

1. Du kannst mindestens vier Anwendungen der Graphentheorie zur Modellierung und Lösung praktischer Aufgaben aufzählen.
2. Du kannst gerichtete und ungerichtete Graphen aufgrund ihrer formalen Darstellung  $G = (V, E)$  skizzieren und umgekehrt anhand des Bildes eines Graphen formal seine Knoten- und Kantenmenge angeben.
3. Du kannst gerichtete und ungerichtete Graphen als Adjazenzmatrix und als Adjazenzliste darstellen und umgekehrt aus diesen Darstellungen den Graphen rekonstruieren.
4. Du kannst einen gerichteten oder ungerichteten Graphen in Python als *Dictionary of Sets* darstellen.
5. Du kannst die beiden im Unterricht behandelten Darstellungen (Adjazenzliste und Adjazenzmatrix) von Graphen bezüglich der Laufzeit folgender Operationen vergleichen.
  - (a) die Speicherung des Graphen  $G = (V, E)$ ,
6. Du kannst in Graphen *Schlingen*, *isolierte Ecken*, *Inzidenz* und *Adjazenz* erkennen.
7. Du kannst erkennen, ob ein ungerichteter Graph *zusammenhängend* oder *nicht zusammenhängend* ist und die Anzahl seiner Zusammenhangskomponenten angeben.
8. Du kannst beschreiben, was ein *vollständiger Graph* ist und berechnen, wie viele Kanten der vollständige Graph mit  $n$ -Knoten – der üblicherweise mit  $K_n$  bezeichnet wird – hat.
9. Du kannst *Cliquen* und *maximale Cliquen* in einem Graphen  $G$  erkennen und die *Cliquenzahl* (die Anzahl der Knoten einer Clique) angeben.
10. Du kannst in ungerichteten Graphen  $G = (V, E)$  für  $v \in V$  den Grad der Knoten  $\deg(v)$  und den Grad des Graphen  $\deg(G) = \sum_{v \in V} \deg(v)$  bestimmen.
11. Du kannst den Zusammenhang zwischen dem Grad eines ungerichteten Graphen (d. h. der Summe seiner Knotengrade) formulieren und damit erkennen, ob die Angaben über die Anzahl der Kanten und der Knotengrade widersprüchlich sind.
12. Du kannst für einen zusammenhängenden Graphen bzw. seiner Adjazenzlistendarstellung eine Tiefensuche durchführen und angeben, in welcher Reihenfolge die Knoten besucht werden.
13. Du kannst Probleme des täglichen Lebens angeben, zu deren Lösung die *topologische Sortierung* verwendet werden kann.
14. Du kannst für einen gerichteten Graphen  $G$  eine topologische Sortierung mit dem *Algorithmus von Kahn* durchführen.