positiver ganzer Zahlen mit $\sum_{i=1}^{n} a_i$ gerade.
Frage:	Kann man A so in zwei disjunkte Listen A_1, A_2 partitionieren, dass $\sum_{a_i \in A_1} a_i = \sum_{a_j \in A_2} a_j$ gilt?

	WEIGHTED MAXCUT
Gegeben:	Ein ungerichteter Graph $G = (V, E)$, eine Gewichtsfunktion $w\colon E \to \mathbb{N}_{\geq 0}$ und eine Schranke $k \in \mathbb{N}_{\geq 0}$.
Frage:	Gibt es eine Partition der Knotenmenge V in zwei disjunkte Teilmengen V_1, V_2, so dass $\sum_{\{u,v\} \in \{\{u,v\} \in E \mid u \in V_1, v \in V_2\}} w(\{u,v\}) \geq k$ gilt?

(a) ▶ Entscheiden und begründen Sie, ob folgende Instanzen JA- oder NEIN-Instanzen für PARTITION sind:

(i) $I_1 = (14, 5, 2, 6, 5, 8, 13, 7)$ und

(ii) $I_2 = (3, 12, 22, 17, 6)$.

(b) ▶ Zeigen Sie, dass PARTITION $\in$ NP gilt.

(c) Betrachten Sie die Funktion g, die eine PARTITION Instanz I auf eine WEIGHTED MAXCUT Instanz $g(I)$ wie folgt abbildet. Sei $I = \{a_1, \dots, a_n\}$ eine PARTITION Instanz. Dann gilt $g(I) = (G, w, k)$ mit $G = (V, E)$, $w\colon E \to \mathbb{N}_{\geq 0}$ und

$$k = \left(\frac{\sum_{i=1}^{n} a_i}{2} \right)^2,$$

wobei $V = \{v_1, \dots, v_n\}$, $E = V \times V \setminus \{\{v_i, v_i\} \mid 1 \leq i \leq n\}$ und $w(\{v_i, v_j\}) = a_i a_j$ sei.

▶ Zeigen Sie, dass für eine PARTITION Instanz I gilt

$$I \in \text{PARTITION} \iff g(I) \in \text{WEIGHTED MAXCUT}$$

und argumentieren Sie, weshalb g in Polynomialzeit bzgl. $|I|$ berechenbar ist.

(d) ► Wenden Sie g auf die Instanzen aus (a) an und entscheiden Sie, ob die resultierenden Instanzen JA- oder NEIN-Instanzen für WEIGHTED MAXCUT sind.

Lösungsvorschlag:

(a) (i) Bei I_1 handelt es sich um eine JA-Instanz, da wir die Liste I_1 in die Teillisten $A_1 = (14, 5, 6, 5)$ und $A_2 = (2, 8, 13, 7)$ mit jeweils Summe 30 partitionieren können.

(ii) Bei I_2 handelt es sich um eine NEIN-Instanz, da die Summe aller Element 60 beträgt, folglich die Teillisten also eine Summe von 30 aufweisen müssten, die Summe 30 aber nicht durch die Elemente aus I_2 darstellbar ist.

(b) Um zu zeigen, dass PARTITION in NP ist, geben wir einen nichtdeterministischen Algorithmus zum Lösen von PARTITION-Instanzen in polynomieller Zeit an. Sei dazu $I = (a_1, \dots, a_n)$ eine PARTITION-Instanz. Wir “raten” nichtdeterministisch eine Partition von I in zwei Teillisten A_1 und A_2 in Zeit in $\mathcal{O}(n)$. Anschließend überprüfen wir deterministisch, ob die Teillisten der geratenen Aufteilung die gleichen Summen aufweisen. Dieses Überprüfen geht ebenfalls in polynomieller Zeit, sodass PARTITION $\in$ NP folgt.

(c) Wir argumentieren zu erst, dass g in Zeit polynomiell bzgl. der Eingabe berechenbar ist. Den Wert k sowie die Gewichte aller Kanten können wir in polynomieller Zeit berechnen. Der Graph $G = (V, E)$ hat n Knoten und ist vollständig, hat also $n(n-1)/2$ Kanten. Beide Größen sind polynomiell in der Eingabegröße, so dass sich die Funktion g insgesamt in polynomieller Zeit berechnen lässt.

Für die Äquivalenz betrachten wir beide Richtungen einzeln.

“$\Rightarrow$” Sei $I = \{a_1, \dots, a_n\}$ eine JA-Instanz für PARTITION. Dann lassen sich die Elemente aus I in zwei disjunkte Teillisten A_1, A_2 aufteilen, so dass die Summen der Teillisten gleich sind und

$$\sum_{a_i \in A_1} a_i = \sum_{a_j \in A_2} a_j = \frac{\sum_{i=1}^n a_i}{2}$$

gilt. Für die mittels g konstruierte WEIGHTED MAXCUT Instanz $g(I) = (G, w, k)$ mit $G = (V, E)$ gilt dann, dass jeder Knoten in V einem Element in I entspricht. Wir setzen nun $V_1 = \{v_i \mid a_i \in A_1\}$ und $V_2 = \{v_i \mid a_i \in A_2\}$. Dann gilt

$$\sum_{\{v_i,v_j\} \in \{\{u,v\} \in E \mid u \in V_1, v \in V_2\}} w(\{v_i, v_j\}) = \sum_{\{v_i,v_j\} \in \{\{u,v\} \in E \mid u \in V_1, v \in V_2\}} a_i a_j.$$

Da G ein vollständiger Graph ist, existieren für jeden Knoten $v_i \in V_1$ Kanten zu allen Knoten in V_2. Folglich können wir die obige Summe weiter vereinfachen zu

$$\sum_{v_i \in V_1} \left[a_i \sum_{v_j \in V_2} a_j \right] = \sum_{v_i \in V_1} \left[a_i \frac{\sum_{j=1}^n a_j}{2} \right] = \frac{\sum_{i=1}^n a_i}{2} \frac{\sum_{i=1}^n a_i}{2} = k.$$

Damit folgt, dass $g(I)$ eine JA-Instanz für WEIGHTED MAXCUT ist.

"$\Leftarrow$" Sei nun $g(I) = (G, w, k)$ eine JA-Instanz für WEIGHTED MAXCUT. Dann existiert eine Partition von V in zwei Teillisten V_1, V_2, so dass

$$\sum_{\{v_i, v_j\} \in \{\{u,v\} \in E \mid u \in V_1, v \in V_2\}} w(\{v_i, v_j\}) \geq k$$

gilt. Da G ein vollständiger Graph ist, können wir auch schreiben

$$\sum_{v_i \in V_1, v_j \in V_2} a_i a_j \geq \left(\frac{\sum_{i=1}^n a_i}{2} \right)^2$$

und dies weiter umformen zu

$$\left(\sum_{v_i \in V_1} a_i \right) \left(\sum_{v_j \in V_2} a_j \right) \geq \left(\frac{\sum_{i=1}^n a_i}{2} \right)^2.$$

Weiterhin gilt $\sum_{i=1}^n a_i = \sum_{v_i \in V_1} a_i + \sum_{v_j \in V_2} a_j$, so dass wir schreiben können

$$\begin{aligned}
\left(\sum_{v_i \in V_1} a_i \right) \left(\sum_{v_j \in V_2} a_j \right) &\geq \left(\frac{\sum_{v_i \in V_1} a_i + \sum_{v_j \in V_2} a_j}{2} \right)^2 \\
\Rightarrow 4 \left(\sum_{v_i \in V_1} a_i \right) \left(\sum_{v_j \in V_2} a_j \right) &\geq \left(\sum_{v_i \in V_1} a_i + \sum_{v_j \in V_2} a_j \right)^2 \\
\Rightarrow 4 \left(\sum_{v_i \in V_1} a_i \right) \left(\sum_{v_j \in V_2} a_j \right) &\geq \left(\sum_{v_i \in V_1} a_i \right)^2 + 2 \left(\sum_{v_i \in V_1} a_i \right) \left(\sum_{v_j \in V_2} a_j \right) + \left(\sum_{v_j \in V_2} a_j \right)^2 \\
\Rightarrow \qquad 0 &\geq \left(\sum_{v_i \in V_1} a_i \right)^2 - 2 \left(\sum_{v_i \in V_1} a_i \right) \left(\sum_{v_j \in V_2} a_j \right) + \left(\sum_{v_j \in V_2} a_j \right)^2 \\
\Rightarrow \qquad 0 &\geq \left(\sum_{v_i \in V_1} a_i - \sum_{v_j \in V_2} a_2 \right)^2.
\end{aligned}$$

Daraus ergibt sich sofort, dass $\sum_{v_i \in V_1} a_i = \sum_{v_j \in V_2} a_2$ gelten muss und somit $A_1 = \{a_i \mid v_i \in V_1\}$ und $A_2 = \{a_j \mid v_j \in V_2\}$ eine Lösung für I sind, so dass I eine JA-Instanz für PARTITION ist.

(d) Der Graph $G = (V, E)$, den g konstruiert, hängt nur von der Anzahl von Objekten in der ursprünglichen PARTITION Instanz $I = \{a_1, \ldots, a_n\}$ ab. g konstruiert immer einen vollständigen Graphen mit n Knoten, in Zeichen K_n. Daher genügt es für beide Instanzen jeweils die Gewichte der Kanten sowie k zu berechnen.

(i) Die Instanz I_1 hat 8 Elemente, so dass g den K_8 konstruiert. Die zugehörige Gewichtsfunktion für $g(I_1)$ ist in der folgenden Tabelle aufgetragen. Dabei ist nur die obere Hälfte der Tabelle ausgefüllt, da die Kanten ungerichtet sind und die Tabelle somit symmetrisch ist.

$w(\{v_i, v_j\})$	2	3	4	5	6	7	8
1	70	28	84	70	112	182	98
2		10	30	25	40	65	35
3			12	10	16	26	14
4				30	48	78	42
5					40	65	35
6						104	56
7							91

Für k ergibt sich der Wert $k = (60/2)^2 = 900$. Gemäß (a) ist I_1 eine JA-Instanz, so dass mit (c) folgt, dass $g(I_1)$ auch eine JA-Instanz sein muss. Wählt man die Knoten $V_1 = \{v_1, v_2, v_4, v_5\}$, so ergibt sich als Summe der Kantengewichte für Kanten zwischen V_1 und $V \setminus V_1$ im K_8:

$$\begin{aligned} & 28 + 112 + 182 + 98 + 10 + 40 + 65 + 35 + 12 \\ + & 10 + 48 + 78 + 42 + 40 + 65 + 35 \\ = & \left(\frac{60}{2}\right)^2. \end{aligned}$$

Damit wird unsere vorherige Überlegung bestätigt und $g(I_1)$ ist eine JA-Instanz.

(ii) Die Instanz I_2 hat 6 Elemente, so dass g den K_6 konstruiert. Die zugehörige Gewichtsfunktion für $g(I_2)$ ist in der folgenden Tabelle aufgetragen.

$w(\{v_i, v_j\})$	2	3	4	5	6
1	6	20	8	14	20
2		30	12	21	30
3			40	70	100
4				28	40
5					70

Für k ergibt sich der Wert $k = (36/2)^2$. Gemäß (a) ist I_2 eine NEIN-Instanz, so dass mit (c) folgt, dass $g(I_2)$ auch eine NEIN-Instanz sein muss. Durchprobieren zeigt, dass es tatsächlich keine Partition von V in V_1 und V_2 gibt, so dass $g(I_2)$ eine JA-Instanz wäre.