Graphen und diskrete Optimierung

im Bachelorstudiengang 'Informatik und Nachhaltigkeit'

Färbungen, Cliquen und Planarität

Marie Schmidt

17.05.2023

Uni Würzburg Instagram 'Takeover'

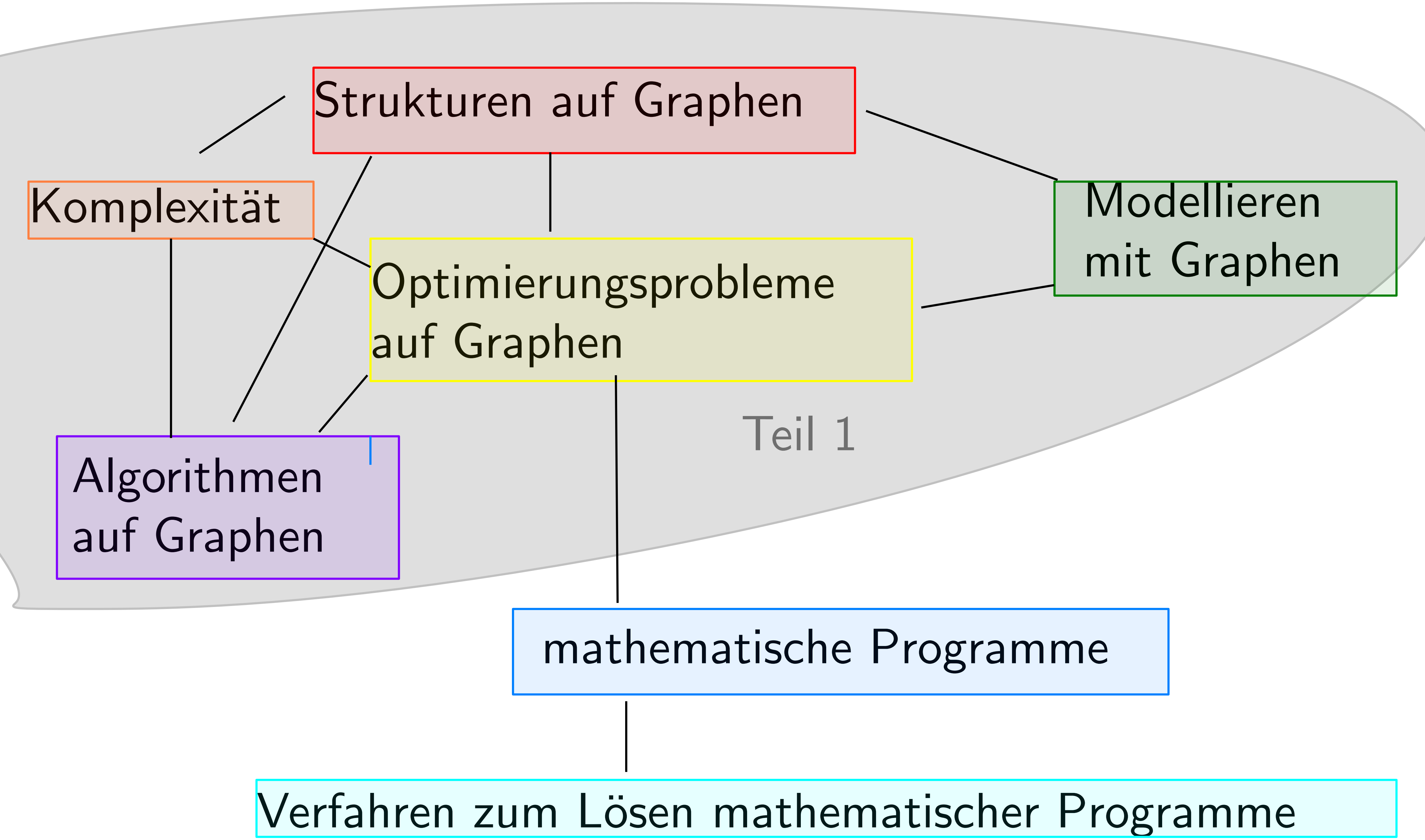
Pressestelle sucht InNa Student*in für TakeOver vom uniwuerzburg Account

Interesse?
Gerne bei mir oder direkt bei Dominika Heublein von der Pressestelle wenden (dominika.heublein@uni-wuerzburg.de)

Übung diese Woche

Übung um 14:15 Uhr fällt voraussichtlich aus

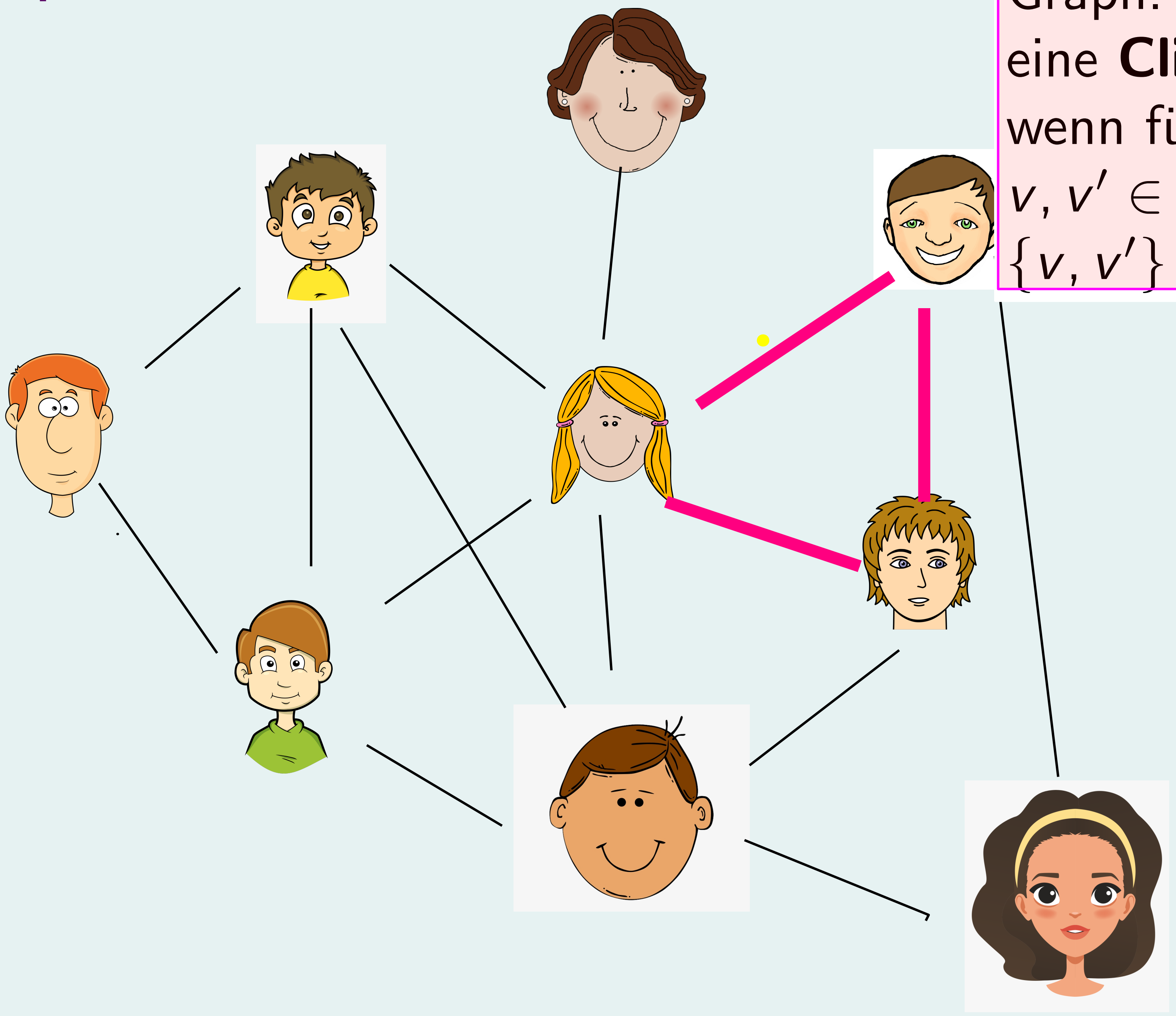
Worum soll er hier gehen?



Vorlesungsübersicht

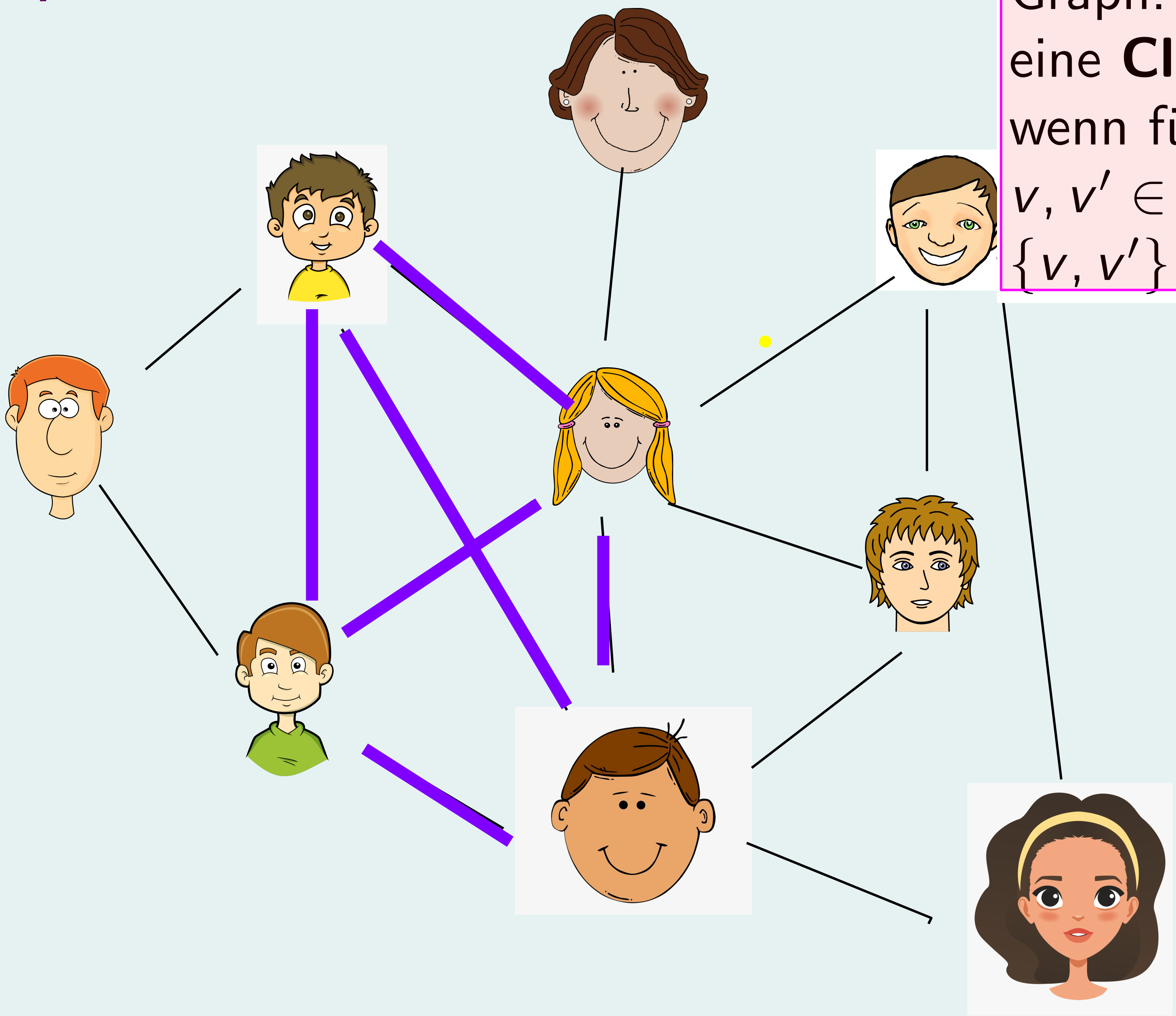
- Einführung: Modellieren auf Graphen ✓
- Teil 1: Graphen und Algorithmen auf Graphen
  - Graphen modellieren räumliche Zusammenhänge
    - Wege: Dijkstra ✓, Bellmann-Ford ✓, multikriterielle Wege ✓
    - Touren ✓
    - Flüsse: maximale Flüsse ✓, Strömungen ✓, b-Flüsse, kostenminimale Flüsse ✓
  - Graphen modellieren Beziehungen
    - Matchings ✓
    - Eckenüberdeckungen ✓
    - unabhängige Mengen
    - Cliquen
    - Färbungsprobleme
    - planare Graphen
  - Algorithmen und (Problem-)Komplexität
- Teil 2: Lineare und ganzzahlige Optimierung auf Graphen
  - ...

Cliquen



Sei $G = (V, E)$ ein Graph. $C \subset V$ ist eine **Clique** in G, wenn für alle $v, v' \in V$, $\{v, v'\} \in E$.

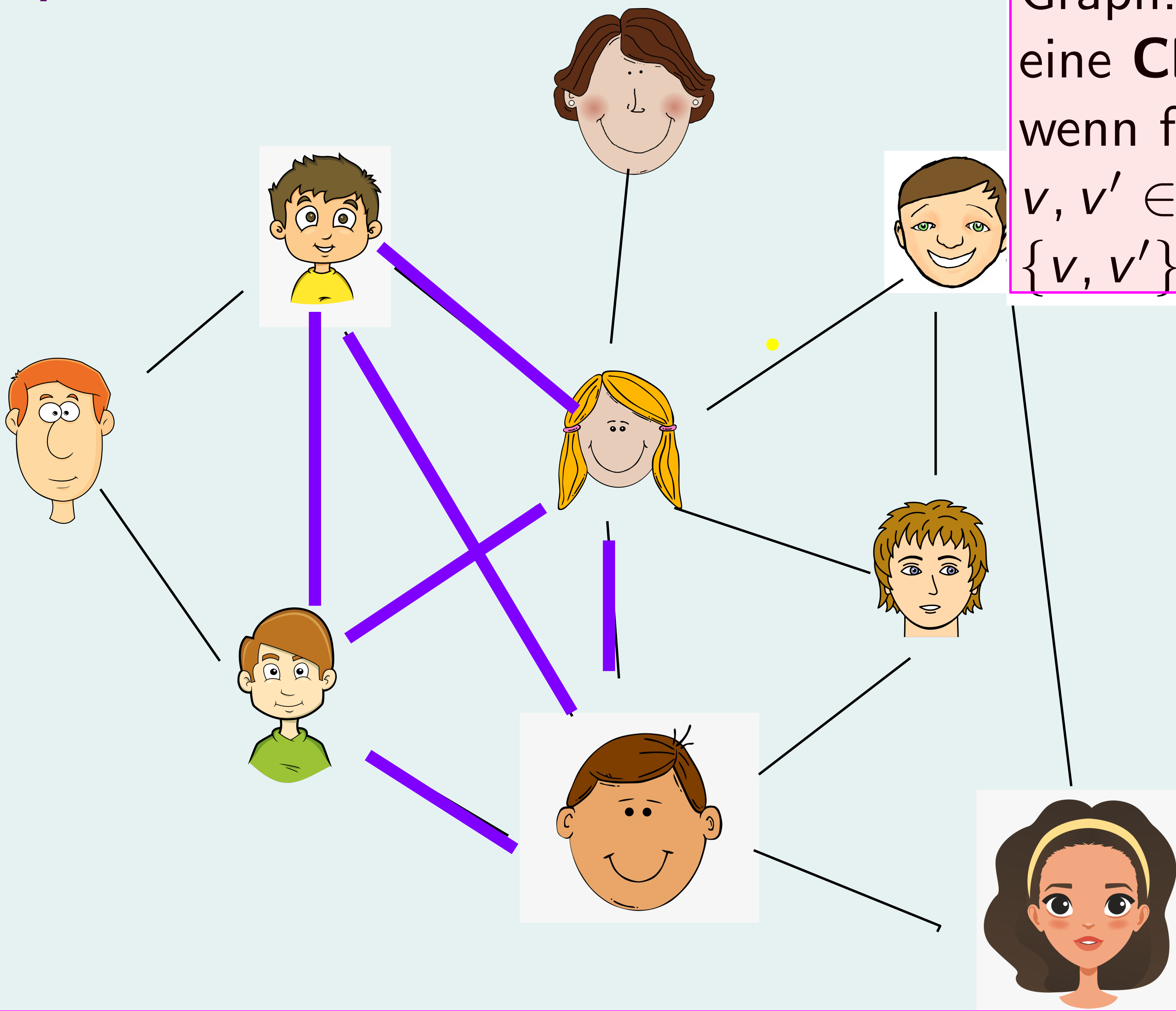
Cliquen



Sei $G = (V, E)$ ein Graph. $C \subset V$ ist eine **Clique** in G, wenn für alle $v, v' \in V$, $\{v, v'\} \in E$.

Cliquen

Sei $G = (V, E)$ ein Graph. $C \subset V$ ist eine **Clique** in G, wenn für alle $v, v' \in V$, $\{v, v'\} \in E$.

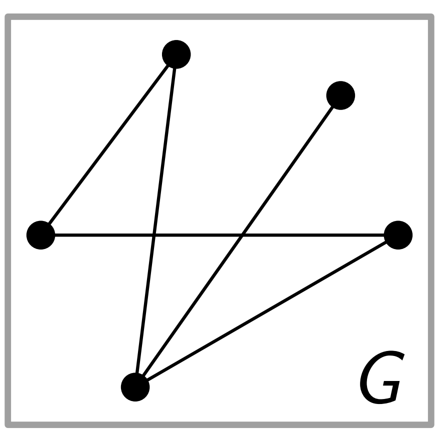


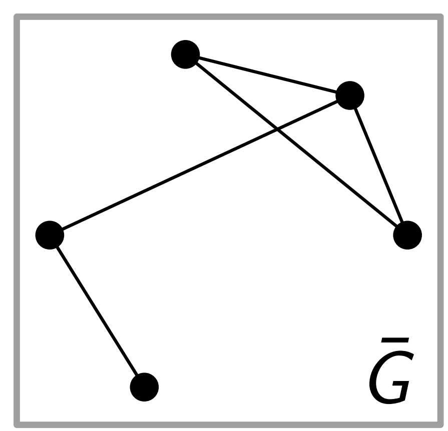
$\omega(G) = \max\{|C| : C \text{ ist Clique in } G\}$ heißt *Cliquenzahl* von G.

Zusammenhang Clique und unabhängige Menge

Zusammenhang Clique und unabhängige Menge

Sei $G = (V, E)$ ein Graph.
Dann ist $\bar{G} = (V, \bar{E})$ mit $\bar{E} := \{\{u, v\} \in V \times V, u \neq v\} \setminus E$ der *Komplementgraph* von G.





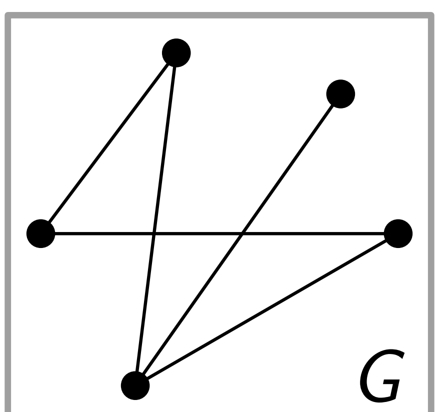
Zusammenhang Clique und unabhängige Menge

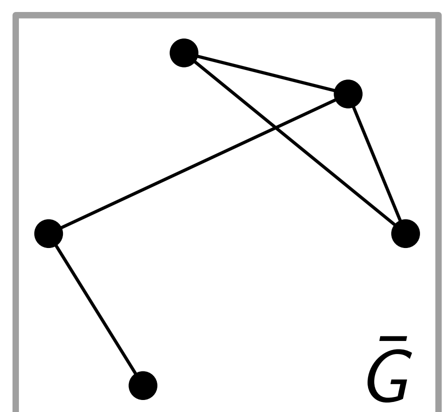
Sei $G = (V, E)$ ein Graph.
Dann ist $\bar{G} = (V, \bar{E})$ mit $\bar{E} := \{\{u, v\} \in V \times V, u \neq v\} \setminus E$ der *Komplementgraph* von G.

Es gilt

1. $E \cup \bar{E} =$ $\neq v\}$
2. $\bar{\bar{G}} =$
3. S Clique in $G \Leftrightarrow$
4. $\omega(G) =$

Cliquenzahl





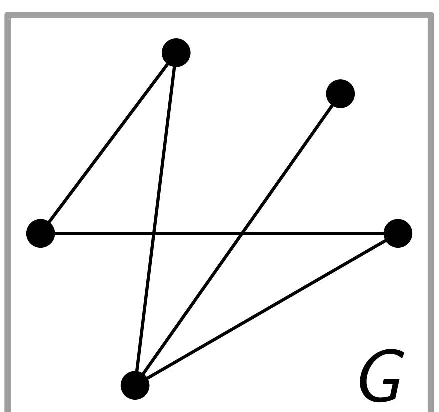
Zusammenhang Clique und unabhängige Menge

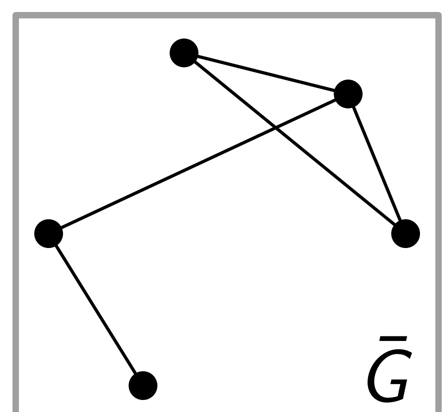
Sei $G = (V, E)$ ein Graph.
Dann ist $\bar{G} = (V, \bar{E})$ mit $\bar{E} := \{\{u, v\} \in V \times V, u \neq v\} \setminus E$ der *Komplementgraph* von G.

Es gilt

1. $E \cup \bar{E} = \{\{u, v\} \in V \times V : u \neq v\}$
2. $\bar{\bar{G}} =$
3. S Clique in $G \Leftrightarrow$
4. $\omega(G) =$

Cliquenzahl





Zusammenhang Clique und unabhängige Menge

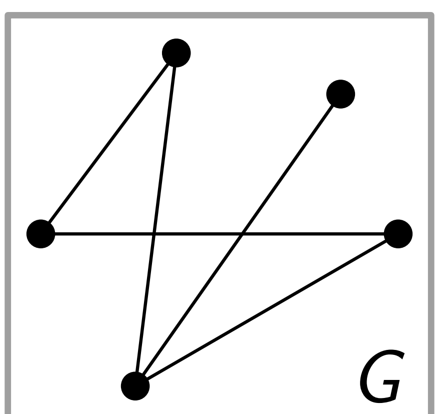
Sei $G = (V, E)$ ein Graph.

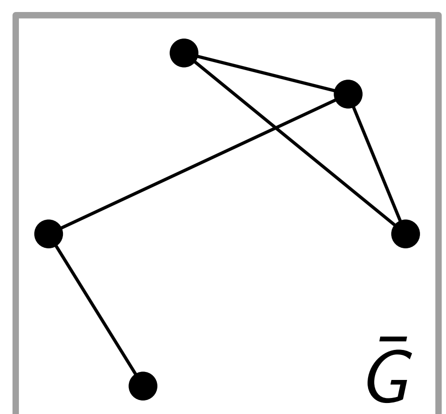
Dann ist $\bar{G} = (V, \bar{E})$ mit $\bar{E} := \{\{u, v\} \in V \times V, u \neq v\} \setminus E$ der *Komplementgraph* von G.

Es gilt

1. $E \cup \bar{E} = \{\{u, v\} \in V \times V : u \neq v\}$
2. $\bar{\bar{G}} = G$
3. S Clique in $G \Leftrightarrow$
4. $\omega(G) =$

Cliquenzahl





Zusammenhang Clique und unabhängige Menge

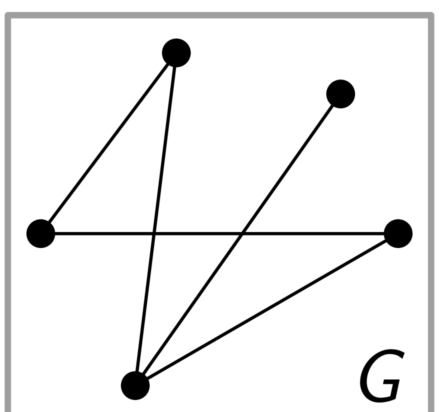
Sei $G = (V, E)$ ein Graph.

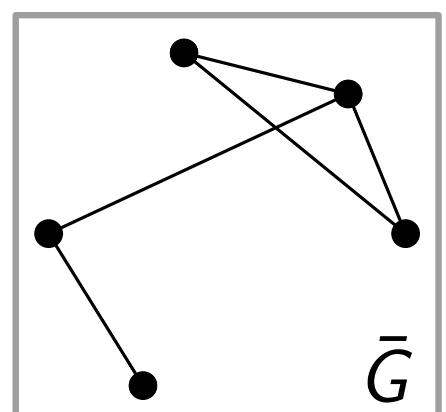
Dann ist $\bar{G} = (V, \bar{E})$ mit $\bar{E} := \{\{u, v\} \in V \times V, u \neq v\} \setminus E$ der *Komplementgraph* von G.

Es gilt

1. $E \cup \bar{E} = \{\{u, v\} \in V \times V : u \neq v\}$
2. $\bar{\bar{G}} = G$
3. S Clique in $G \Leftrightarrow S$ unabhängig in $\bar{G}$
4. $\omega(G) =$

Cliquenzahl





Zusammenhang Clique und unabhängige Menge

Sei $G = (V, E)$ ein Graph.

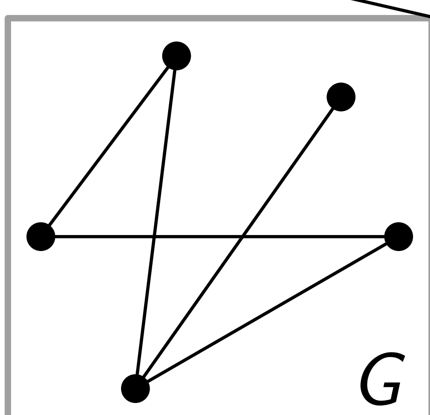
Dann ist $\bar{G} = (V, \bar{E})$ mit $\bar{E} := \{\{u, v\} \in V \times V, u \neq v\} \setminus E$ der *Komplementgraph* von G.

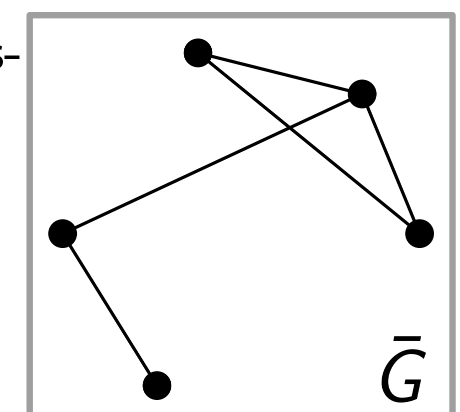
Es gilt

1. $E \cup \bar{E} = \{\{u, v\} \in V \times V : u \neq v\}$
2. $\bar{\bar{G}} = G$
3. S Clique in $G \Leftrightarrow S$ unabhängig in $\bar{G}$
4. $\omega(G) = \alpha(\bar{G})$ und $\alpha(G) = \omega(\bar{G})$

Cliquenzahl

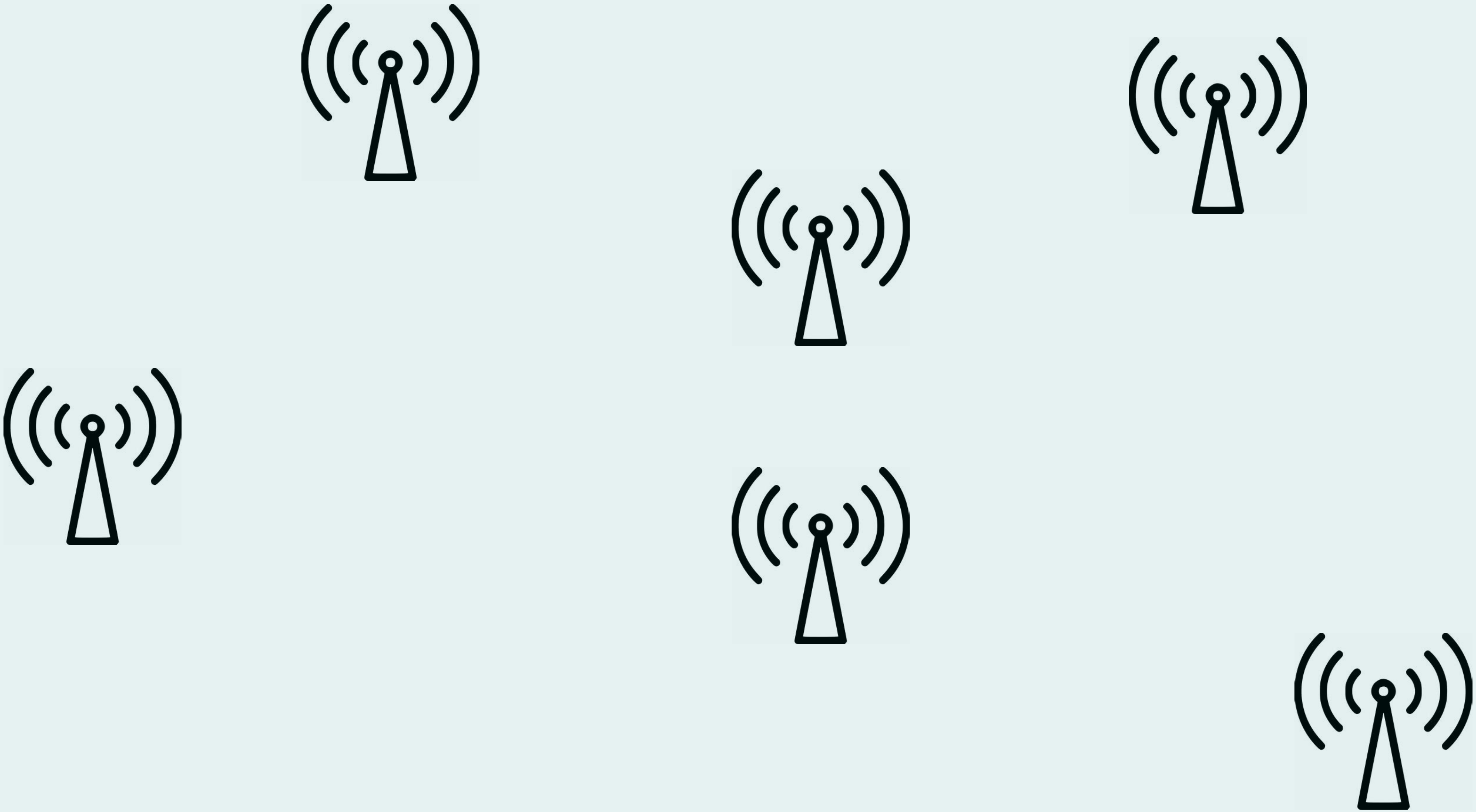
Unabhängigkeitszahl





Frequenzzuweisungen

Wieviele unterschiedliche Frequenzen brauche ich mindestens, um Interferenz zu vermeiden?

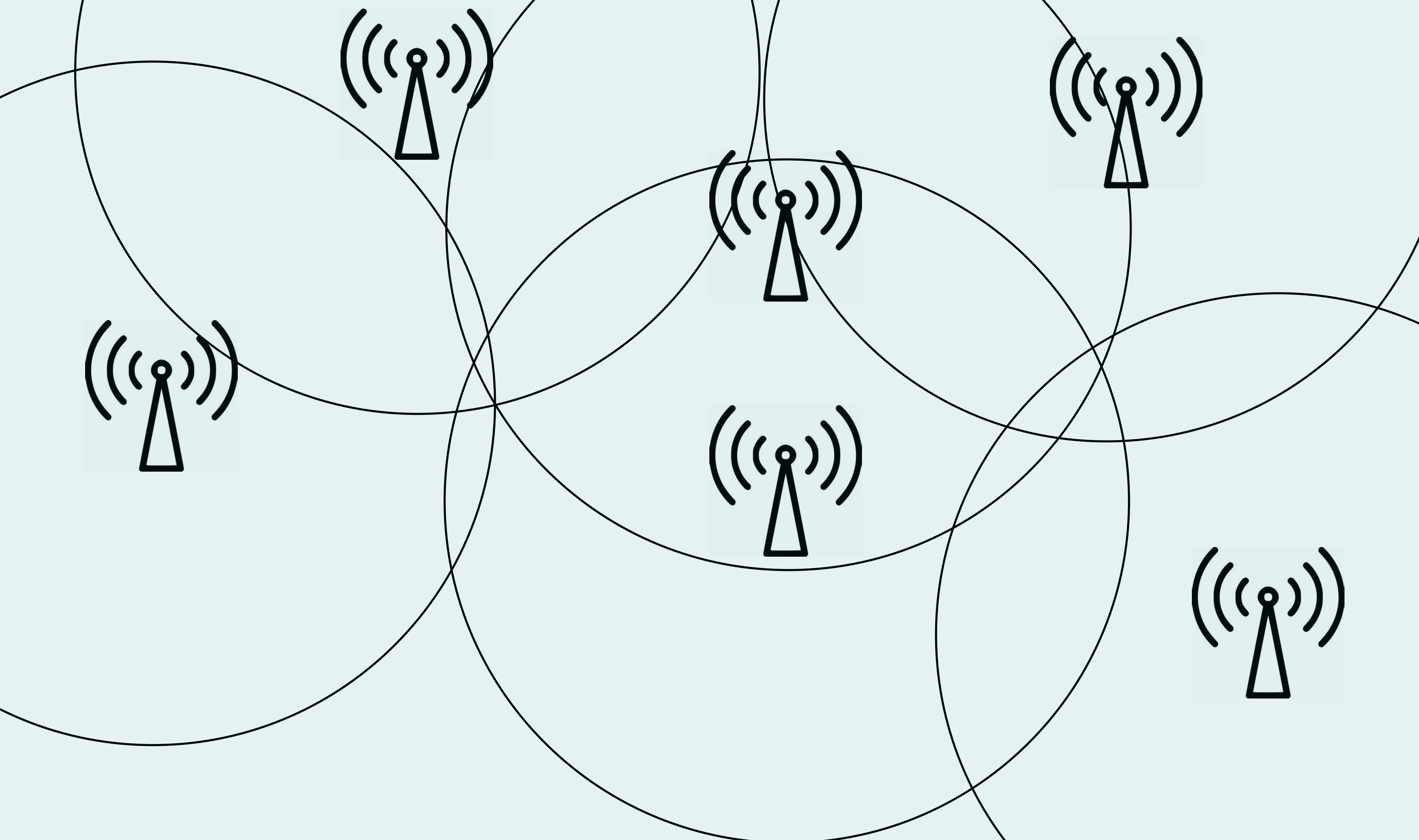


Frequenzzuweisungen

Wieviele unterschiedliche Frequenzen brauche ich mindestens, um Interferenz zu vermeiden?

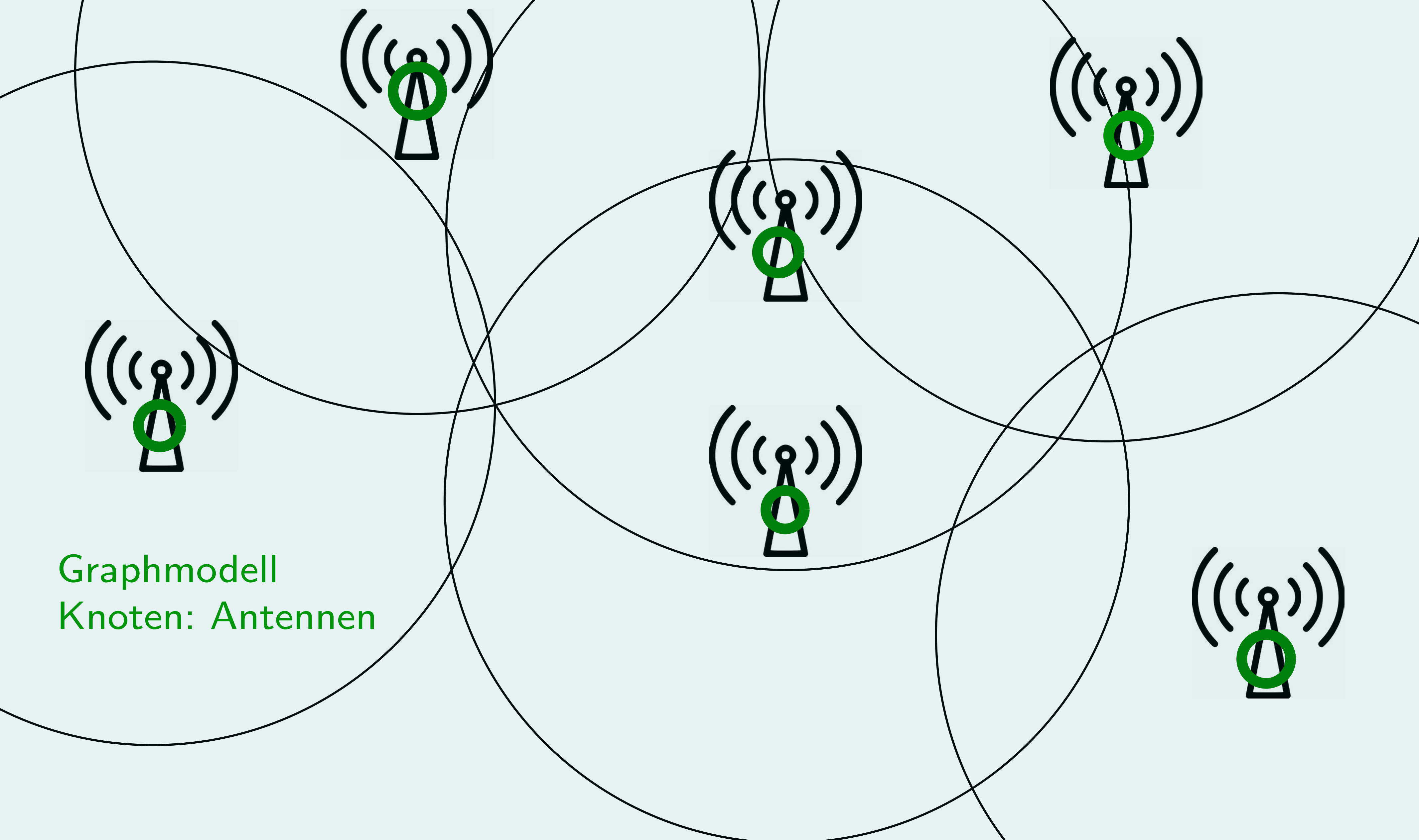
Frequenzzuweisungen

Wieviele unterschiedliche Frequenzen brauche ich mindestens, um Interferenz zu vermeiden?



Frequenzzuweisungen

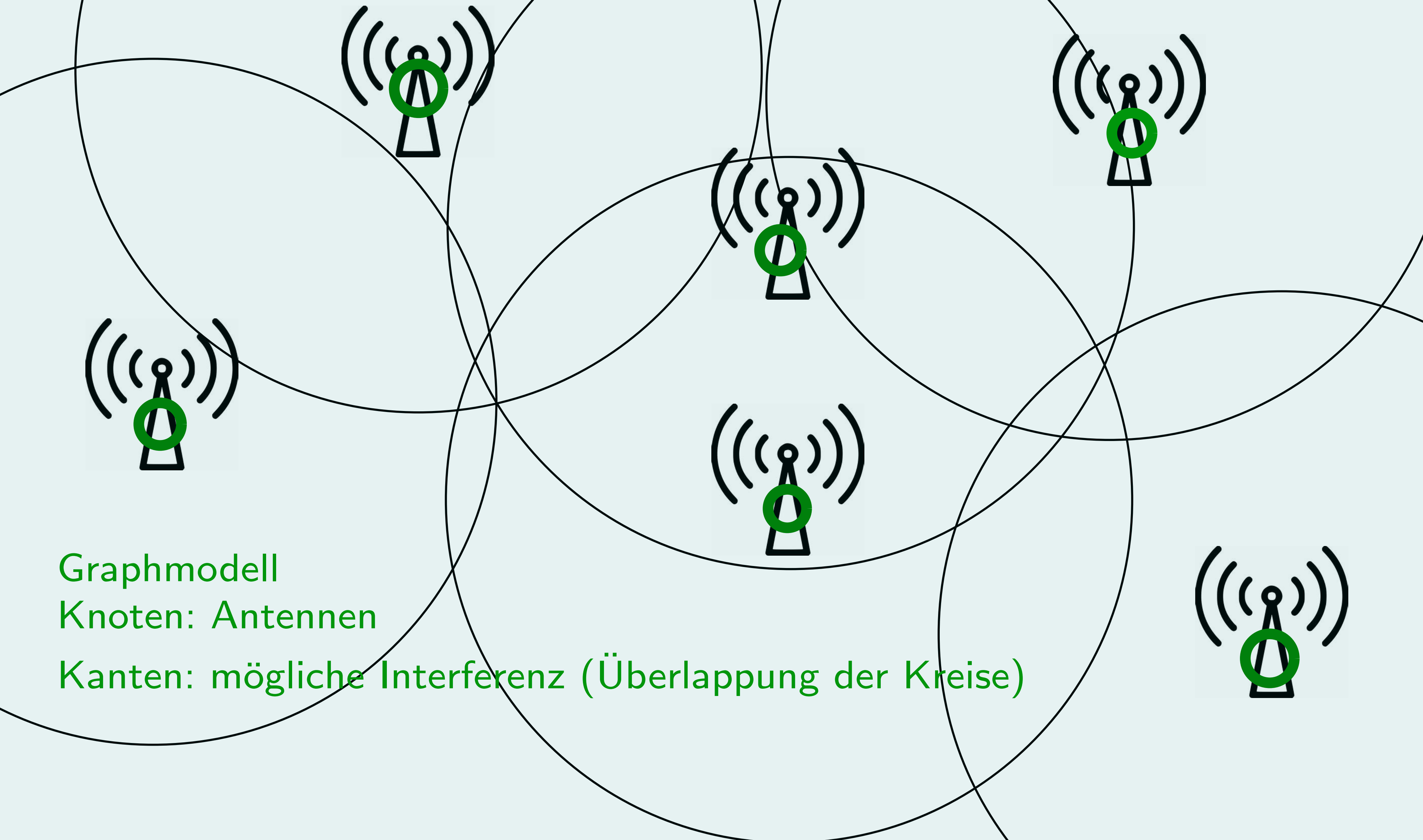
Wieviele unterschiedliche Frequenzen brauche ich mindestens, um Interferenz zu vermeiden?



Graphmodell
Knoten: Antennen

Frequenzzuweisungen

Wieviele unterschiedliche Frequenzen brauche ich mindestens, um Interferenz zu vermeiden?

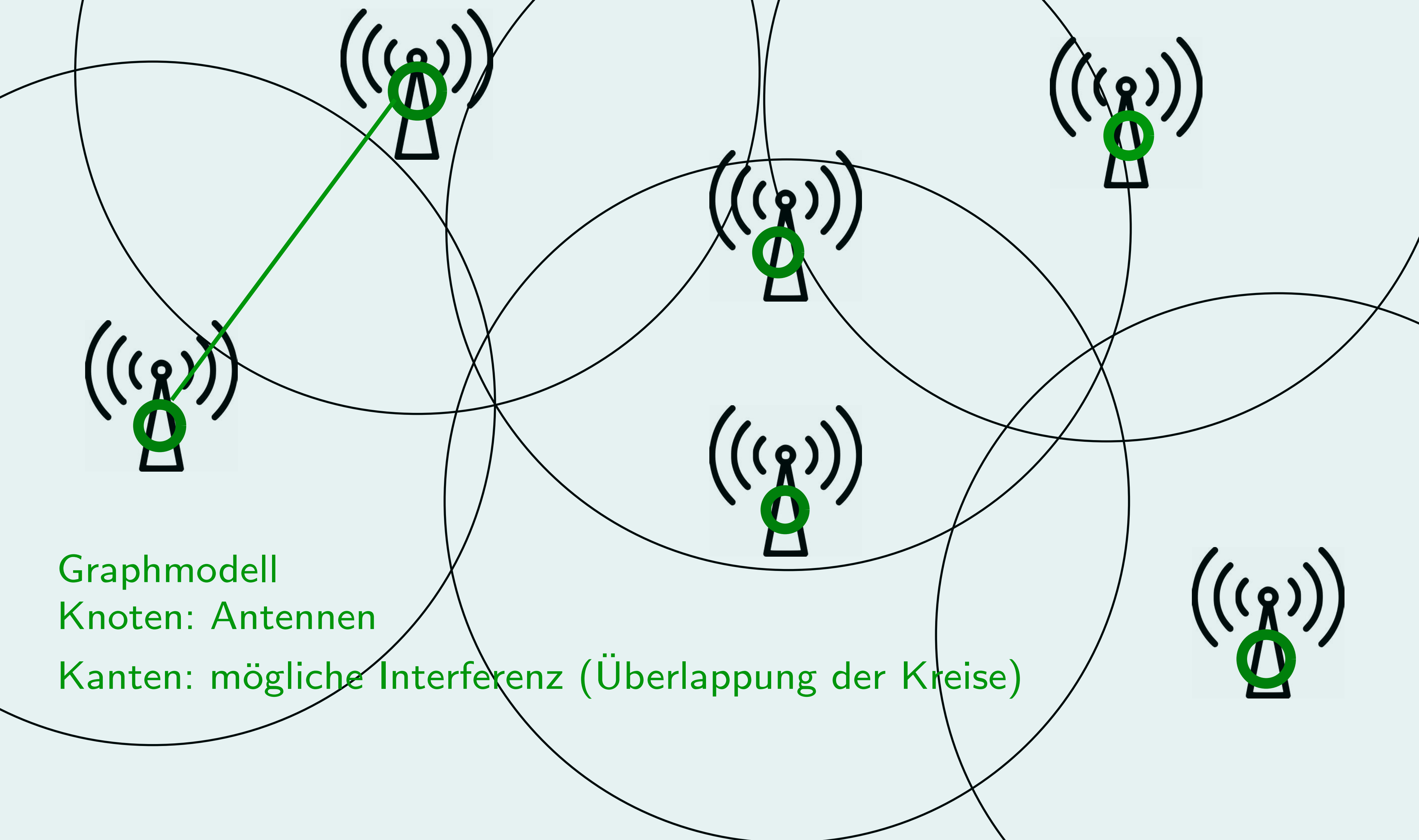


Graphmodell
Knoten: Antennen

Kanten: mögliche Interferenz (Überlappung der Kreise)

Frequenzzuweisungen

Wieviele unterschiedliche Frequenzen brauche ich mindestens, um Interferenz zu vermeiden?



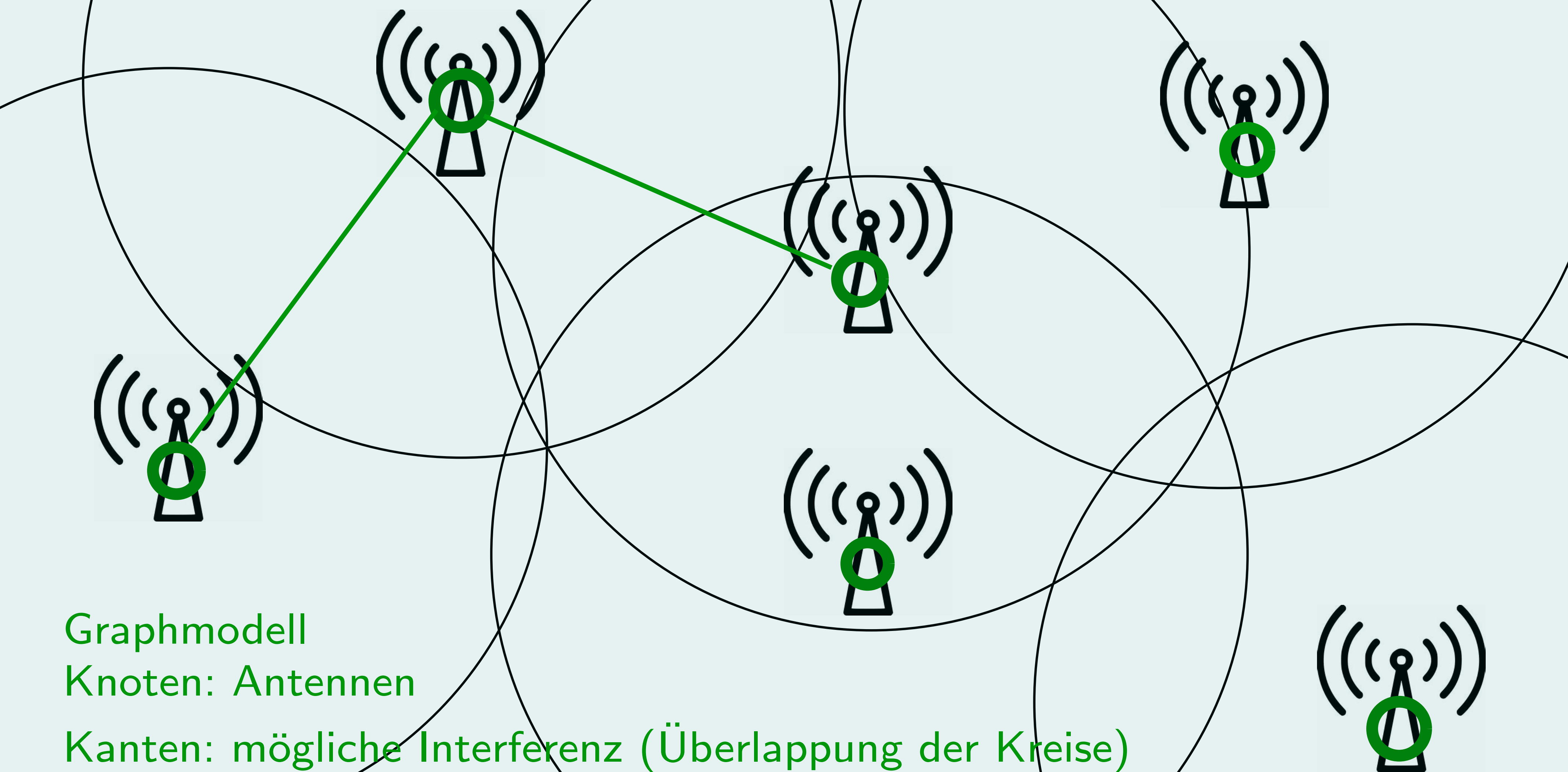
Graphmodell

Knoten: Antennen

Kanten: mögliche Interferenz (Überlappung der Kreise)

Frequenzzuweisungen

Wieviele unterschiedliche Frequenzen brauche ich mindestens, um Interferenz zu vermeiden?



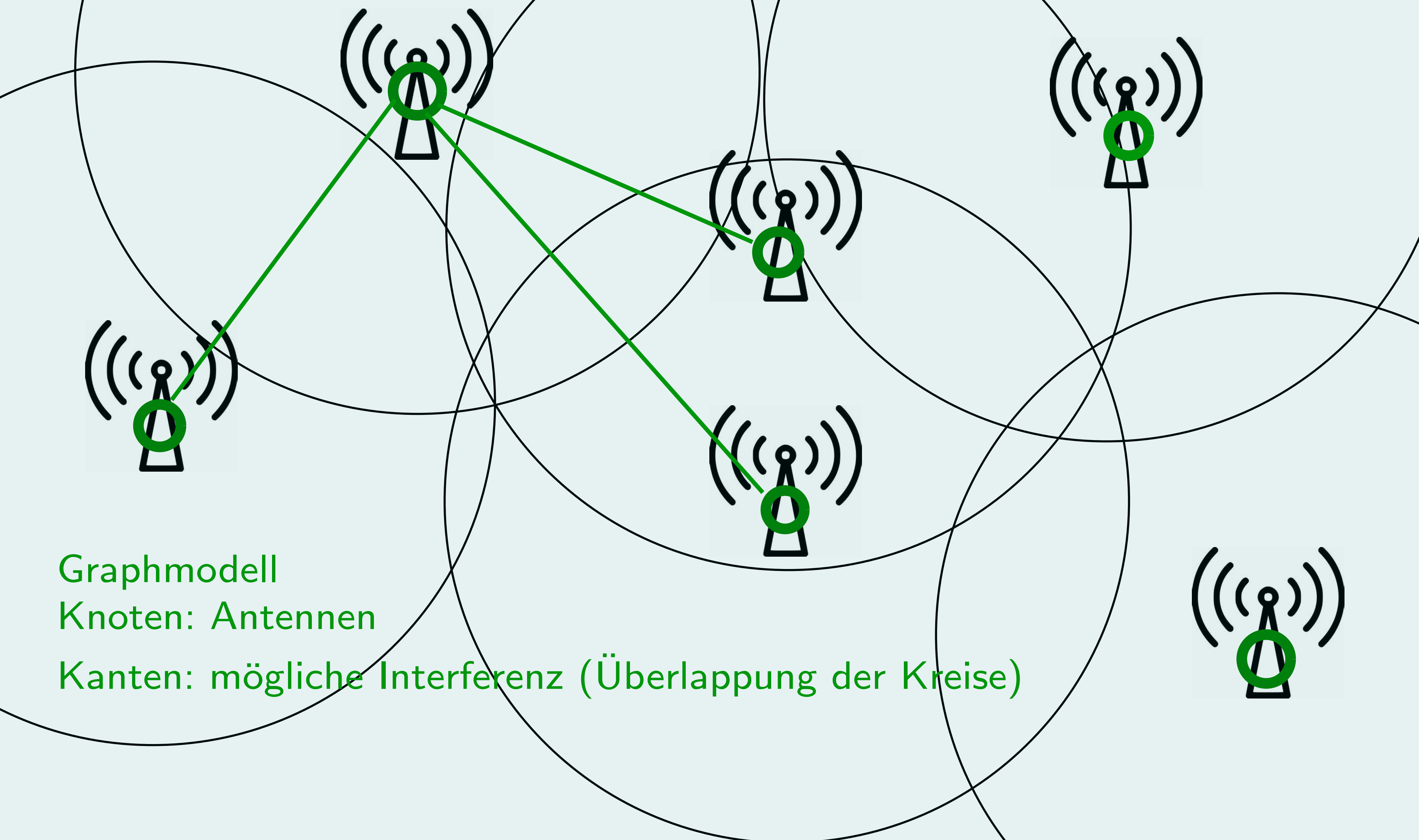
Graphmodell

Knoten: Antennen

Kanten: mögliche Interferenz (Überlappung der Kreise)

Frequenzzuweisungen

Wieviele unterschiedliche Frequenzen brauche ich mindestens, um Interferenz zu vermeiden?



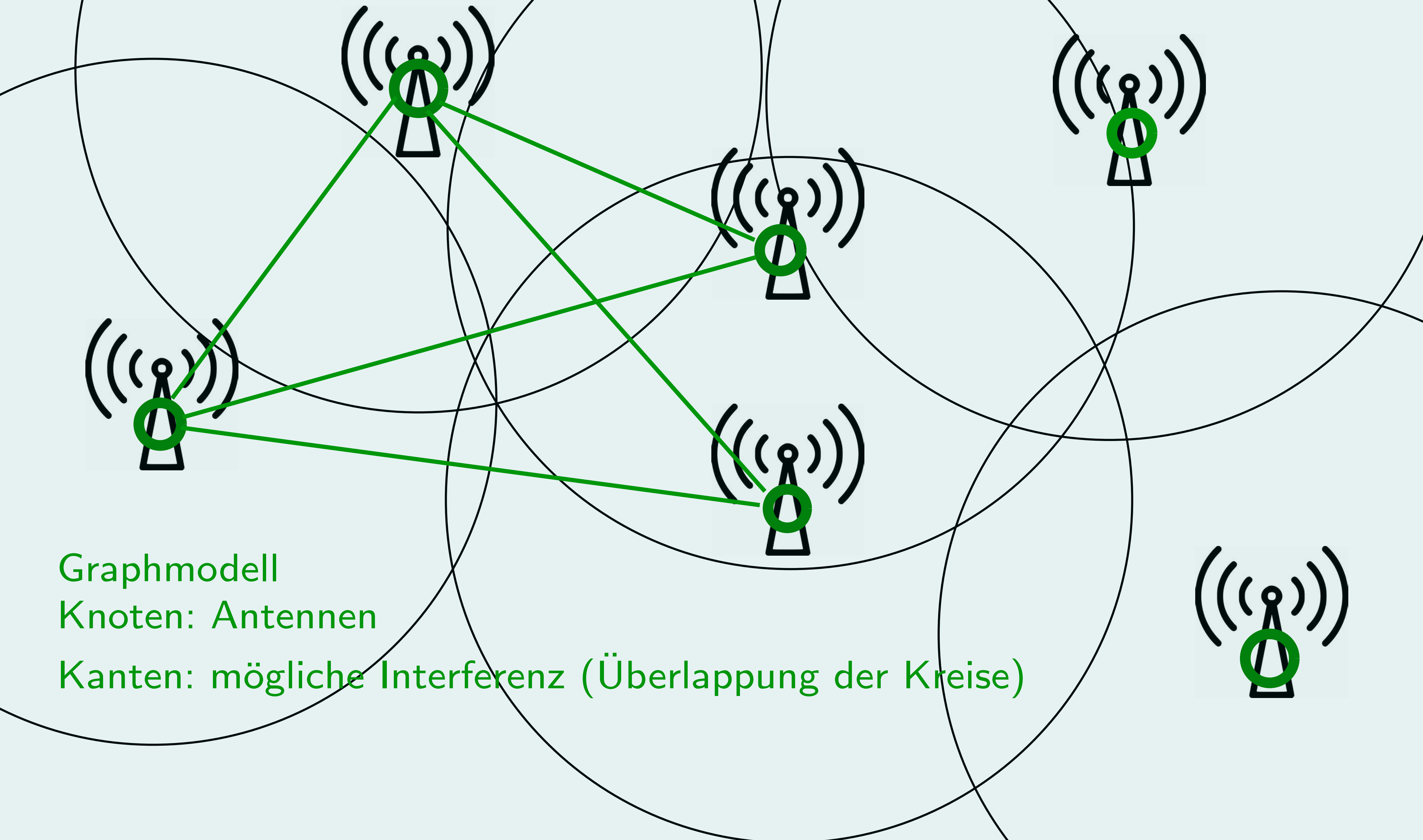
Graphmodell

Knoten: Antennen

Kanten: mögliche Interferenz (Überlappung der Kreise)

Frequenzzuweisungen

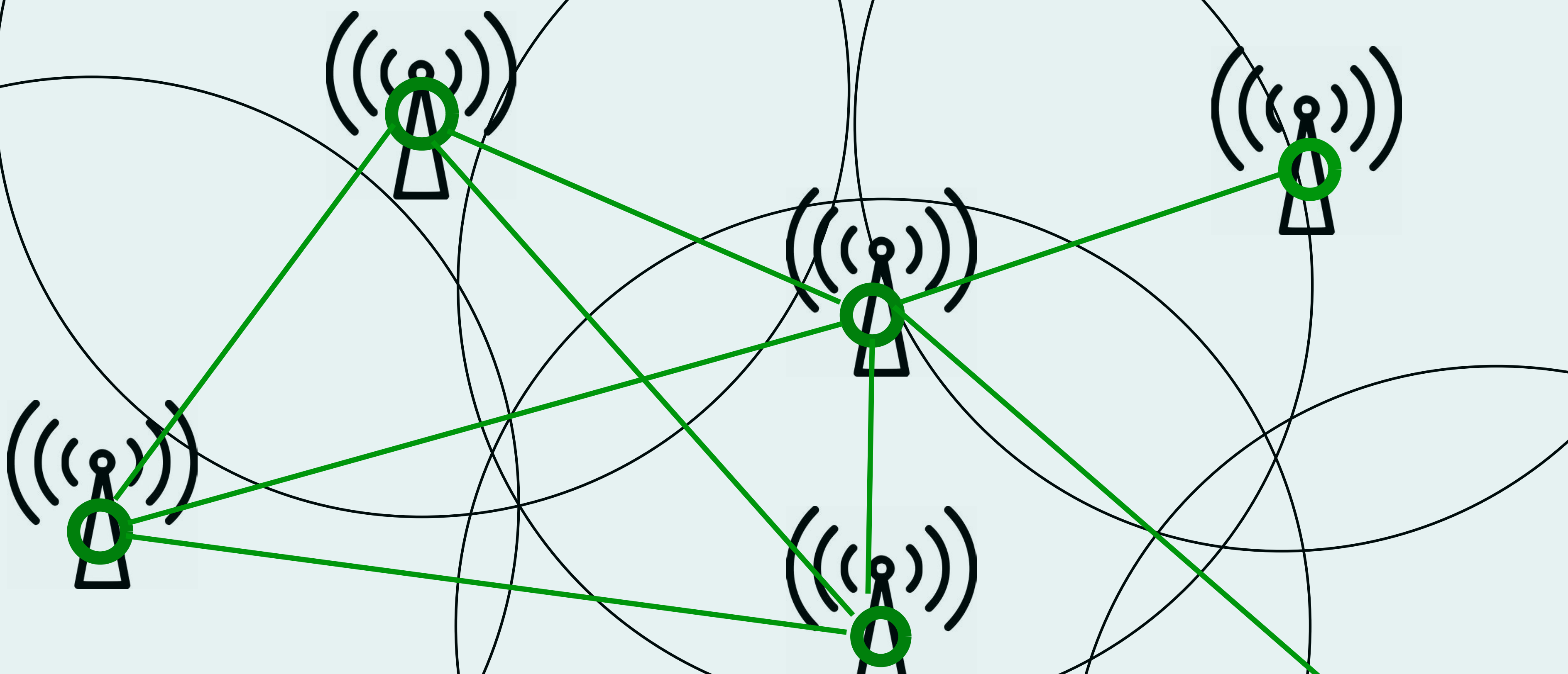
Wieviele unterschiedliche Frequenzen brauche ich mindestens, um Interferenz zu vermeiden?



Graphmodell
Knoten: Antennen
Kanten: mögliche Interferenz (Überlappung der Kreise)

Frequenzzuweisungen

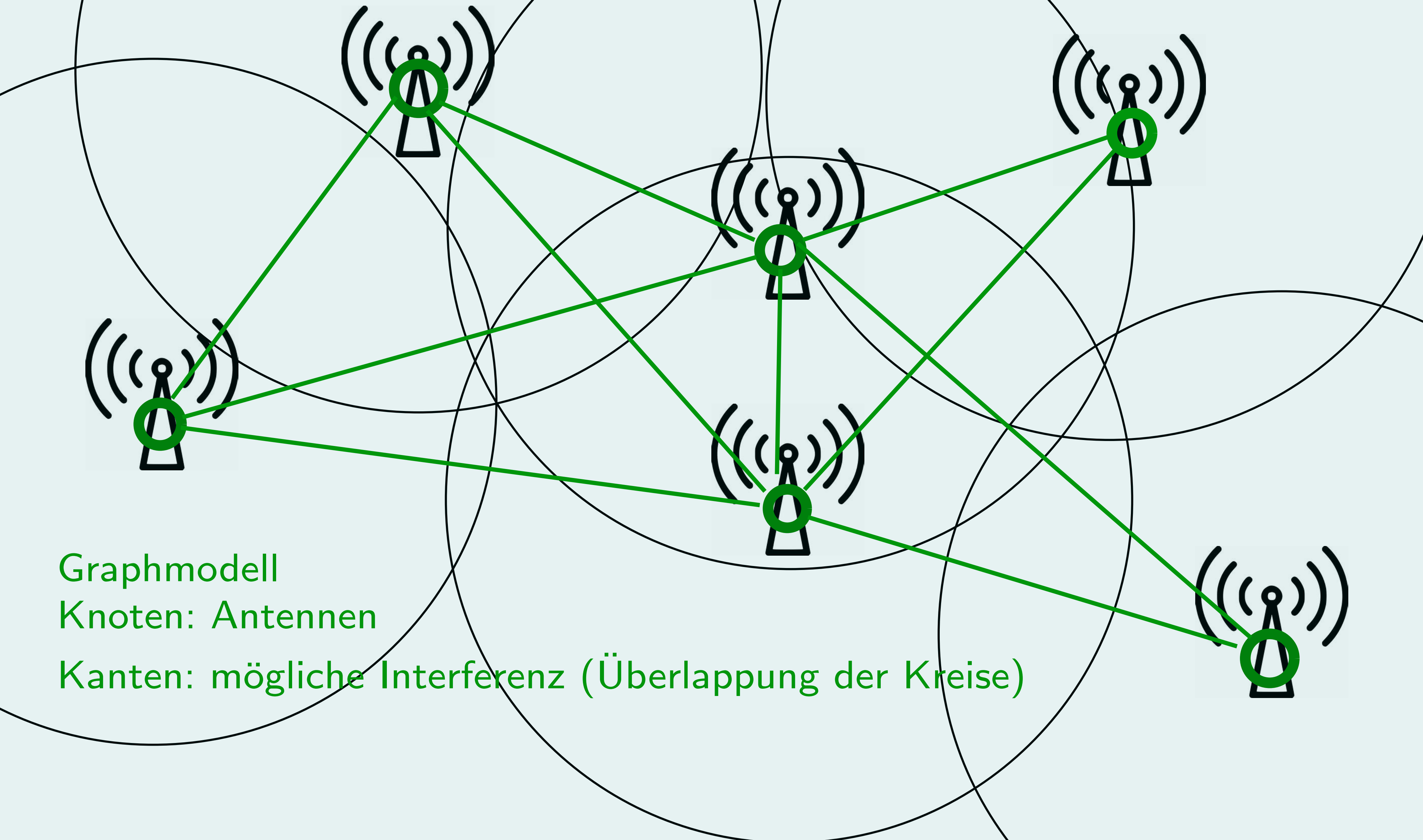
Wieviele unterschiedliche Frequenzen brauche ich mindestens, um Interferenz zu vermeiden?



Graphmodell
Knoten: Antennen
Kanten: mögliche Interferenz (Überlappung der Kreise)

Frequenzzuweisungen

Wieviele unterschiedliche Frequenzen brauche ich mindestens, um Interferenz zu vermeiden?



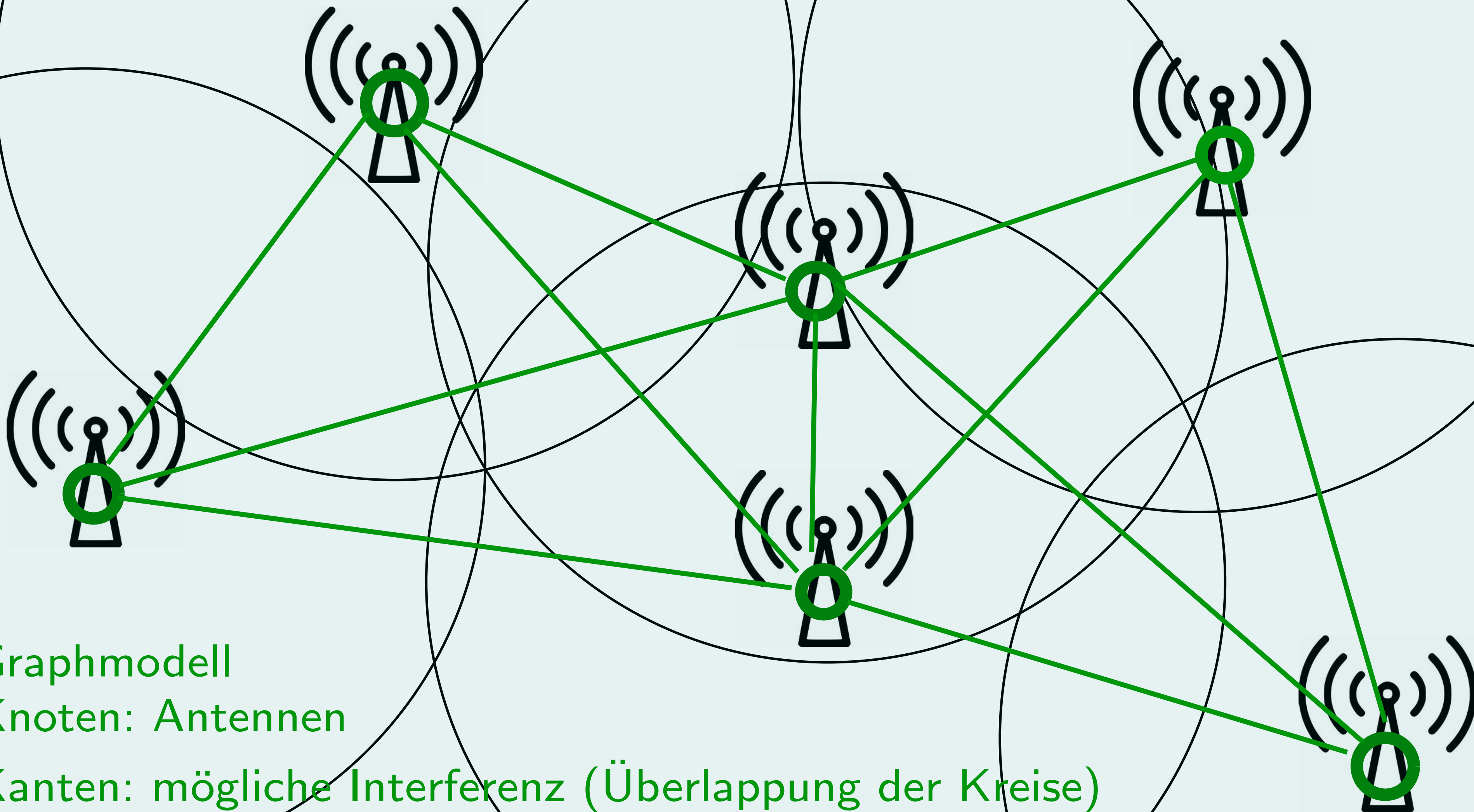
Graphmodell

Knoten: Antennen

Kanten: mögliche Interferenz (Überlappung der Kreise)

Frequenzzuweisungen

Wieviele unterschiedliche Frequenzen brauche ich mindestens, um Interferenz zu vermeiden?



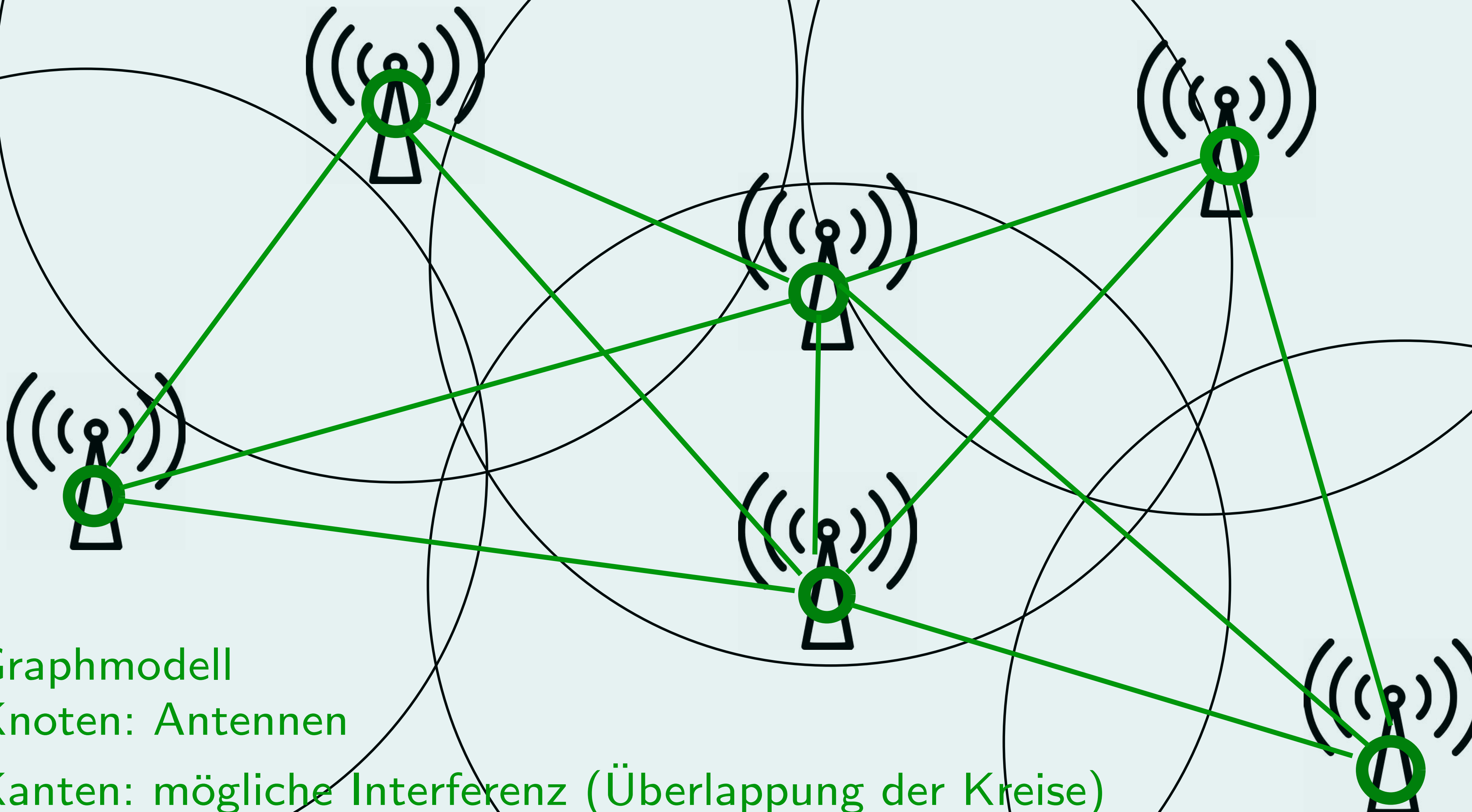
Graphmodell

Knoten: Antennen

Kanten: mögliche Interferenz (Überlappung der Kreise)

Frequenzzuweisungen

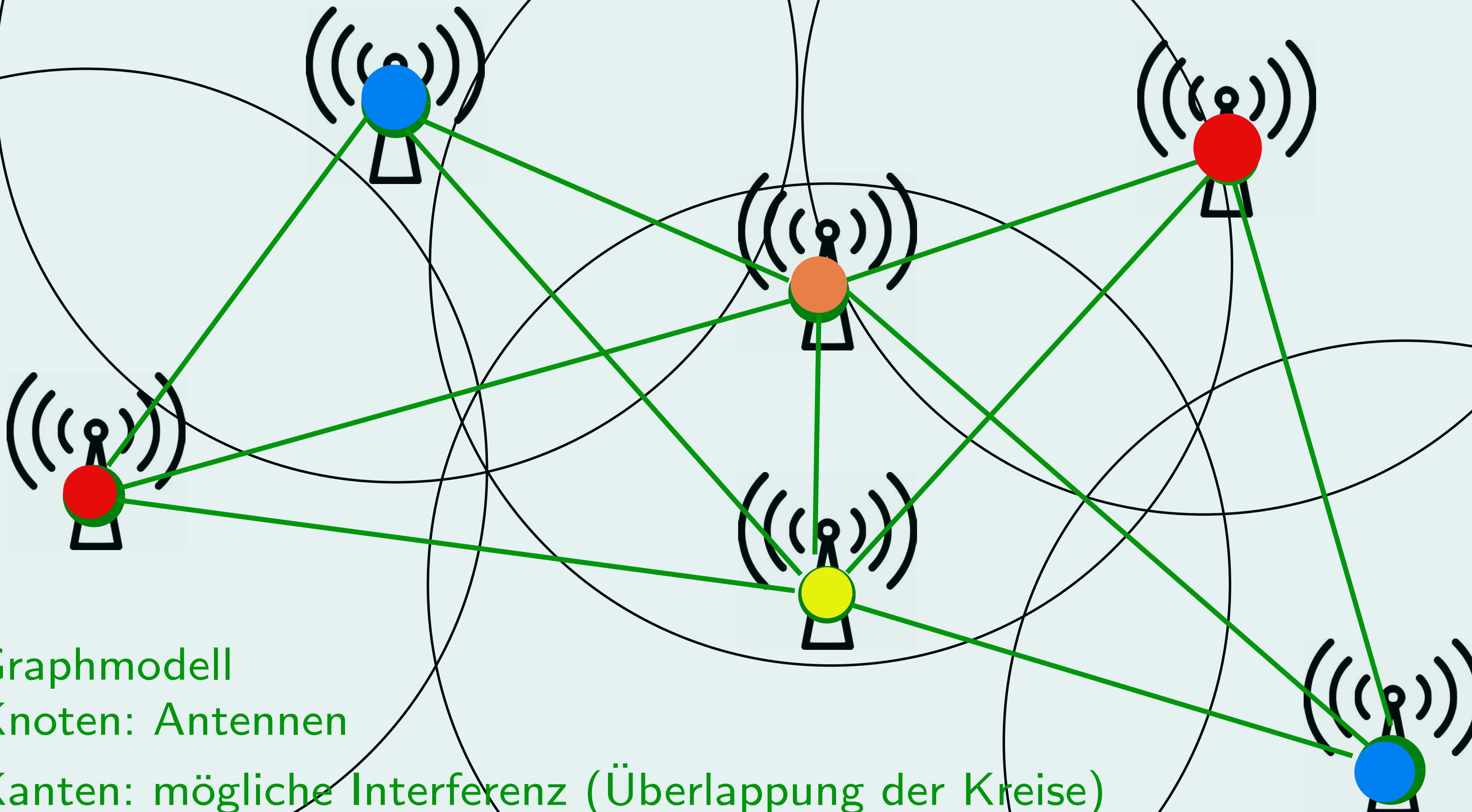
Wieviele unterschiedliche Frequenzen brauche ich mindestens, um Interferenz zu vermeiden?



Graphmodell
Knoten: Antennen
Kanten: mögliche Interferenz (Überlappung der Kreise)
Farben: Frequenzen

Frequenzzuweisungen

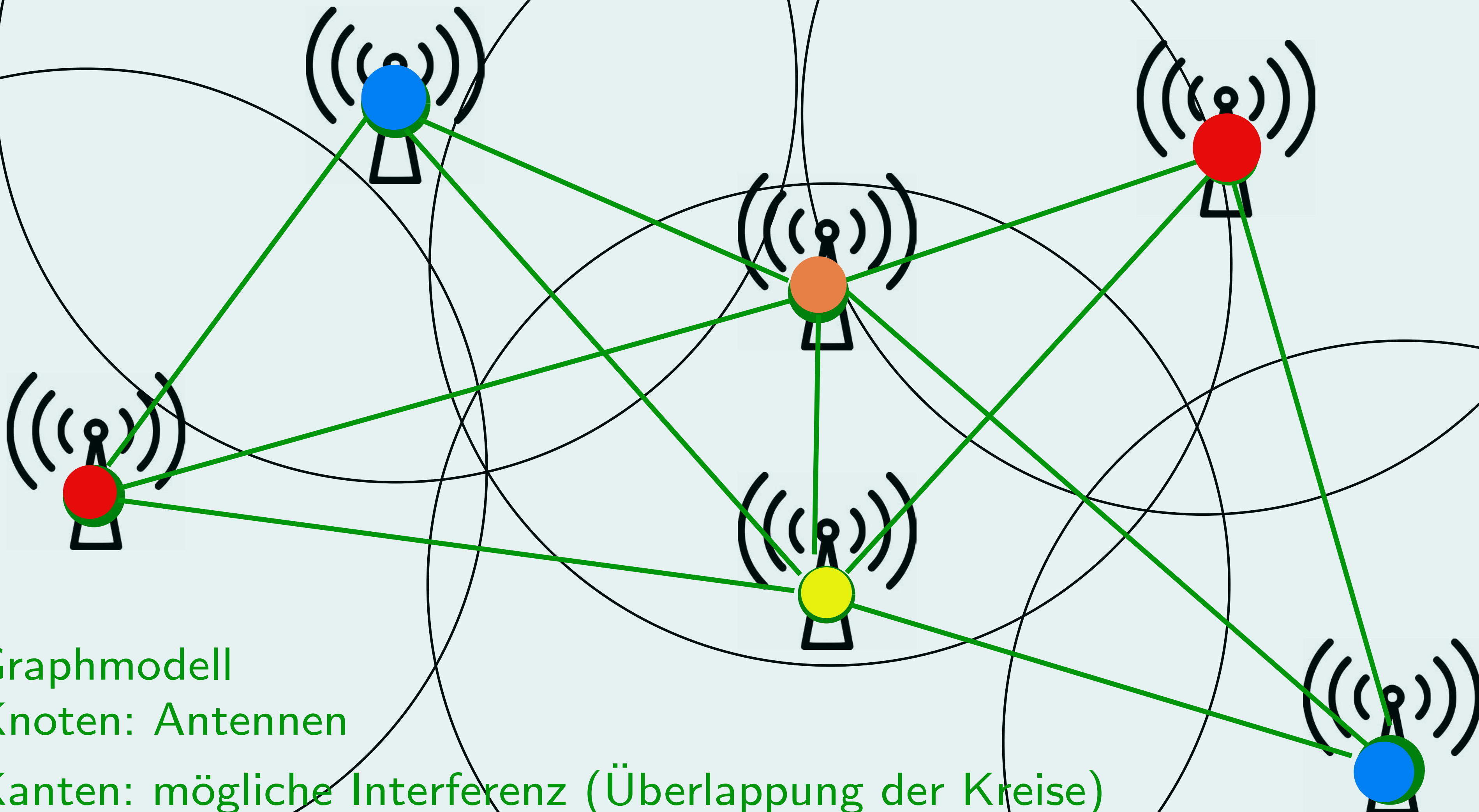
Wieviele unterschiedliche Frequenzen brauche ich mindestens, um Interferenz zu vermeiden?



Graphmodell
Knoten: Antennen
Kanten: mögliche Interferenz (Überlappung der Kreise)
Farben: Frequenzen

Frequenzzuweisungen

Wieviele unterschiedliche Frequenzen brauche ich mindestens, um Interferenz zu vermeiden?



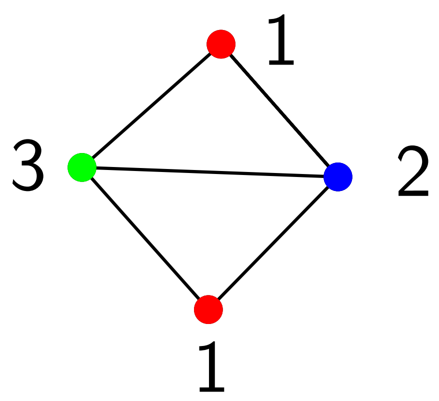
Graphmodell
Knoten: Antennen

Kanten: mögliche Interferenz (Überlappung der Kreise)
Farben: Frequenzen

Wie viele Farben brauchen wir, um die Knoten des Graphen so zu färben, dass keine zwei benachbarten Knoten die gleiche Farbe haben?

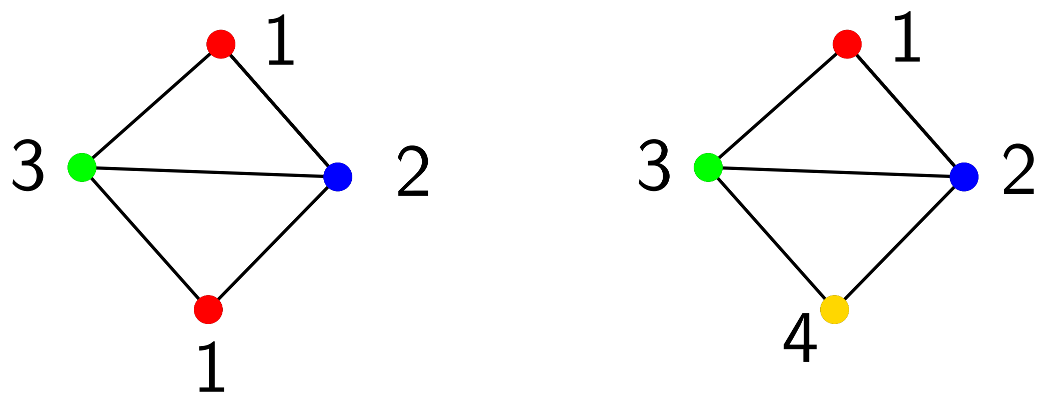
Färbungen und chromatische Zahl

Sei $G = (V, E)$ ein Graph.
Eine *k-Färbung* ist eine Abbildung $f\colon V \to \{1, \ldots, k\}$,
so dass für alle $\{u, v\} \in E$ gilt $f(u) \neq f(v)$.



Färbungen und chromatische Zahl

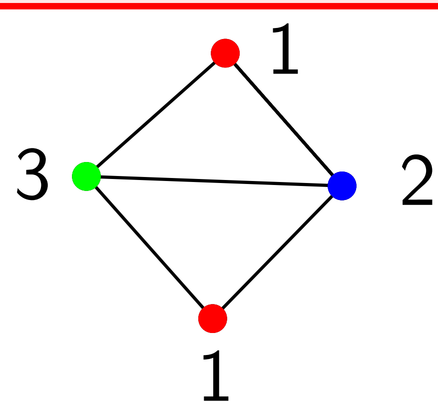
Sei $G = (V, E)$ ein Graph.
Eine *k-Färbung* ist eine Abbildung $f\colon V \to \{1, \dots, k\}$,
so dass für alle $\{u, v\} \in E$ gilt $f(u) \neq f(v)$.

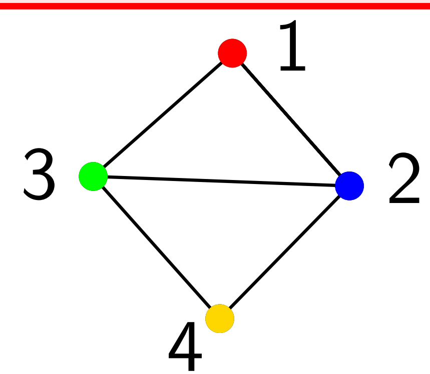


Färbungen und chromatische Zahl

Sei $G = (V, E)$ ein Graph.
Eine *k-Färbung* ist eine Abbildung $f\colon V \to \{1, \ldots, k\}$, so dass für alle $\{u, v\} \in E$ gilt $f(u) \neq f(v)$.

$\chi(G) = \min\{k \mid G \text{ hat eine } k\text{-Färbung}\}$ heißt *chromatische Zahl* von G.

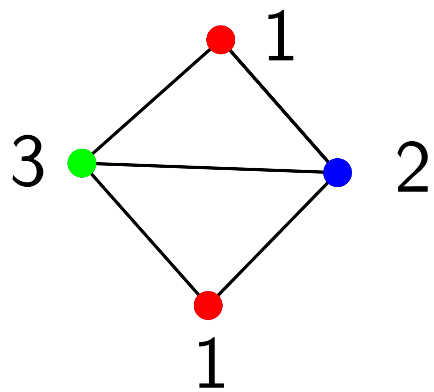


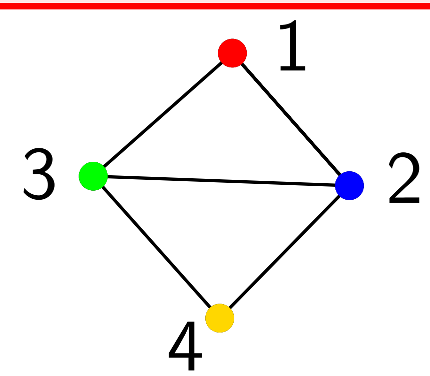


Färbungen und chromatische Zahl

Sei $G = (V, E)$ ein Graph.
Eine *k-Färbung* ist eine Abbildung $f\colon V \to \{1, \dots, k\}$, so dass für alle $\{u, v\} \in E$ gilt $f(u) \neq f(v)$.

$\chi(G) = \min\{k \mid G \text{ hat eine } k\text{-Färbung}\}$ heißt *chromatische Zahl* von G.



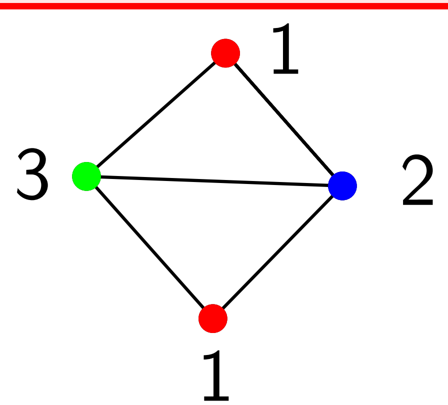


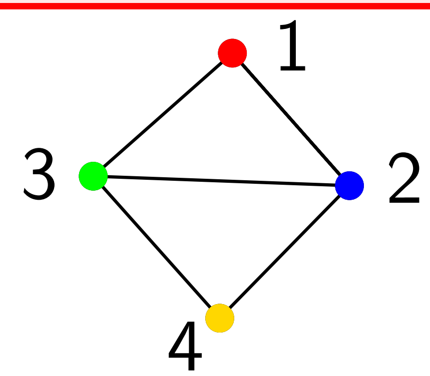
$\chi(G) = 3$

Färbungen und chromatische Zahl

Sei $G = (V, E)$ ein Graph.
Eine *k-Färbung* ist eine Abbildung $f\colon V \to \{1, \ldots, k\}$, so dass für alle $\{u, v\} \in E$ gilt $f(u) \neq f(v)$.

$\chi(G) = \min\{k \mid G \text{ hat eine } k\text{-Färbung}\}$ heißt *chromatische Zahl* von G.





Cliquenzahl

Beobachtung: Für jeden Graphen G gilt: $\chi(G) \quad \omega(G)$.

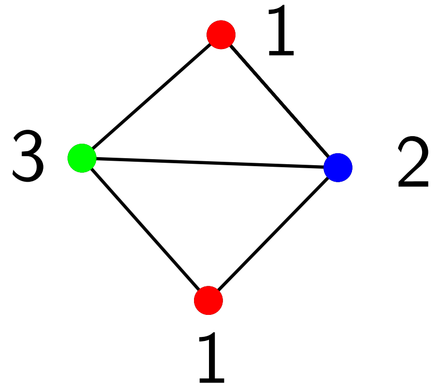
$\leq, \geq, <, >, =$

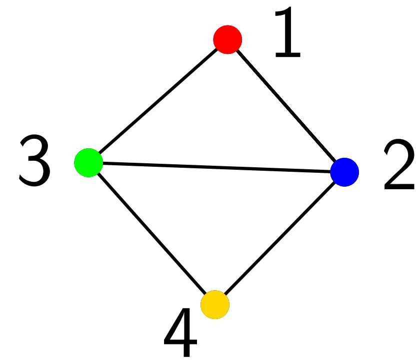
?

Färbungen und chromatische Zahl

Sei $G = (V, E)$ ein Graph.
Eine *k-Färbung* ist eine Abbildung $f\colon V \to \{1, \ldots, k\}$,
so dass für alle $\{u, v\} \in E$ gilt $f(u) \neq f(v)$.

$\chi(G) = \min\{k \mid G \text{ hat eine } k\text{-Färbung}\}$ heißt *chromatische Zahl* von G.





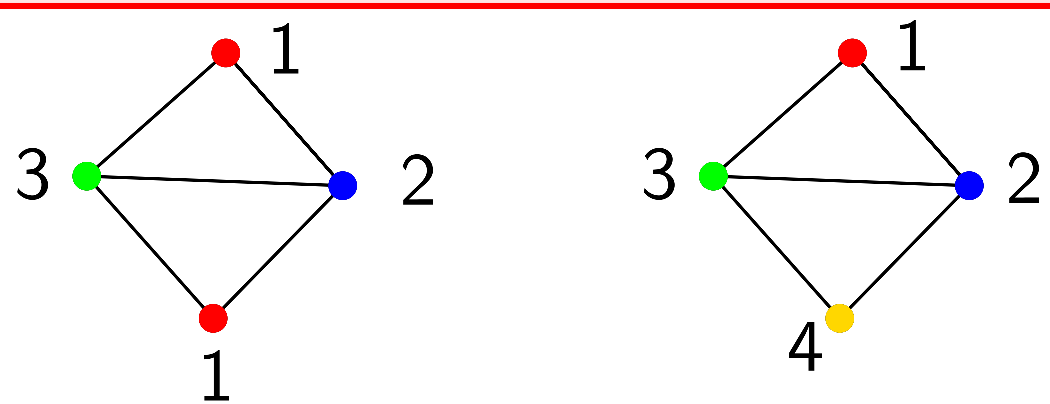
Cliquenzahl

Beobachtung: Für jeden Graphen G gilt: $\chi(G) \geq \omega(G)$.

Färbungen und chromatische Zahl

Sei $G = (V, E)$ ein Graph.
Eine *k-Färbung* ist eine Abbildung $f\colon V \to \{1, \ldots, k\}$, so dass für alle $\{u, v\} \in E$ gilt $f(u) \neq f(v)$.

$\chi(G) = \min\{k \mid G \text{ hat eine } k\text{-Färbung}\}$ heißt *chromatische Zahl* von G.



Cliquenzahl

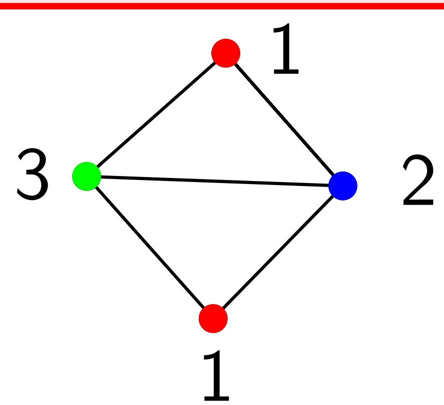
Beobachtung: Für jeden Graphen G gilt: $\chi(G) \geq \omega(G)$.

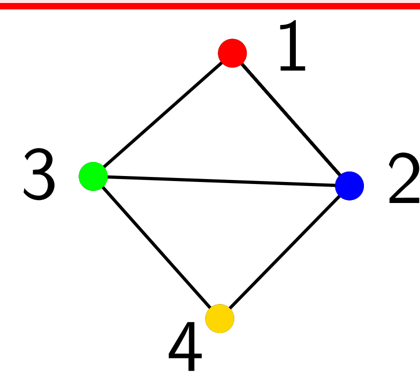
Graph mit $\chi(G) = \omega(G)$?

Färbungen und chromatische Zahl

Sei $G = (V, E)$ ein Graph.
Eine *k-Färbung* ist eine Abbildung $f\colon V \to \{1, \ldots, k\}$, so dass für alle $\{u, v\} \in E$ gilt $f(u) \neq f(v)$.

$\chi(G) = \min\{k \mid G$ hat eine k-Färbung$\}$ heißt *chromatische Zahl* von G.





Cliquenzahl

Beobachtung: Für jeden Graphen G gilt: $\chi(G) \geq \omega(G)$.

Graph mit $\chi(G) > \omega(G)$?

Zusammenhang: Clique - unabhängige Menge

Zusammenhang: Clique - unabhängige Menge

Farbe von i

Beobachtung:
Sei f eine k-Färbung von G. Dann ist $f^{-1}(i) \subseteq V$ für $i = 1, \ldots, k$ eine unabhängige Menge.

Zusammenhang: Clique - unabhängige Menge

Farbe von i

Beobachtung:
Sei f eine k-Färbung von G. Dann ist $f^{-1}(i) \subseteq V$ für $i = 1, \ldots, k$ eine unabhängige Menge.

Unabhängigkeitszahl

$$\Rightarrow |f^{-1}(i)| \leq \alpha(G)$$

Zusammenhang: Clique - unabhängige Menge

Farbe von i

Beobachtung:
Sei f eine k-Färbung von G. Dann ist $f^{-1}(i) \subseteq V$ für $i = 1, \ldots, k$ eine unabhängige Menge.

Unabhängigkeitszahl

$$\Rightarrow |f^{-1}(i)| \leq \alpha(G) \Rightarrow n \leq \chi(G) \cdot \alpha(G)$$

chromatische Zahl

Zusammenhang: Clique - unabhängige Menge

Farbe von i

Beobachtung:

Sei f eine k-Färbung von G. Dann ist $f^{-1}(i) \subseteq V$ für $i = 1, \ldots, k$ eine unabhängige Menge.

Unabhängigkeitszahl

$$\Rightarrow |f^{-1}(i)| \leq \alpha(G) \Rightarrow n \leq \chi(G) \cdot \alpha(G)$$

chromatische Zahl

Korollar:

$\chi(G) \geq \max\{\omega(G), n/\alpha(G)\}$.

Obere Schranke an die chromatische Zahl?

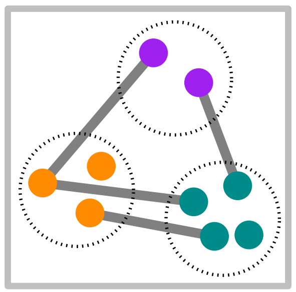
Obere Schranke an die chromatische Zahl?

$$\boxed{\chi(G) \leq \left\lfloor \frac{1}{2} + \sqrt{2|E| + \frac{1}{4}} \right\rfloor .}$$

Obere Schranke an die chromatische Zahl?

$$\boxed{\chi(G) \leq \left\lfloor \tfrac{1}{2} + \sqrt{2|E| + \tfrac{1}{4}} \right\rfloor .}$$

Beweis. Sei f eine k-Färbung von G mit $k = \chi(G)$



Zwischen je 2 Farbklassen bzgl. f gibt's ≥ 1 Kante.

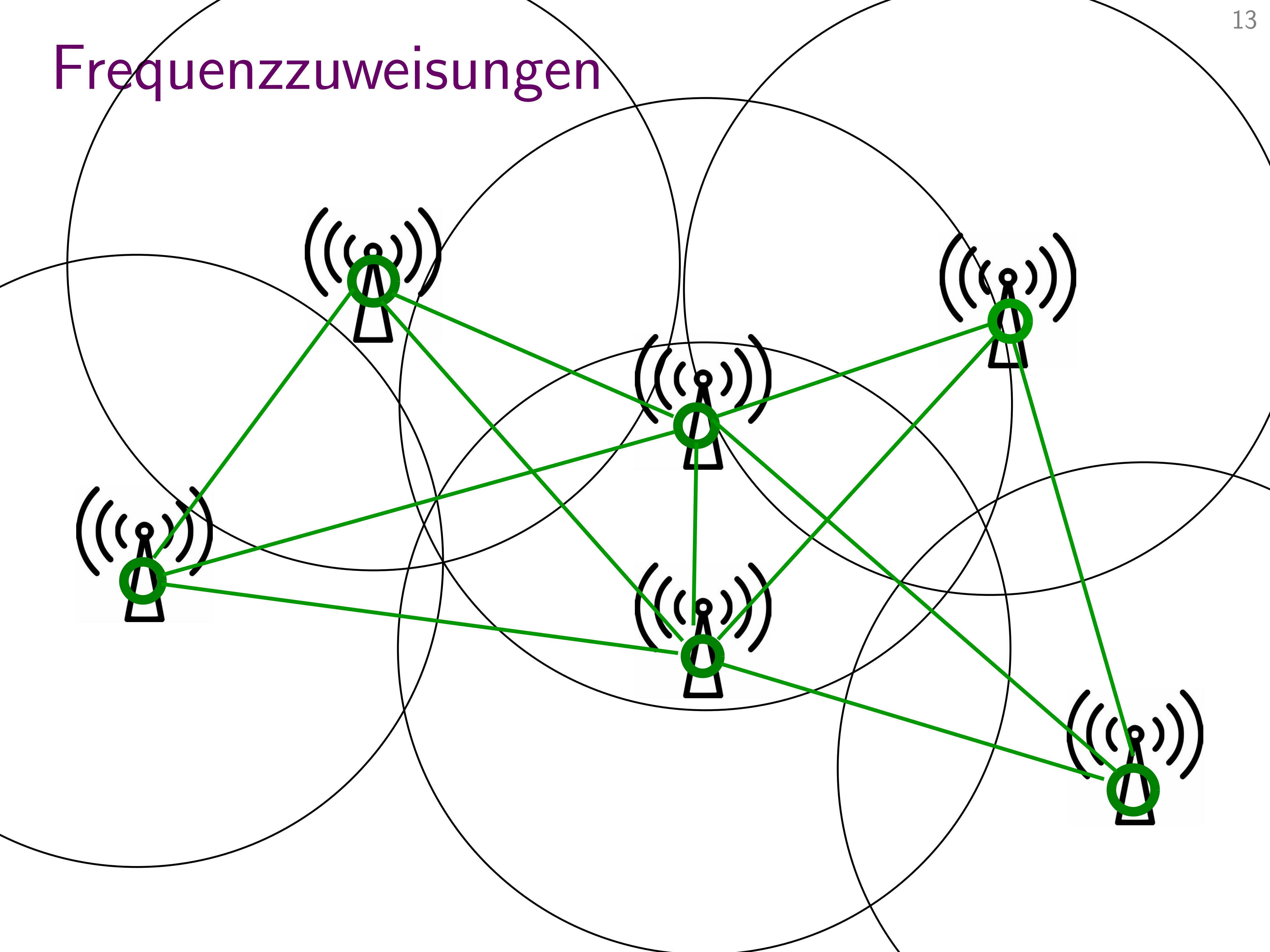
(Sonst gäb's eine $(k-1)$-Färbung!)

$\Rightarrow |E| \geq k(k-1)/2 \;\Rightarrow \chi(G) = k \leq \frac{1}{2} + \sqrt{2|E| + \frac{1}{4}}$ $\square$

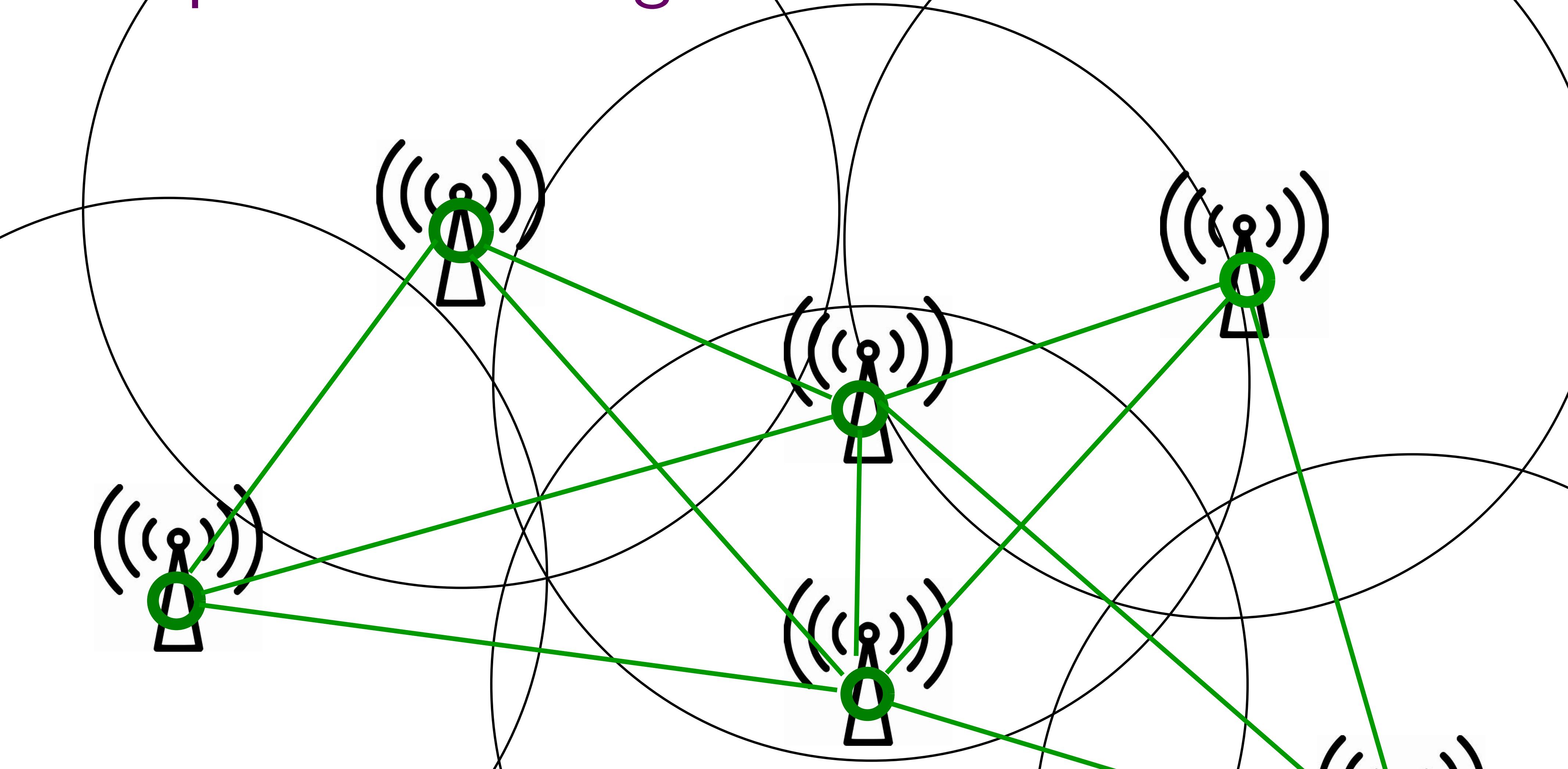
Greedy-Algorithmus zum Graphenfärben

Graphenfärbungsproblem
Gegeben: Graph $G = (V, E)$
Gesucht: Färbung mit möglichst wenig Farben (so dass keine zwei adjazenten Knoten dieselbe Farbe haben)

Frequenzzuweisungen



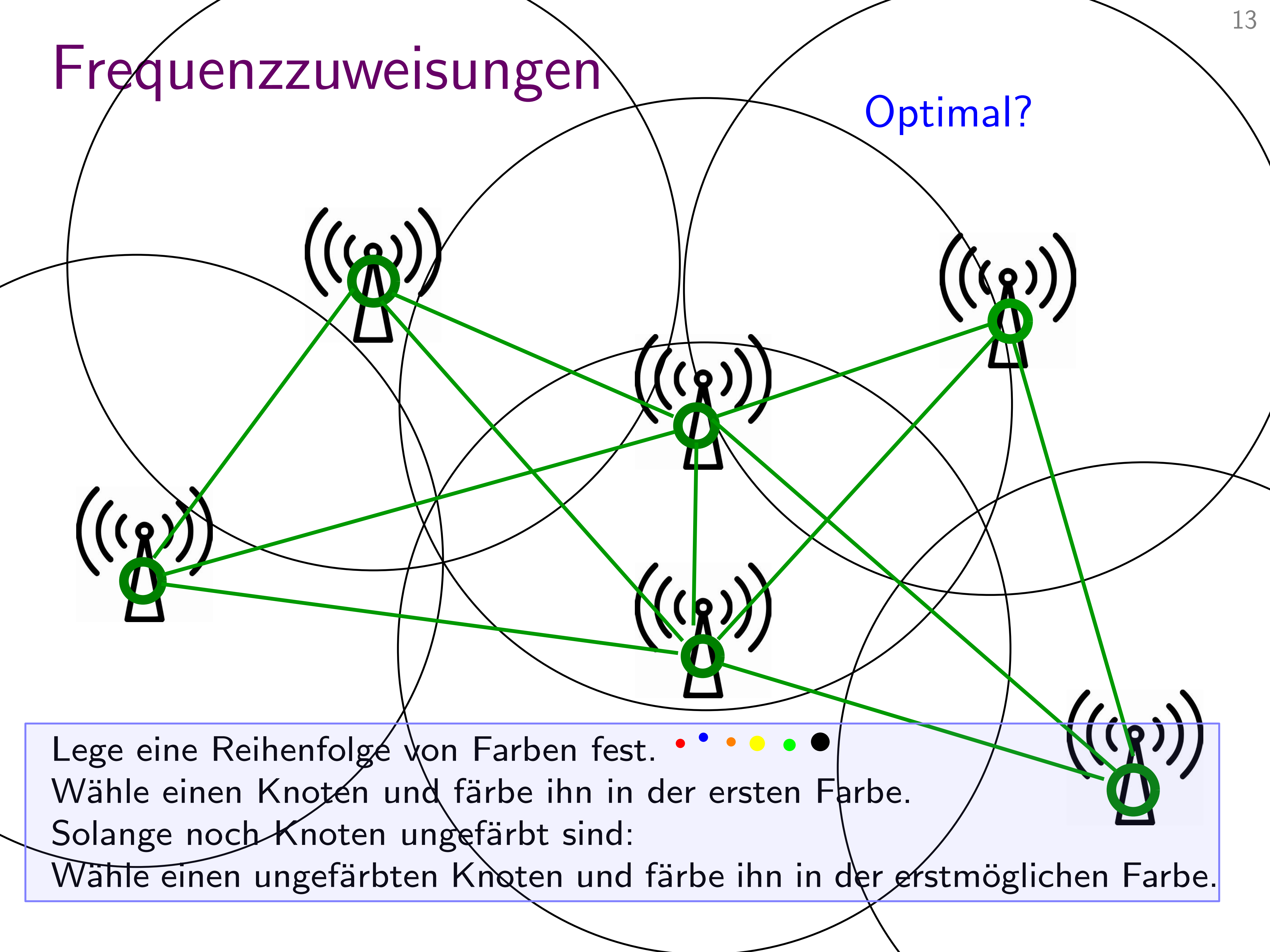
Frequenzzuweisungen



Lege eine Reihenfolge von Farben fest.
Wähle einen Knoten und färbe ihn in der ersten Farbe.
Solange noch Knoten ungefärbt sind:
Wähle einen ungefärbten Knoten und färbe ihn in der erstmöglichen Farbe.

Frequenzzuweisungen

Optimal?



Lege eine Reihenfolge von Farben fest.
Wähle einen Knoten und färbe ihn in der ersten Farbe.
Solange noch Knoten ungefärbt sind:
Wähle einen ungefärbten Knoten und färbe ihn in der erstmöglichen Farbe.

Greedy-Algorithmus zum Graphenfärben

Graphenfärbungsproblem
Gegeben: Graph $G = (V, E)$
Gesucht: Färbung mit möglichst wenig Farben (so dass keine zwei adjazenten Knoten dieselbe Farbe haben)

Greedy-Algorithmus zum Graphenfärben
Lege eine Reihenfolge von Farben fest.
Wähle einen Knoten und färbe ihn in der ersten Farbe.
Solange noch Knoten ungefärbt sind:
Wähle einen ungefärbten Knoten und färbe ihn in der erstmöglichen Farbe.

Greedy-Algorithmus zum Graphenfärben

Graphenfärbungsproblem
Gegeben: Graph $G = (V, E)$
Gesucht: Färbung mit möglichst wenig Farben (so dass keine zwei adjazenten Knoten dieselbe Farbe haben)

Greedy-Algorithmus zum Graphenfärben
Lege eine Reihenfolge von Farben fest.
Wähle einen Knoten und färbe ihn in der ersten Farbe.
Solange noch Knoten ungefärbt sind:
Wähle einen ungefärbten Knoten und färbe ihn in der erstmöglichen Farbe.

Exakter Algorithmus für Problem P: Algorithmus, der für jede Instanz des Problems P eine Optimallösung findet

Heuristischer Algorithmus (kurz: **Heuristik**) für Problem P: Algorithmus, der eine *möglichst gute zulässige* Lösung für P findet.

Greedy-Algorithmus zum Graphenfärben

Graphenfärbungsproblem
Gegeben: Graph $G = (V, E)$
Gesucht: Färbung mit möglichst wenig Farben (so dass keine zwei adjazenten Knoten dieselbe Farbe haben)

Greedy-Algorithmus zum Graphenfärben
Lege eine Reihenfolge von Farben fest.
Wähle einen Knoten und färbe ihn in der ersten Farbe.
Solange noch Knoten ungefärbt sind:
Wähle einen ungefärbten Knoten und färbe ihn in der erstmöglichen Farbe.

Exakter Algorithmus für Problem P: Algorithmus, der für jede Instanz des Problems P eine Optimallösung findet

Heuristischer Algorithmus (kurz: **Heuristik**) für Problem P: Algorithmus, der eine *möglichst gute zulässige* Lösung für P findet.

Für welche Graphen können wir leicht eine gute Reihenfolge festlegen?

Greedy-Algorithmus zum Graphenfärben

Für welche Graphen können wir leicht eine gute Färbungsreihenfolge für den Greedy-Algorithmus angeben?

Greedy-Algorithmus zum Graphenfärben

Für welche Graphen können wir leicht eine gute Färbungsreihenfolge für den Greedy-Algorithmus angeben?

- Pfade
- Bäume
- Kreise
- bipartite Graphen
- ...

Ein Scheduling-Problem

Eine Firma besitzt fünf Lötmaschinen des gleichen Typs. Die Lötbäder der Maschinen müssen, wenn diese benutzt werden sollen, aufwändig auf Betriebstemperatur gebracht werden. Für folgende Zeitfenster sind die Maschinen heute gebucht: 10:15-12:00, 10:45-13:45, 13:15-15:30, 12:30-14:30, 10:30-11:15, 12:15-12:45, 11:45-15:30.

Ein Scheduling-Problem

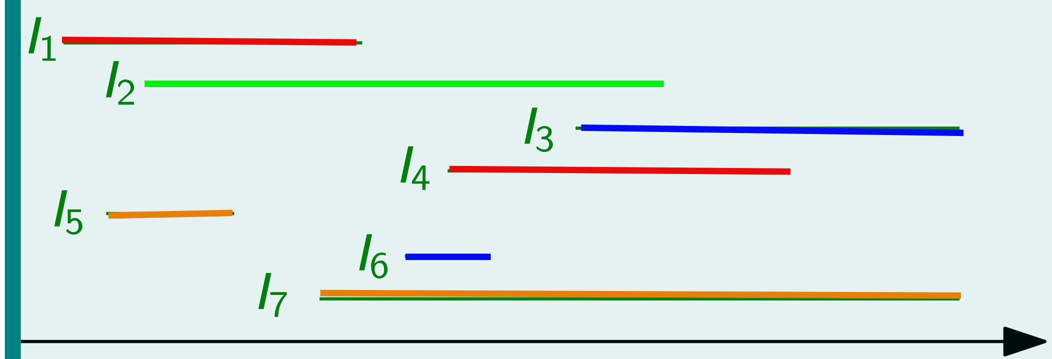
Eine Firma besitzt fünf Lötmaschinen des gleichen Typs. Die Lötbäder der Maschinen müssen, wenn diese benutzt werden sollen, aufwändig auf Betriebstemperatur gebracht werden. Für folgende Zeitfenster sind die Maschinen heute gebucht: 10:15-12:00, 10:45-13:45, 13:15-15:30, 12:30-14:30, 10:30-11:15, 12:15-12:45, 11:45-15:30.

Wie viele Maschinen muss die Firma heute aufwärmen?

Ein Scheduling-Problem

Eine Firma besitzt fünf Lötmaschinen des gleichen Typs. Die Lötbäder der Maschinen müssen, wenn diese benutzt werden sollen, aufwändig auf Betriebstemperatur gebracht werden. Für folgende Zeitfenster sind die Maschinen heute gebucht: 10:15-12:00, 10:45-13:45, 13:15-15:30, 12:30-14:30, 10:30-11:15, 12:15-12:45, 11:45-15:30.

Wie viele Maschinen muss die Firma heute aufwärmen?

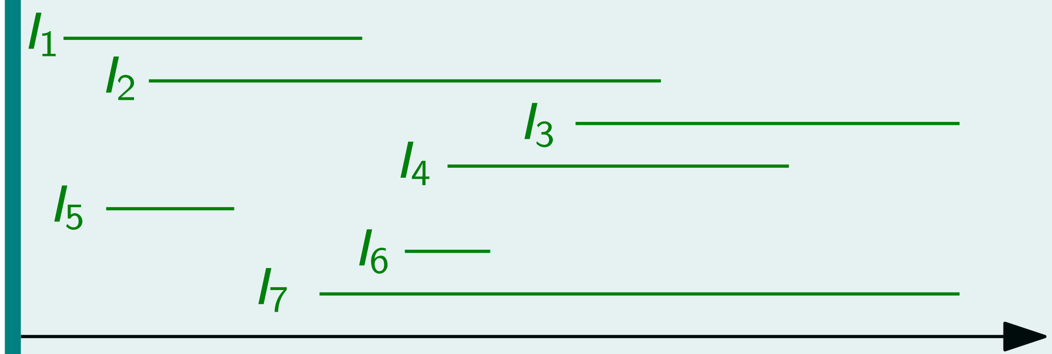


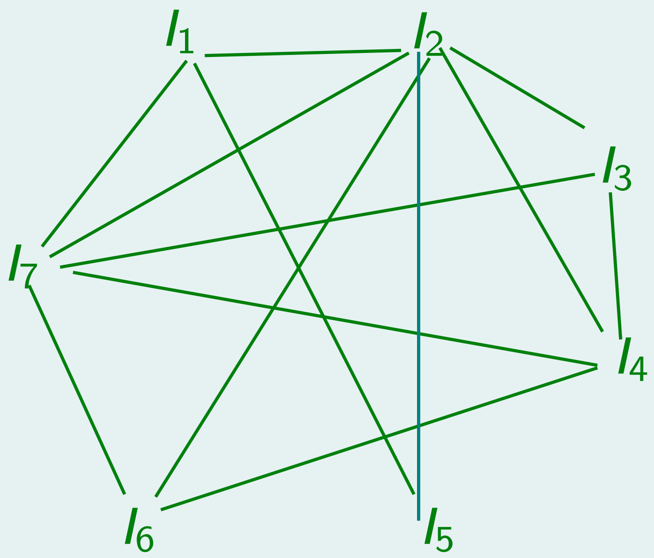
Ein Scheduling-Problem

Eine Firma besitzt fünf Lötmaschinen des gleichen Typs. Die Lötbäder der Maschinen müssen, wenn diese benutzt werden sollen, aufwändig auf Betriebstemperatur gebracht werden. Für folgende Zeitfenster sind die Maschinen heute gebucht: 10:15-12:00, 10:45-13:45, 13:15-15:30, 12:30-14:30, 10:30-11:15, 12:15-12:45, 11:45-15:30.

Ein Scheduling-Problem

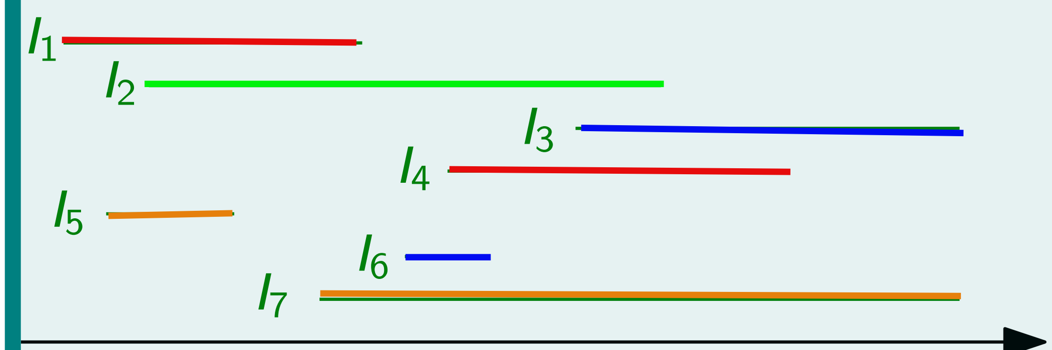
Eine Firma besitzt fünf Lötmaschinen des gleichen Typs. Die Lötbäder der Maschinen müssen, wenn diese benutzt werden sollen, aufwändig auf Betriebstemperatur gebracht werden. Für folgende Zeitfenster sind die Maschinen heute gebucht: 10:15-12:00, 10:45-13:45, 13:15-15:30, 12:30-14:30, 10:30-11:15, 12:15-12:45, 11:45-15:30.

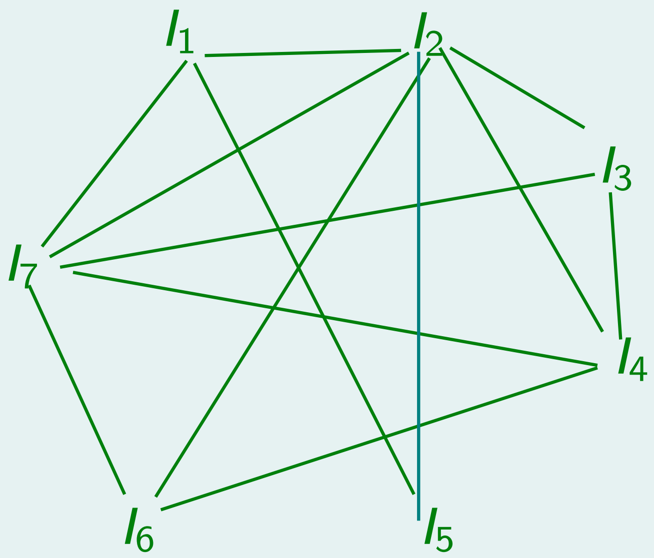




Ein Scheduling-Problem

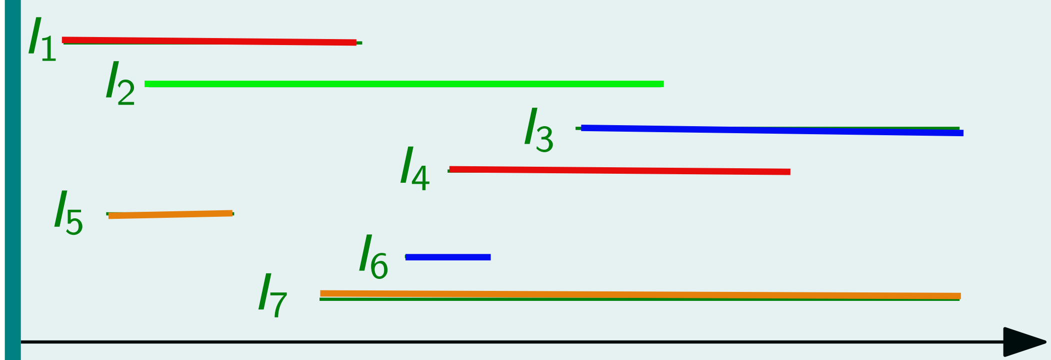
Eine Firma besitzt fünf Lötmaschinen des gleichen Typs. Die Lötbäder der Maschinen müssen, wenn diese benutzt werden sollen, aufwändig auf Betriebstemperatur gebracht werden. Für folgende Zeitfenster sind die Maschinen heute gebucht: 10:15-12:00, 10:45-13:45, 13:15-15:30, 12:30-14:30, 10:30-11:15, 12:15-12:45, 11:45-15:30.

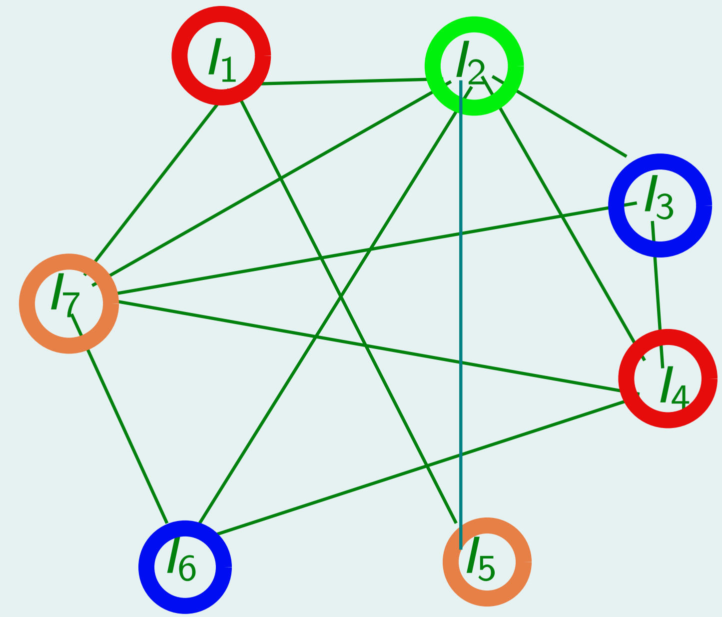




Ein Scheduling-Problem

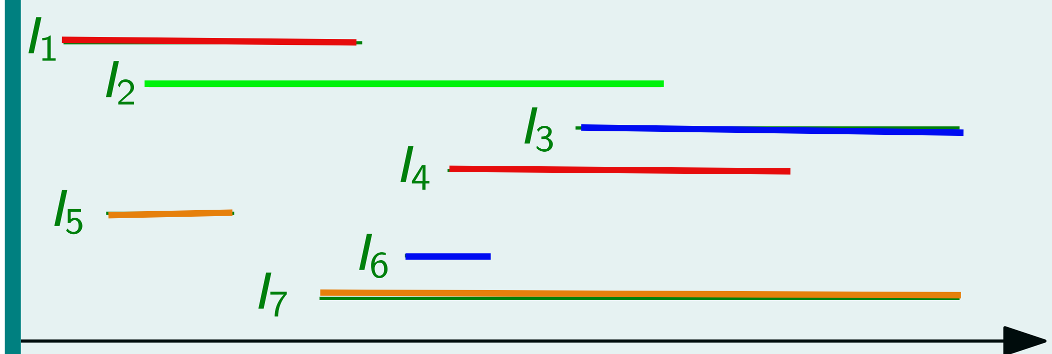
Eine Firma besitzt fünf Lötmaschinen des gleichen Typs. Die Lötbäder der Maschinen müssen, wenn diese benutzt werden sollen, aufwändig auf Betriebstemperatur gebracht werden. Für folgende Zeitfenster sind die Maschinen heute gebucht: 10:15-12:00, 10:45-13:45, 13:15-15:30, 12:30-14:30, 10:30-11:15, 12:15-12:45, 11:45-15:30.

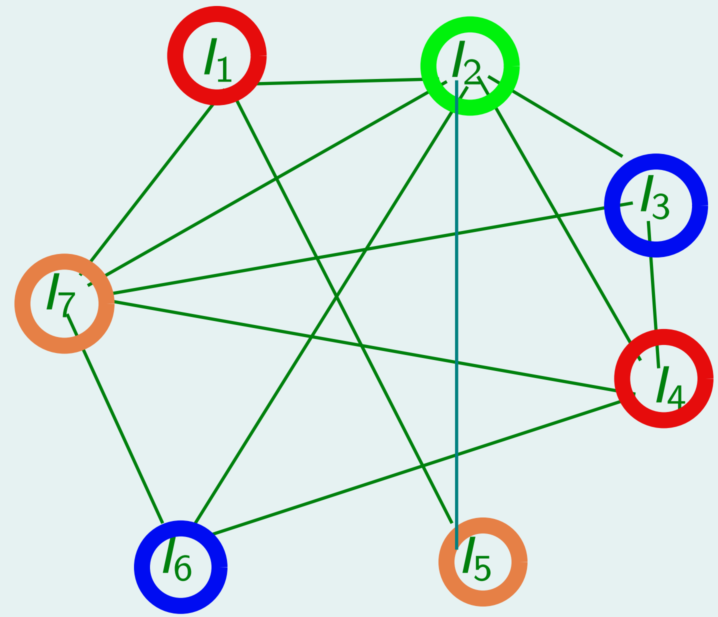




Ein Scheduling-Problem

Eine Firma besitzt fünf Lötmaschinen des gleichen Typs. Die Lötbäder der Maschinen müssen, wenn diese benutzt werden sollen, aufwändig auf Betriebstemperatur gebracht werden. Für folgende Zeitfenster sind die Maschinen heute gebucht: 10:15-12:00, 10:45-13:45, 13:15-15:30, 12:30-14:30, 10:30-11:15, 12:15-12:45, 11:45-15:30.

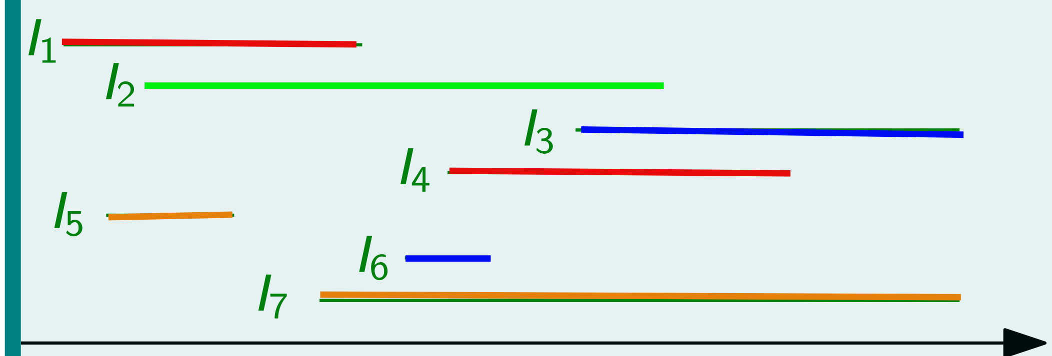


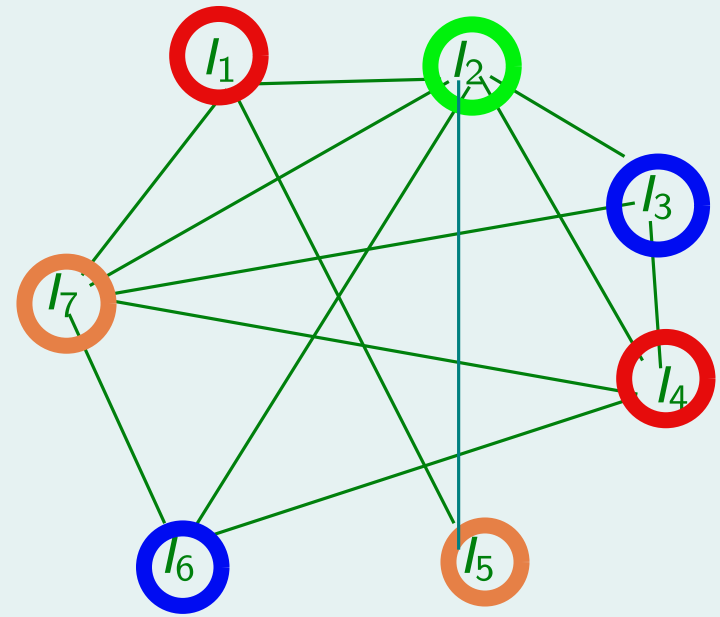


Ein Graph $G=(V,E)$ mit $|V|=n$ heißt **Intervallgraph**, wenn er ein Intervallmodell besitzt, d.h., wenn es eine bijektive Zuordnung von einer Menge $\Gamma=\{I_1,\ldots,I_n\}$ mit $I_j=[a_j,b_j]\subset\mathbb{R}$ für $j=1,\ldots,n$ auf die Knotenmenge V gibt, so dass $e=\{v_i,v_j\}\in E \Leftrightarrow I_i\cap I_j\neq\emptyset$.

Ein Scheduling-Problem

Eine Firma besitzt fünf Lötmaschinen des gleichen Typs. Die Lötbäder der Maschinen müssen, wenn diese benutzt werden sollen, aufwändig auf Betriebstemperatur gebracht werden. Für folgende Zeitfenster sind die Maschinen heute gebucht: 10:15-12:00, 10:45-13:45, 13:15-15:30, 12:30-14:30, 10:30-11:15, 12:15-12:45, 11:45-15:30.

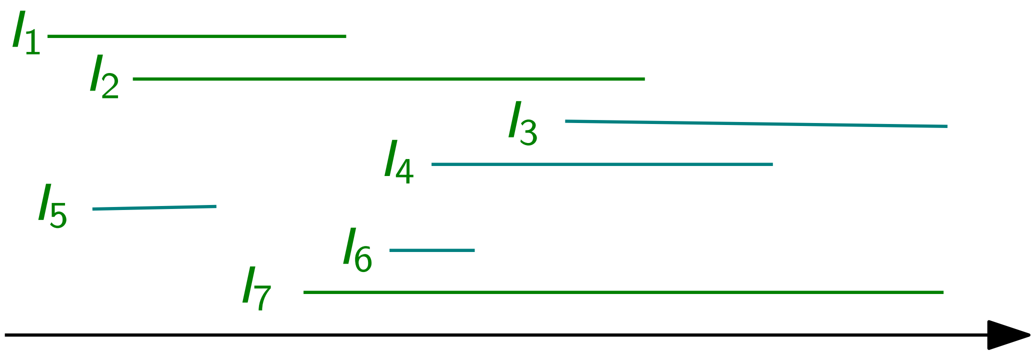


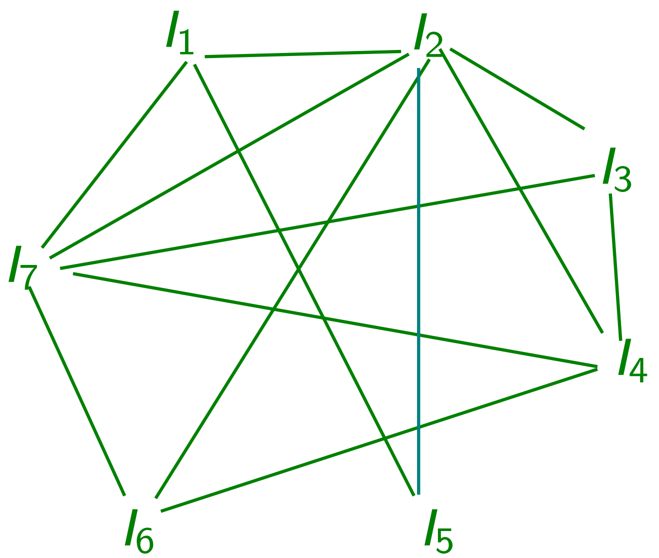


Ein Graph $G = (V, E)$ mit $|V| = n$ heißt **Intervallgraph**, wenn er ein Intervallmodell besitzt, d.h., wenn es eine bijektive Zuordnung von einer Menge $\Gamma = \{I_1, \ldots, I_n\}$ mit $I_j = [a_j, b_j] \subset \mathbb{R}$ für $j = 1, \ldots, n$ auf die Knotenmenge V gibt, so dass $e = \{v_i, v_j\} \in E \Leftrightarrow I_i \cap I_j \neq \emptyset$.

In welcher Reihenfolge färben wir die Knoten eines Intervallgraphen mit dem Greedy-Algorithmus?

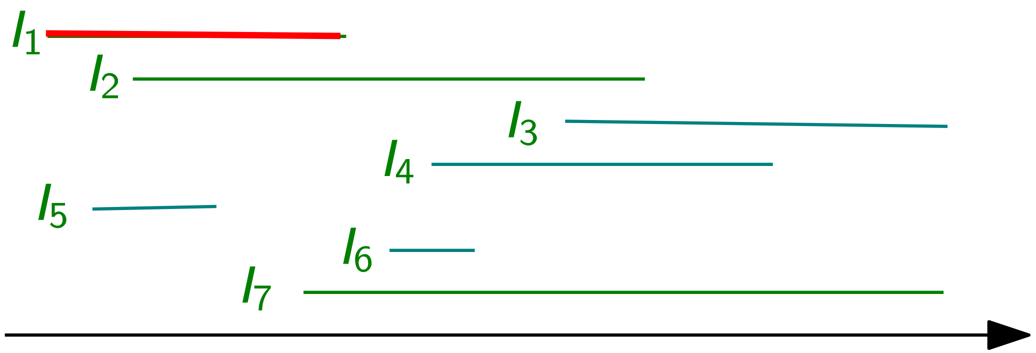
Färben von Intervallgraphen

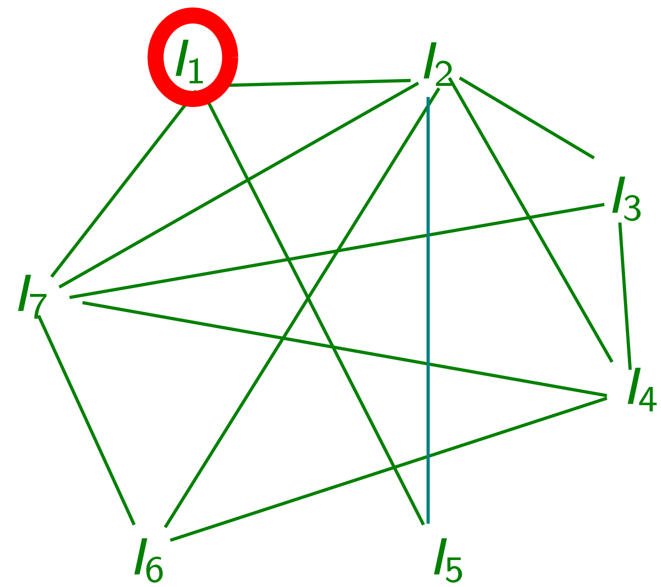




Färben von Intervallgraphen mit Greedy-Algorithmus
Ordne die Intervalle nach Anfangszeit
Färbe die Intervalle (bzw die zugehörigen Knoten) in dieser Reihenfolge mit dem Greedy-Algorithmus

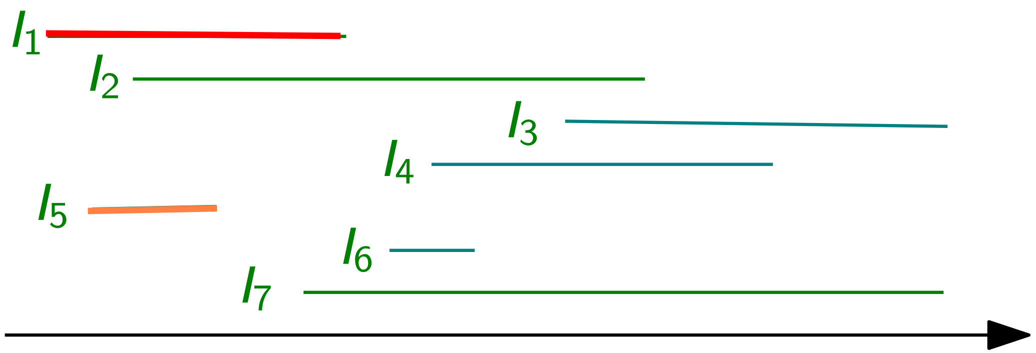
Färben von Intervallgraphen

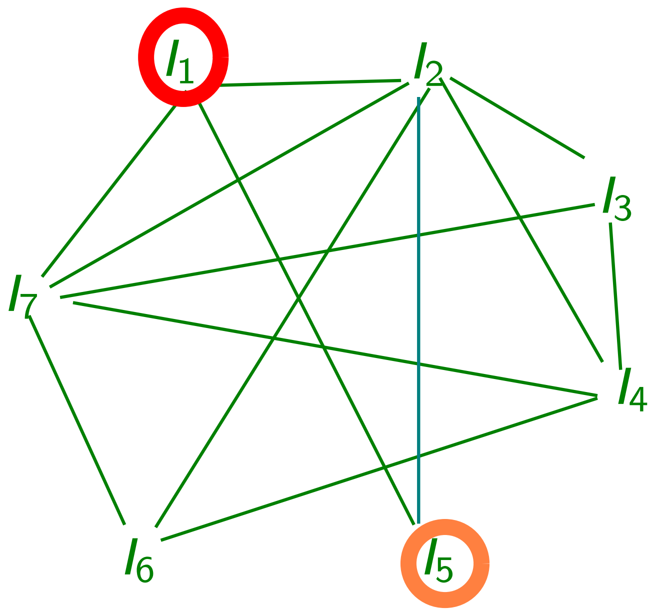




Färben von Intervallgraphen mit Greedy-Algorithmus
Ordne die Intervalle nach Anfangszeit
Färbe die Intervalle (bzw die zugehörigen Knoten) in dieser Reihenfolge mit dem Greedy-Algorithmus

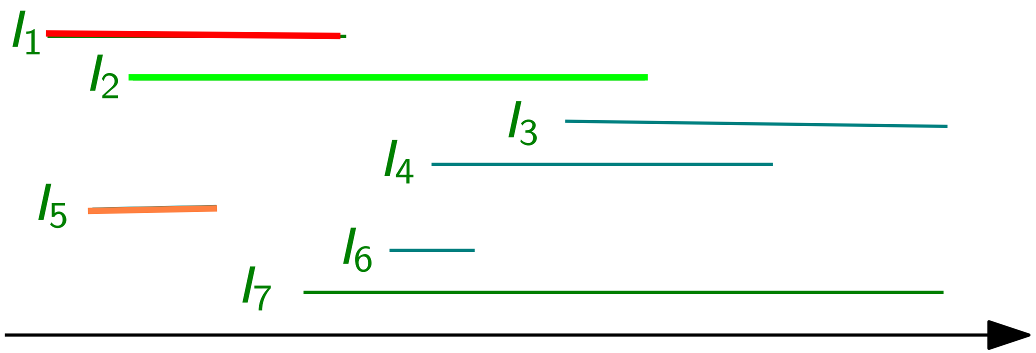
Färben von Intervallgraphen

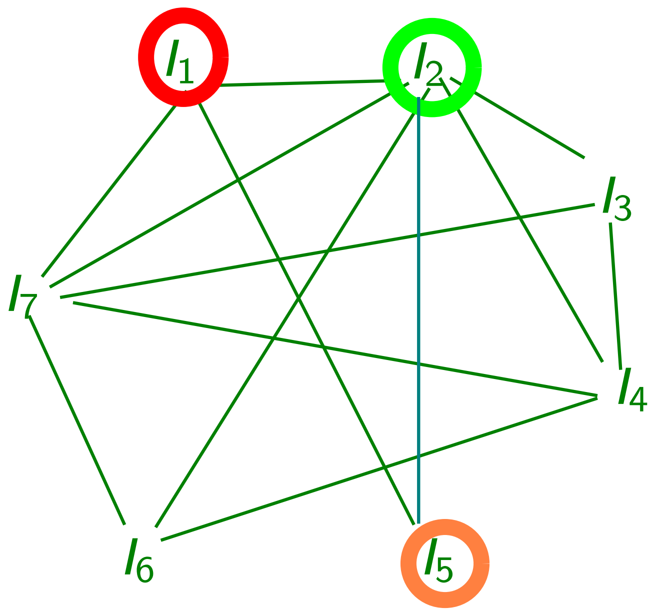




Färben von Intervallgraphen mit Greedy-Algorithmus
Ordne die Intervalle nach Anfangszeit
Färbe die Intervalle (bzw die zugehörigen Knoten) in dieser Reihenfolge mit dem Greedy-Algorithmus

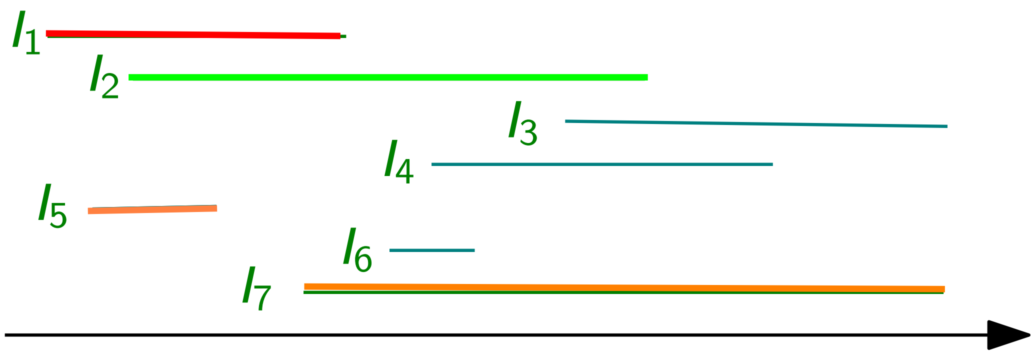
Färben von Intervallgraphen

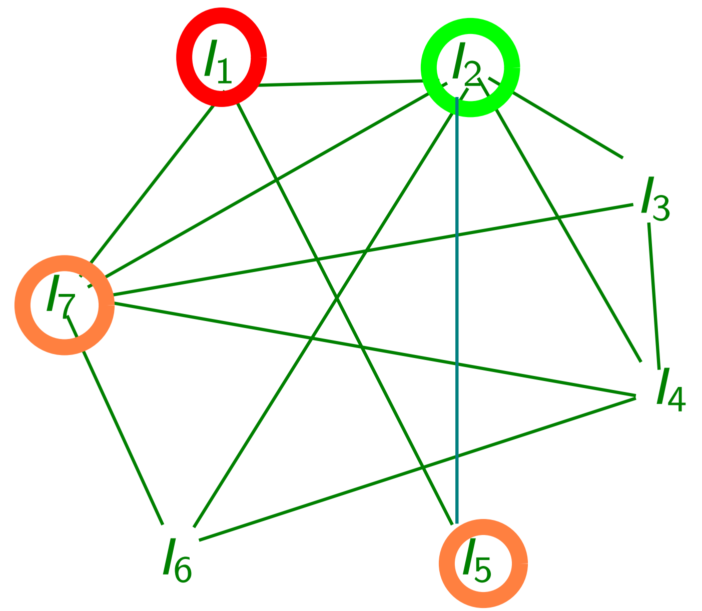




Färben von Intervallgraphen mit Greedy-Algorithmus
Ordne die Intervalle nach Anfangszeit
Färbe die Intervalle (bzw die zugehörigen Knoten) in dieser Reihenfolge mit dem Greedy-Algorithmus

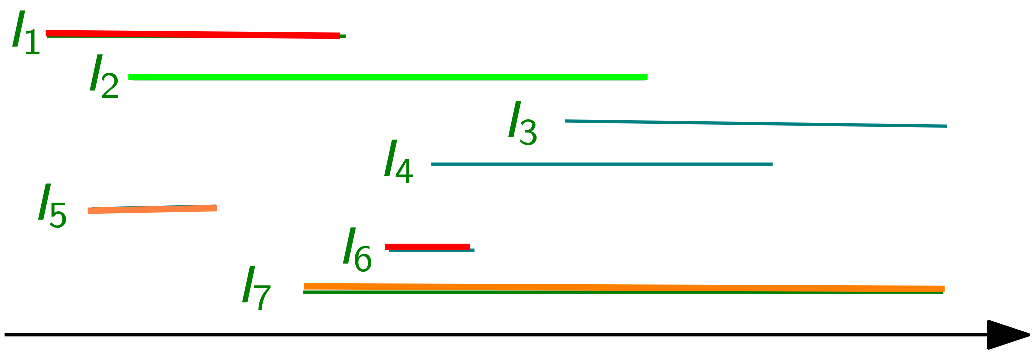
Färben von Intervallgraphen

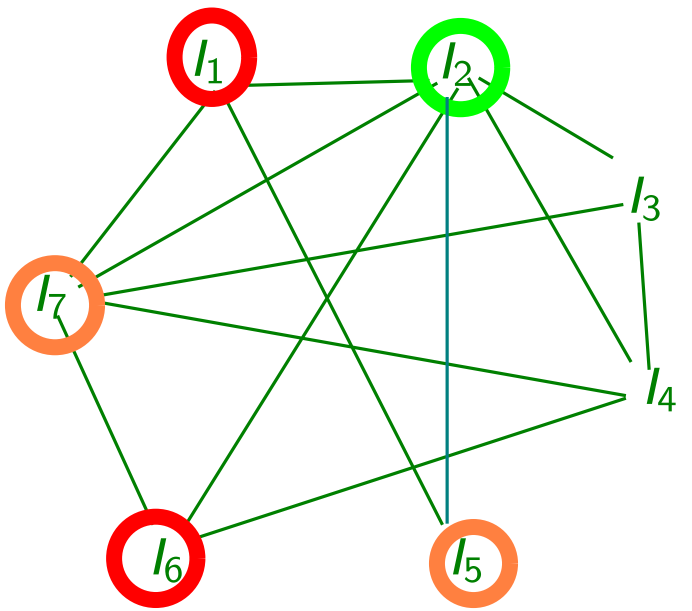




Färben von Intervallgraphen mit Greedy-Algorithmus
Ordne die Intervalle nach Anfangszeit
Färbe die Intervalle (bzw die zugehörigen Knoten) in dieser Reihenfolge mit dem Greedy-Algorithmus

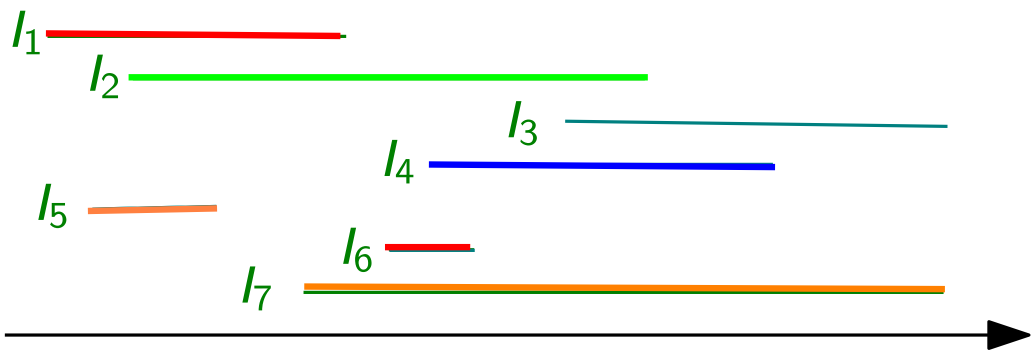
Färben von Intervallgraphen

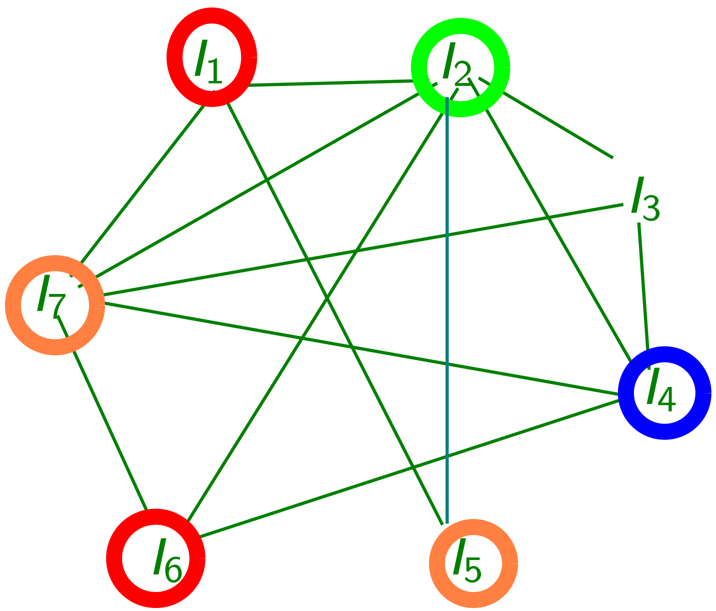




Färben von Intervallgraphen mit Greedy-Algorithmus
Ordne die Intervalle nach Anfangszeit
Färbe die Intervalle (bzw die zugehörigen Knoten) in dieser Reihenfolge mit dem Greedy-Algorithmus

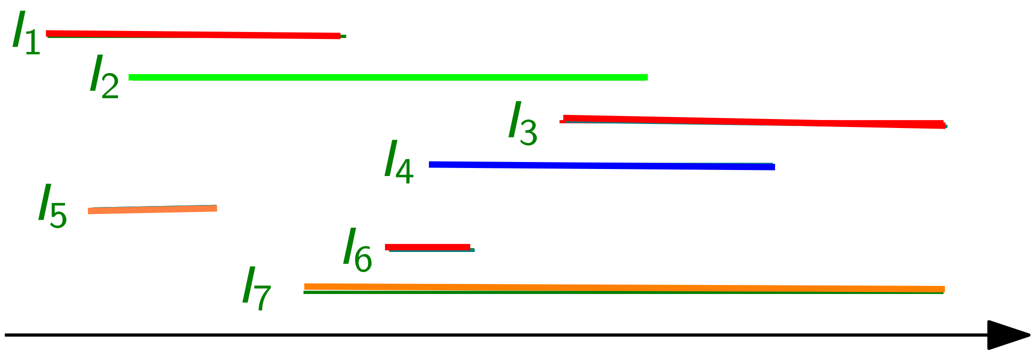
Färben von Intervallgraphen

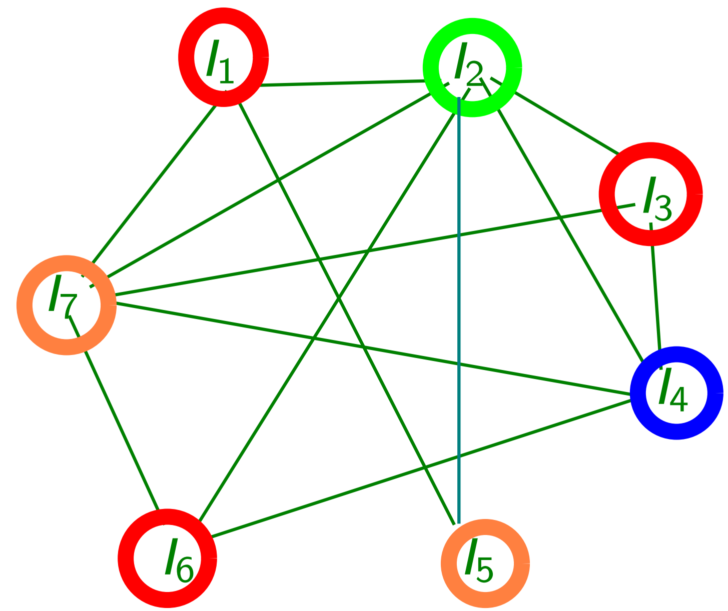




Färben von Intervallgraphen mit Greedy-Algorithmus
Ordne die Intervalle nach Anfangszeit
Färbe die Intervalle (bzw die zugehörigen Knoten) in dieser Reihenfolge mit dem Greedy-Algorithmus

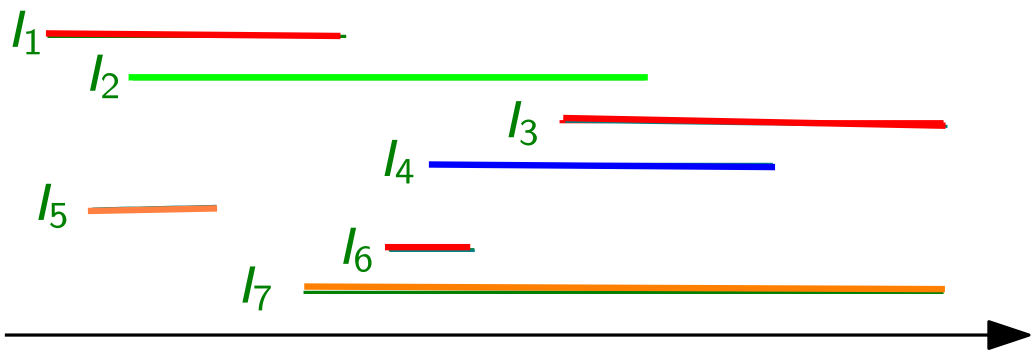
Färben von Intervallgraphen

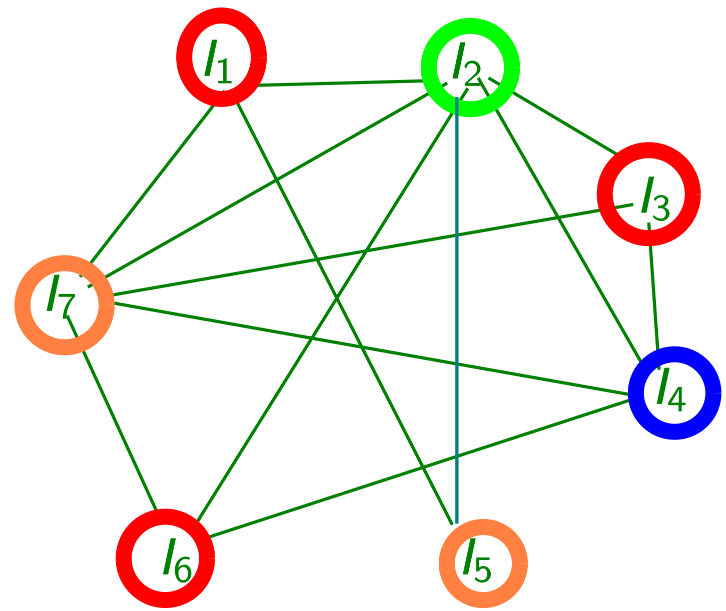




Färben von Intervallgraphen mit Greedy-Algorithmus
Ordne die Intervalle nach Anfangszeit
Färbe die Intervalle (bzw die zugehörigen Knoten) in dieser Reihenfolge mit dem Greedy-Algorithmus

Färben von Intervallgraphen

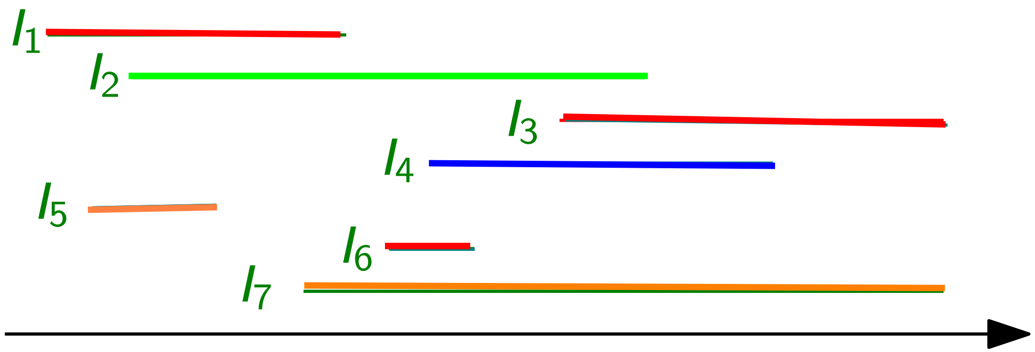


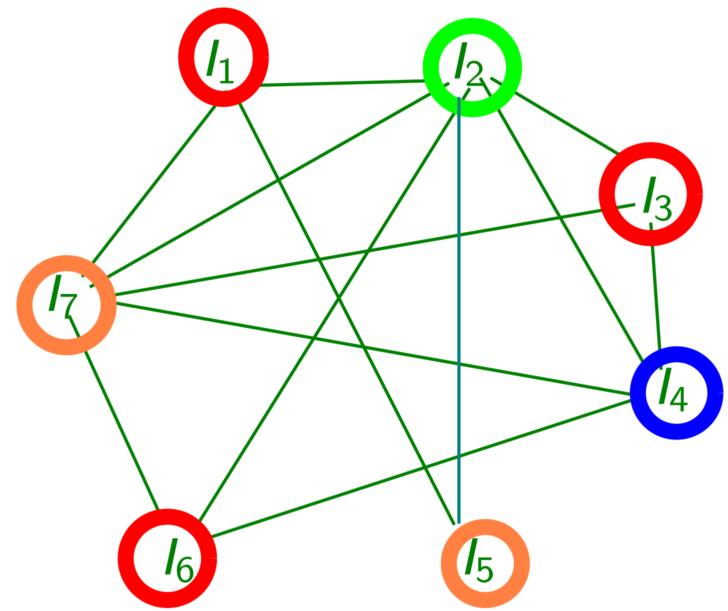


Färben von Intervallgraphen mit Greedy-Algorithmus
Ordne die Intervalle nach Anfangszeit
Färbe die Intervalle (bzw die zugehörigen Knoten) in dieser Reihenfolge mit dem Greedy-Algorithmus

Behauptung: Für diese Reihenfolge bestimmt der Greedy-Färbungsalgorithmus auf Intervallgraphen eine minimale Färbung.

Färben von Intervallgraphen



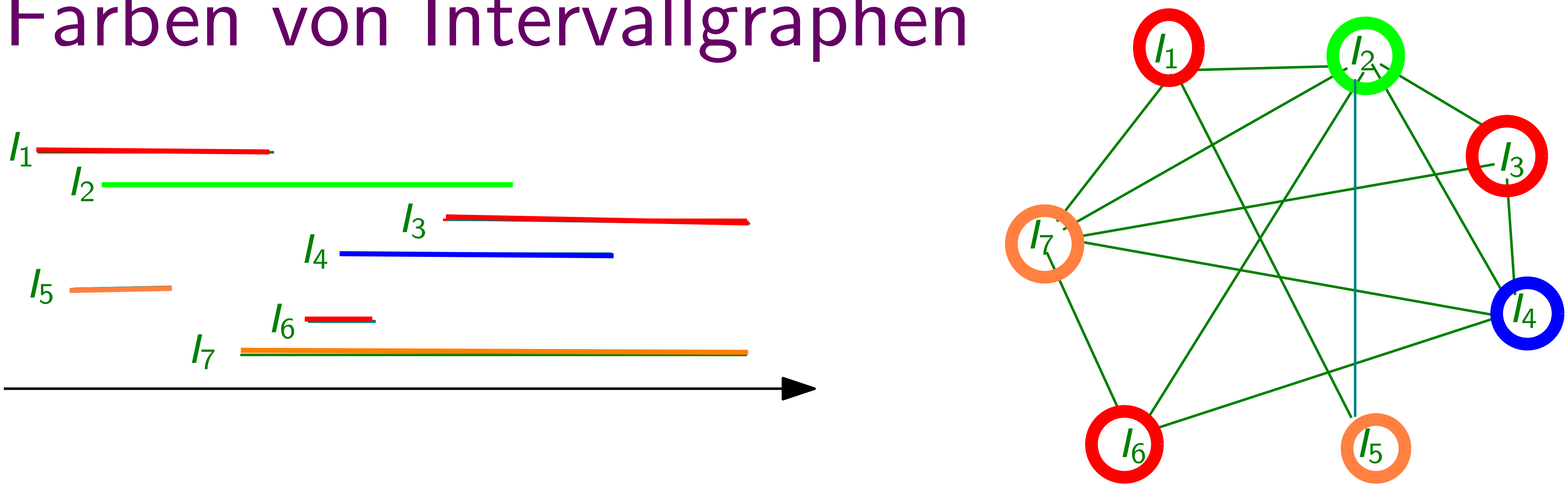


Färben von Intervallgraphen mit Greedy-Algorithmus
Ordne die Intervalle nach Anfangszeit
Färbe die Intervalle (bzw die zugehörigen Knoten) in dieser Reihenfolge mit dem Greedy-Algorithmus

Behauptung: Für diese Reihenfolge bestimmt der Greedy-Färbungsalgorithmus auf Intervallgraphen eine minimale Färbung.

Beweis: Zeige per Induktion über die Anzahl der benutzten Farben k: wird Farbe k benutzt, gibt es eine k-Clique im Graph.

Färben von Intervallgraphen



Färben von Intervallgraphen mit Greedy-Algorithmus
Ordne die Intervalle nach Anfangszeit
Färbe die Intervalle (bzw die zugehörigen Knoten) in dieser Reihenfolge mit dem Greedy-Algorithmus

Behauptung: Für diese Reihenfolge bestimmt der Greedy-Färbungsalgorithmus auf Intervallgraphen eine minimale Färbung.

Beweis: Zeige per Induktion über die Anzahl der benutzten Farben k: wird Farbe k benutzt, gibt es eine k-Clique im Graph.
Wegen $\chi(G) \geq \omega(G)$ folgt die Aussage.

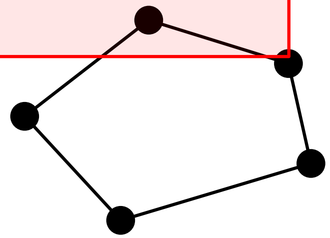
Chordale Graphen

Ein Graph $G = (V, E)$ heißt **chordal**, wenn jeder einfache Kreis der Länge ≥ 4 mindestens eine **Sehne** besitzt, d.h. eine Kante, die zwei nicht aufeinander folgende Knoten des Kreises verbindet.

Chordale Graphen

Kein Knoten wird mehr als einmal besucht.

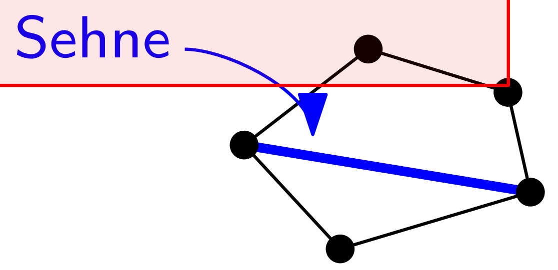
Ein Graph $G = (V, E)$ heißt **chordal**, wenn jeder $\boxed{\text{einfache}}$ Kreis der Länge ≥ 4 mindestens eine **Sehne** besitzt, d.h. eine Kante, die zwei nicht aufeinander folgende Knoten des Kreises verbindet.



Chordale Graphen

Kein Knoten wird mehr als einmal besucht.

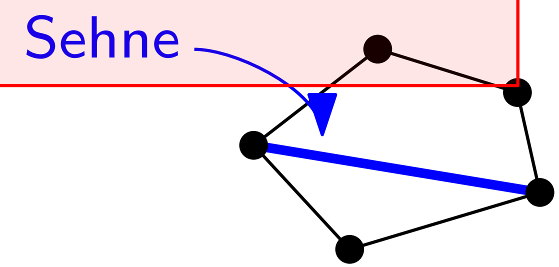
Ein Graph $G = (V, E)$ heißt **chordal**, wenn jeder $\boxed{\text{einfache}}$ Kreis der Länge ≥ 4 mindestens eine **Sehne** besitzt, d.h. eine Kante, die zwei nicht aufeinander folgende Knoten des Kreises verbindet.

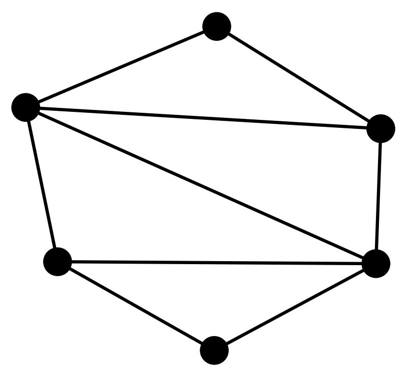


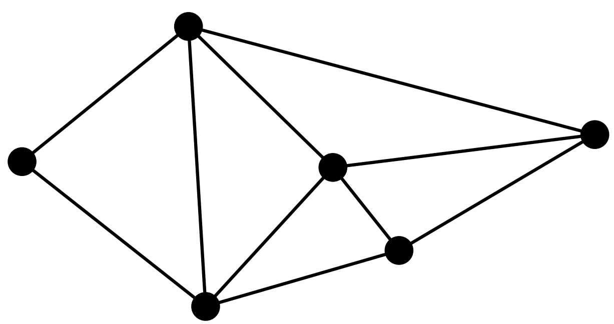
Chordale Graphen

Kein Knoten wird mehr als einmal besucht.

Ein Graph $G = (V, E)$ heißt **chordal**, wenn jeder einfache Kreis der Länge ≥ 4 mindestens eine **Sehne** besitzt, d.h. eine Kante, die zwei nicht aufeinander folgende Knoten des Kreises verbindet.





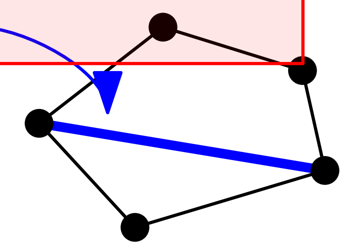


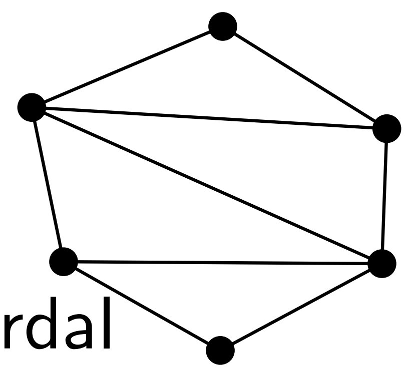
Chordale Graphen

Kein Knoten wird mehr als einmal besucht.

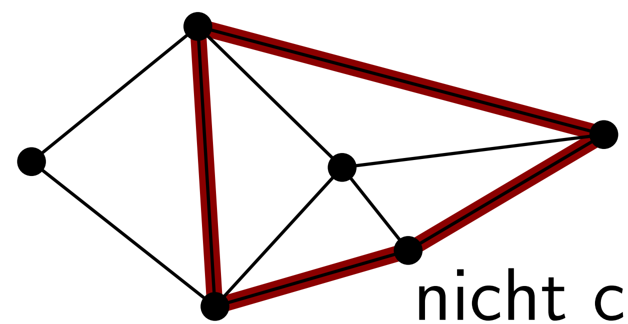
Ein Graph $G = (V, E)$ heißt **chordal**, wenn jeder einfache Kreis der Länge ≥ 4 mindestens eine **Sehne** besitzt, d.h. eine Kante, die zwei nicht aufeinander folgende Knoten des Kreises verbindet.

Sehne





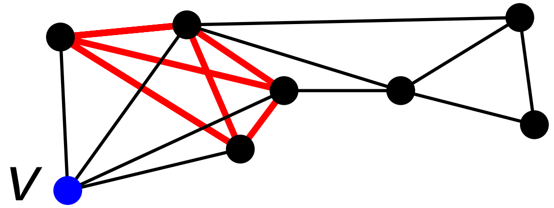
chordal



nicht chordal

Simpliziale Knoten

Knoten v heißt **simplizial**, falls seine Nachbarn $N(v)$ eine Clique in G bilden.

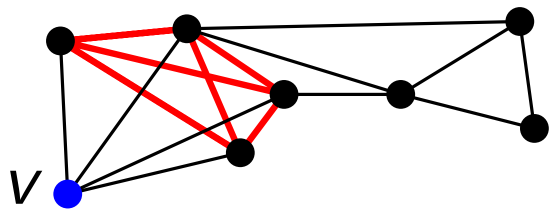


Simpliziale Knoten

Knoten v heißt **simplizial**, falls seine Nachbarn $N(v)$ eine Clique in G bilden.

Satz von Dirac
Jeder chordale Graph enthält einen simplizialen Knoten.

(Beweis: siehe zB Krumke & Noltemeier)



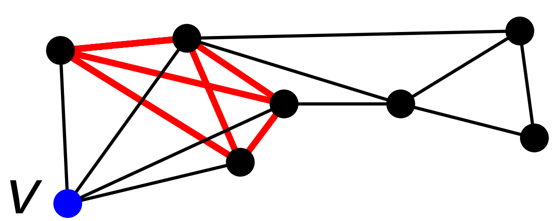
Simpliziale Knoten

Knoten v heißt **simplizial**, falls seine Nachbarn $N(v)$ eine Clique in G bilden.

Satz von Dirac
Jeder chordale Graph enthält einen simplizialen Knoten.

(Beweis: siehe zB Krumke & Noltemeier)

Sei $G = (V, E)$ ein Graph. Der **induzierte Subgraph** zur Menge $U \subset V$ ist der Graph $G = (V, E(U))$ mit $E(U) := \{uv \in E \mid u, v \in U\}$



Simpliziale Knoten

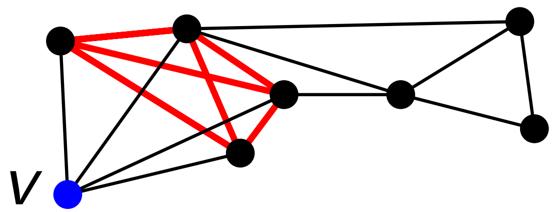
Knoten v heißt **simplizial**, falls seine Nachbarn $N(v)$ eine Clique in G bilden.

Satz von Dirac
Jeder chordale Graph enthält einen simplizialen Knoten.

(Beweis: siehe zB Krumke & Noltemeier)

Sei $G = (V, E)$ ein Graph. Der **induzierte Subgraph** zur Menge $U \subset V$ ist der Graph $G = (V, E(U))$ mit $E(U) := \{uv \in E \mid u, v \in U\}$

Beobachtung: G chordal $\Rightarrow$ jeder induzierte Subgraph von G ist chordal



Simpliziale Knoten

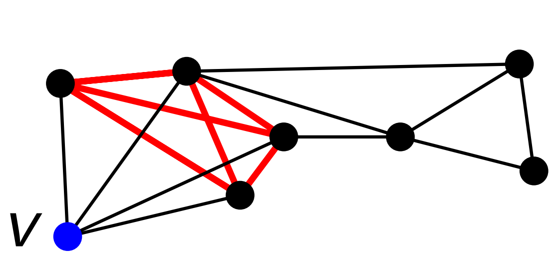
Knoten v heißt **simplizial**, falls seine Nachbarn $N(v)$ eine Clique in G bilden.

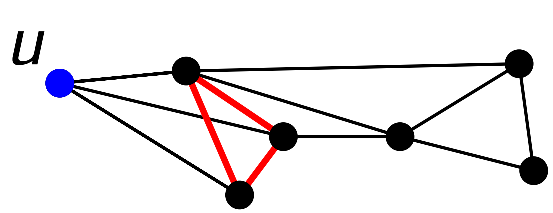
Satz von Dirac
Jeder chordale Graph enthält einen simplizialen Knoten.

(Beweis: siehe zB Krumke & Noltemeier)

Sei $G = (V, E)$ ein Graph. Der **induzierte Subgraph** zur Menge $U \subset V$ ist der Graph $G = (V, E(U))$ mit $E(U) := \{uv \in E \mid u, v \in U\}$

Beobachtung: G chordal $\Rightarrow$ jeder induzierte Subgraph von G ist chordal
$\Rightarrow$ $G - v$ enthält ebenfalls einen simplizialen Knoten u





Simpliziale Knoten

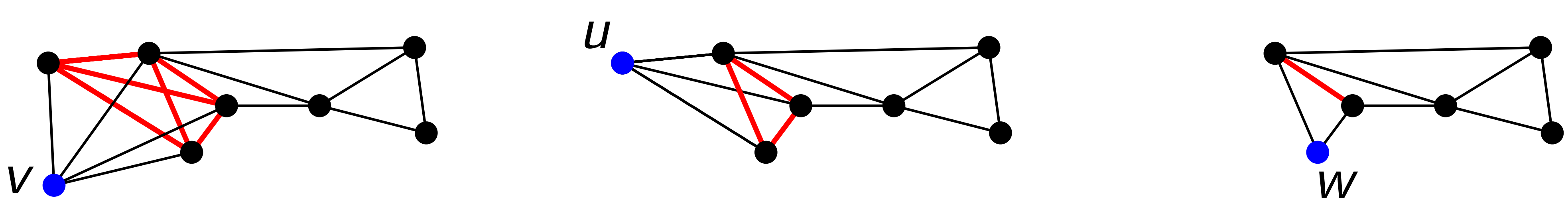
Knoten v heißt **simplizial**, falls seine Nachbarn $N(v)$ eine Clique in G bilden.

Satz von Dirac
Jeder chordale Graph enthält einen simplizialen Knoten.

(Beweis: siehe zB Krumke & Noltemeier)

Sei $G = (V, E)$ ein Graph. Der **induzierte Subgraph** zur Menge $U \subset V$ ist der Graph $G = (V, E(U))$ mit $E(U) := \{uv \in E \mid u, v \in U\}$

Beobachtung: G chordal $\Rightarrow$ jeder induzierte Subgraph von G ist chordal
$\Rightarrow$ $G - v$ enthält ebenfalls einen simplizialen Knoten u
und $G - \{v, u\}$ enthält simplizialen Knoten w, ...



Perfektes Eliminationsschema

Eine Nummerierung $(v_1, \ldots, v_n)$ der Knotenmenge V heißt **perfektes Eliminationsschema**, wenn für $i = 1, \ldots, n$ gilt: v_i ist simplizial in $G[\{v_i, \ldots, v_n\}]$.

Perfektes Eliminationsschema

Eine Nummerierung $(v_1, \ldots, v_n)$ der Knotenmenge V heißt **perfektes Eliminationsschema**, wenn für $i = 1, \ldots, n$ gilt: v_i ist simplizial in $G[\{v_i, \ldots, v_n\}]$.

v_1 ist simplizial in $G[V] = G$

Perfektes Eliminationsschema

Eine Nummerierung $(v_1, \ldots, v_n)$ der Knotenmenge V heißt **perfektes Eliminationsschema**, wenn für $i = 1, \ldots, n$ gilt: v_i ist simplizial in $G[\{v_i, \ldots, v_n\}]$.

v_1 ist simplizial in $G[V] = G$

v_2 ist simplizial in $G[V - v_1] = G - v_1$

Perfektes Eliminationsschema

Eine Nummerierung $(v_1, \ldots, v_n)$ der Knotenmenge V heißt **perfektes Eliminationsschema**, wenn für $i = 1, \ldots, n$ gilt: v_i ist simplizial in $G[\{v_i, \ldots, v_n\}]$.

v_1 ist simplizial in $G[V] = G$
v_2 ist simplizial in $G[V - v_1] = G - v_1$
v_3 ist simplizial in $G - \{v_1, v_2\}$ usw.

Perfektes Eliminationsschema

Eine Nummerierung $(v_1, \ldots, v_n)$ der Knotenmenge V heißt **perfektes Eliminationsschema**, wenn für $i = 1, \ldots, n$ gilt: v_i ist simplizial in $G[\{v_i, \ldots, v_n\}]$.

v_1 ist simplizial in $G[V] = G$
v_2 ist simplizial in $G[V - v_1] = G - v_1$
v_3 ist simplizial in $G - \{v_1, v_2\}$ usw.

Satz: G chordal $\Leftrightarrow$ G hat perfektes Eliminationsschema

Perfektes Eliminationsschema

Eine Nummerierung $(v_1, \ldots, v_n)$ der Knotenmenge V heißt **perfektes Eliminationsschema**, wenn für $i = 1, \ldots, n$ gilt: v_i ist simplizial in $G[\{v_i, \ldots, v_n\}]$.

v_1 ist simplizial in $G[V] = G$
v_2 ist simplizial in $G[V - v_1] = G - v_1$
v_3 ist simplizial in $G - \{v_1, v_2\}$ usw.

Satz: G chordal $\Leftrightarrow$ G hat perfektes Eliminationsschema

Beweis. „$\Rightarrow$“ Satz von Dirac (siehe oben)

Perfektes Eliminationsschema

Eine Nummerierung $(v_1, \ldots, v_n)$ der Knotenmenge V heißt **perfektes Eliminationsschema**, wenn für $i = 1, \ldots, n$ gilt: v_i ist simplizial in $G[\{v_i, \ldots, v_n\}]$.

v_1 ist simplizial in $G[V] = G$
v_2 ist simplizial in $G[V - v_1] = G - v_1$
v_3 ist simplizial in $G - \{v_1, v_2\}$ usw.

Satz: G chordal $\Leftrightarrow$ G hat perfektes Eliminationsschema

Beweis. „$\Rightarrow$“ Satz von Dirac (siehe oben)

„$\Leftarrow$“ Sei K einfacher Kreis der Länge ≥ 4.

Perfektes Eliminationsschema

Eine Nummerierung $(v_1, \dots, v_n)$ der Knotenmenge V heißt **perfektes Eliminationsschema**, wenn für $i = 1, \dots, n$ gilt: v_i ist simplizial in $G[\{v_i, \dots, v_n\}]$.

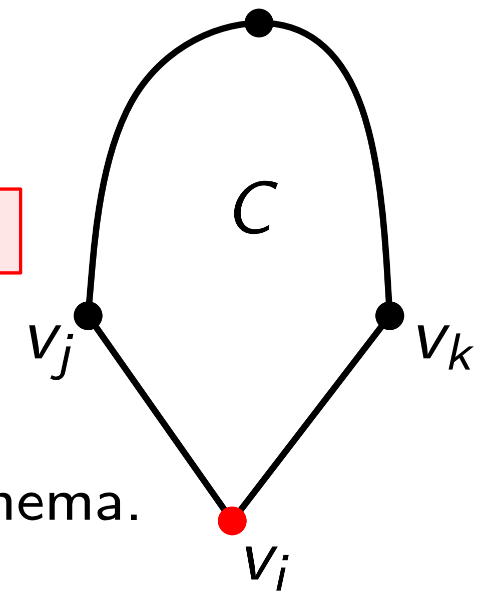
v_1 ist simplizial in $G[V] = G$
v_2 ist simplizial in $G[V - v_1] = G - v_1$
v_3 ist simplizial in $G - \{v_1, v_2\}$ usw.

Satz: G chordal $\Leftrightarrow$ G hat perfektes Eliminationsschema

Beweis. „$\Rightarrow$“ Satz von Dirac (siehe oben)

„$\Leftarrow$“ Sei K einfacher Kreis der Länge ≥ 4.

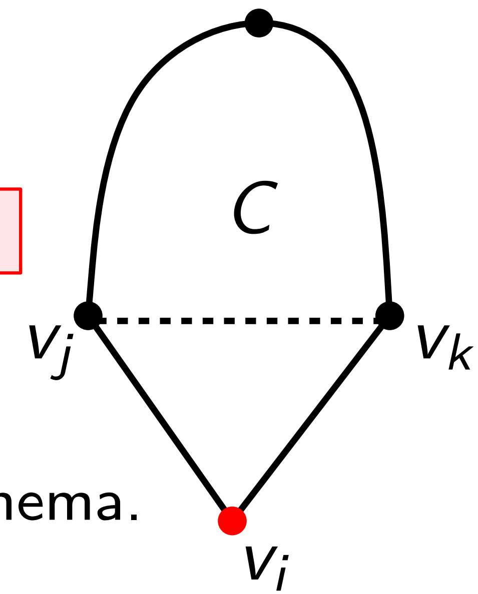
Sei v_i Knoten mit kleinster Nummer i im Eliminationsschema.



Perfektes Eliminationsschema

Eine Nummerierung $(v_1, \ldots, v_n)$ der Knotenmenge V heißt **perfektes Eliminationsschema**, wenn für $i = 1, \ldots, n$ gilt: v_i ist simplizial in $G[\{v_i, \ldots, v_n\}]$.

v_1 ist simplizial in $G[V] = G$
v_2 ist simplizial in $G[V - v_1] = G - v_1$
v_3 ist simplizial in $G - \{v_1, v_2\}$ usw.



Satz: G chordal $\Leftrightarrow$ G hat perfektes Eliminationsschema

Beweis. „$\Rightarrow$“ Satz von Dirac (siehe oben)

„$\Leftarrow$“ Sei K einfacher Kreis der Länge ≥ 4.

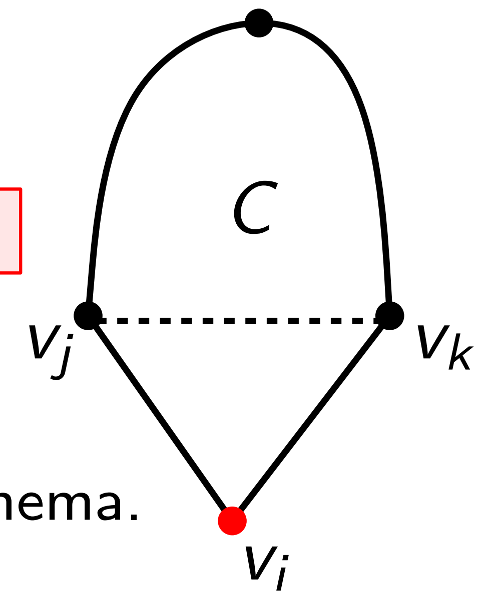
Sei v_i Knoten mit kleinster Nummer i im Eliminationsschema.

Nachbarn v_j, v_k von v_i auf K sind adjazent in G,
da v_i simplizial in $G[v_i, \ldots, v_n]$ und $j, k \geq i$. $\square$

Perfektes Eliminationsschema

Eine Nummerierung $(v_1, \ldots, v_n)$ der Knotenmenge V heißt **perfektes Eliminationsschema**, wenn für $i = 1, \ldots, n$ gilt: v_i ist simplizial in $G[\{v_i, \ldots, v_n\}]$.

v_1 ist simplizial in $G[V] = G$
v_2 ist simplizial in $G[V - v_1] = G - v_1$
v_3 ist simplizial in $G - \{v_1, v_2\}$ usw.



Satz: G chordal $\Leftrightarrow$ G hat perfektes Eliminationsschema

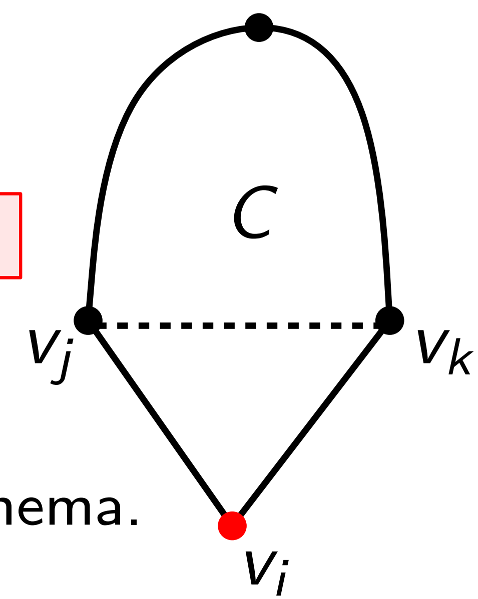
Beweis. „$\Rightarrow$“ Satz von Dirac (siehe oben)
„$\Leftarrow$“ Sei K einfacher Kreis der Länge ≥ 4.
Sei v_i Knoten mit kleinster Nummer i im Eliminationsschema.
Nachbarn v_j, v_k von v_i auf K sind adjazent in G,
da v_i simplizial in $G[v_i, \ldots, v_n]$ und $j, k \geq i$. $\square$

Wie berechnet man perfektes Eliminationsschema?

Perfektes Eliminationsschema

Eine Nummerierung $(v_1, \ldots, v_n)$ der Knotenmenge V heißt **perfektes Eliminationsschema**, wenn für $i = 1, \ldots, n$ gilt: v_i ist simplizial in $G[\{v_i, \ldots, v_n\}]$.

v_1 ist simplizial in $G[V] = G$
v_2 ist simplizial in $G[V - v_1] = G - v_1$
v_3 ist simplizial in $G - \{v_1, v_2\}$ usw.



Satz: G chordal $\Leftrightarrow$ G hat perfektes Eliminationsschema

Beweis. „$\Rightarrow$" Satz von Dirac (siehe oben)
„$\Leftarrow$" Sei K einfacher Kreis der Länge ≥ 4.
Sei v_i Knoten mit kleinster Nummer i im Eliminationsschema.
Nachbarn v_j, v_k von v_i auf K sind adjazent in G,
da v_i simplizial in $G[v_i, \ldots, v_n]$ und $j, k \geq i$. $\square$

Wie berechnet man perfektes Eliminationsschema?

Perfektes Eliminationsschema lässt sich in Polynomialzeit errechnen.
(naiv: $O(|V|^4) = O(|V| \cdot |V| \cdot |V|^2)$, clever: in Linearzeit)

Färben von chordalen Graphen

Färben von chordalen Graphen mit Greedy-Algorithmus
Finde ein perfektes Eliminationsschema $(v_1, v_2, \ldots, v_n)$
Färbe die Knoten in Reihenfolge $v_n, v_{n-1}, \ldots, v_1$ mit dem Greedy-Algorithmus

Färben von chordalen Graphen

Färben von chordalen Graphen mit Greedy-Algorithmus
Finde ein perfektes Eliminationsschema $(v_1, v_2, \ldots, v_n)$
Färbe die Knoten in Reihenfolge $v_n, v_{n-1}, \ldots, v_1$ mit dem Greedy-Algorithmus

Behauptung: Für diese Reihenfolge bestimmt der Greedy-Färbungsalgorithmus auf chordalen Graphen eine minimale Färbung.

Färben von chordalen Graphen

Färben von chordalen Graphen mit Greedy-Algorithmus
Finde ein perfektes Eliminationsschema $(v_1, v_2, \ldots, v_n)$
Färbe die Knoten in Reihenfolge $v_n, v_{n-1}, \ldots, v_1$ mit dem Greedy-Algorithmus

Behauptung: Für diese Reihenfolge bestimmt der Greedy-Färbungsalgorithmus auf chordalen Graphen eine minimale Färbung.

Beweis: Induktion über i (von n nach 1):

Da $v_i \cup (N(v_i) \cap \{v_{i+1}, \ldots, v_n\}) =: C_i$ Clique, gilt $|C_i| \leq \omega(G)$.

Also $|N(v_i) \cap \{v_{i+1}, \ldots, v_n\}| \leq \omega(G) - 1$.

$\Rightarrow$ Können v_i stets mit einer der Farben $\{1, \ldots, \omega(G)\}$ färben.

Wegen $\omega(G) \leq \chi(G)$ gilt, dass unsere Färbung optimal ist!

Färben von planare Graphen

Sei $G = (V, E)$ ein Graph.
Wir nennen G **planar**, wenn wir ihn (in der Ebene) zeichnen können, ohne dass sich Kanten kreuzen.

Färben von planare Graphen

Sei $G = (V, E)$ ein Graph.
Wir nennen G **planar**, wenn wir ihn (in der Ebene) zeichnen können, ohne dass sich Kanten kreuzen.

(siehe zum Beispiel Krumke & Noltemeier)

Man kann beweisen, dass jeder planare Graph mindestens einen Knoten mit Knotengrad ≤ 5 hat

Färben von planare Graphen

Sei $G = (V, E)$ ein Graph.
Wir nennen G **planar**, wenn wir ihn (in der Ebene) zeichnen können, ohne dass sich Kanten kreuzen.

(siehe zum Beispiel Krumke & Noltemeier)

Man kann beweisen, dass jeder planare Graph mindestens einen Knoten mit Knotengrad ≤ 5 hat

Färben von planare Graphen

Sei $G = (V, E)$ ein Graph.
Wir nennen G **planar**, wenn wir ihn (in der Ebene) zeichnen können, ohne dass sich Kanten kreuzen.

(siehe zum Beispiel Krumke & Noltemeier)

Man kann beweisen, dass jeder planare Graph mindestens einen Knoten mit Knotengrad ≤ 5 hat

Wie kann man diese Einsicht nutzen, um planare Graphen mit dem Greedy-Algorithmus in 6 Farben zu färben?

Färben von planare Graphen

Sei $G = (V, E)$ ein Graph.
Wir nennen G **planar**, wenn wir ihn (in der Ebene) zeichnen können, ohne dass sich Kanten kreuzen.

(siehe zum Beispiel Krumke & Noltemeier)

Man kann beweisen, dass jeder planare Graph mindestens einen Knoten mit Knotengrad ≤ 5 hat

Färben von planaren Graphen mit Greedy-Algorithmus
Bestimme Eliminationsschema:
- Finde Knoten v_1 mit Knotengrad ≤ 5 in G,
- Finde Knoten v_2 mit Knotengrad ≤ 5 in $G - \{v_1\}$,
- Finde Knoten v_3 mit Knotengrad ≤ 5 in $G - \{v_1, v_2\}$,- ...

Färbe die Knoten in Reihenfolge $v_n, v_{n-1}, \ldots, v_1$ mit dem Greedy-Algorithmus

Färben von planare Graphen

Sei $G = (V, E)$ ein Graph.
Wir nennen G **planar**, wenn wir ihn (in der Ebene) zeichnen können, ohne dass sich Kanten kreuzen.

(siehe zum Beispiel Krumke & Noltemeier)

Man kann beweisen, dass jeder planare Graph mindestens einen Knoten mit Knotengrad ≤ 5 hat

Färben von planaren Graphen mit Greedy-Algorithmus
Bestimme Eliminationsschema:
- Finde Knoten v_1 mit Knotengrad ≤ 5 in G,
- Finde Knoten v_2 mit Knotengrad ≤ 5 in $G - \{v_1\}$,
- Finde Knoten v_3 mit Knotengrad ≤ 5 in $G - \{v_1, v_2\}$,- ...

Färbe die Knoten in Reihenfolge $v_n, v_{n-1}, \ldots, v_1$ mit dem Greedy-Algorithmus

Beh: Für diese Reihenfolge bestimmt der Greedy-Färbungsalgorithmus auf planaren Graphen eine Färbung mit höchstens 6 Farben.

Färben von planare Graphen

Sei $G = (V, E)$ ein Graph.
Wir nennen G **planar**, wenn wir ihn (in der Ebene) zeichnen können, ohne dass sich Kanten kreuzen.

(siehe zum Beispiel Krumke & Noltemeier)

Man kann beweisen, dass jeder planare Graph mindestens einen Knoten mit Knotengrad ≤ 5 hat

Färben von planaren Graphen mit Greedy-Algorithmus
Bestimme Eliminationsschema:
- Finde Knoten v_1 mit Knotengrad ≤ 5 in G,
- Finde Knoten v_2 mit Knotengrad ≤ 5 in $G - \{v_1\}$,
- Finde Knoten v_3 mit Knotengrad ≤ 5 in $G - \{v_1, v_2\}$,- ...

Färbe die Knoten in Reihenfolge $v_n, v_{n-1}, \ldots, v_1$ mit dem Greedy-Algorithmus

Beh: Für diese Reihenfolge bestimmt der Greedy-Färbungsalgorithmus auf planaren Graphen eine Färbung mit höchstens 6 Farben.

Beweis: Jeder betrachtete Knoten v_i hat in $G - \{v_{i+1}, \ldots, v_n\}$ höchstens 5 Nachbarn - deswegen ist noch eine Farbe verfügbar.

Färben von planaren Graphen

6-Farbensatz

Jeder planare Graph ist mit 6 Farben färbbar.

Beweis: Siehe vorherige Folie.

Färben von planaren Graphen

6-Farbensatz
Jeder planare Graph ist mit 6 Farben färbbar.
Beweis: Siehe vorherige Folie.

5-Farbensatz
Jeder planare Graph ist mit 5 Farben färbbar.
Beweis: Siehe zum Beispiel Krumke&Noltemeier

Färben von planaren Graphen

6-Farbensatz

Jeder planare Graph ist mit 6 Farben färbbar.

Beweis: Siehe vorherige Folie.

5-Farbensatz

Jeder planare Graph ist mit 5 Farben färbbar.

Beweis: Siehe zum Beispiel Krumke&Noltemeier

4-Farbensatz

Jeder planare Graph ist mit 4 Farben färbbar.

Erster großer Satz, der mit Computerbeweis bewiesen wurde.

Färben von planaren Graphen

6-Farbensatz
Jeder planare Graph ist mit 6 Farben färbbar.
Beweis: Siehe vorherige Folie.

5-Farbensatz
Jeder planare Graph ist mit 5 Farben färbbar.
Beweis: Siehe zum Beispiel Krumke&Noltemeier

4-Farbensatz
Jeder planare Graph ist mit 4 Farben färbbar.
Erster großer Satz, der mit Computerbeweis bewiesen wurde.

Es gibt planare Graphen, für deren Färbung man mindestens 4 Farben braucht.
Beispiel?

Greedy-Algorithmus zum Graphenfärben

Für welche Graphen können wir leicht eine gute Färbungsreihenfolge für den Greedy-Algorithmus angeben?

- Pfade
- Bäume
- Kreise
- bipartite Graphen
- Intervallgraphen
- chordale Graphen

- planare Graphen (für eine 6-Färbung)

Stichworte heute

Konzepte: Clique, Cliquenzahl, Färbung, chromatische Zahl, Intervallgraph, chordaler Graph, Sehne, perfektes Eliminationsschema, planar, induzierter Subgraph

Optimierungsprobleme: größte Clique, minimale Färbung

Modellierung: Frequenzzuweisungen, Maschinenbelegung

Algorithmen: Greedy-Färbungsalgorithmus