# 1 Einleitung

Graphen sind hervorragend zur anschaulichen Darstellung von Beziehungen geeignet. Sehen wir uns ein Beispiel aus dem Alltag an: Ein Lehrer möchte mit seiner Klasse für ein paar Tage in eine Jugendherberge fahren. Nun überlegt er, wie viele Zimmer er reservieren muss, damit die Tage der Klassenfahrt für ihn so stressfrei wie möglich werden.

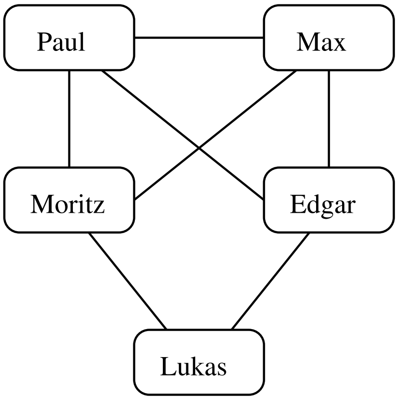


**Abb. 1.1.** Ein Graph, der zeigt, welche Schülerpaare nicht dasselbe Zimmer belegen sollten.

Klar ist, dass Mädchen und Jungen in verschiedenen Zimmern untergebracht werden müssen. Aber auch innerhalb der Gruppe der Schülerinnen und der Gruppe der Schüler gibt es Kombinationen von Kindern, die – aus unterschiedlichen Gründen – nicht unbedingt im selben Zimmer übernachten sollten. Zum Beispiel prügelt sich Paul ständig mit Max und Moritz, Max und Moritz hecken zusammen gern böse Streiche aus, Edgar ist genau wie Paul in Lisa verliebt (weshalb sich die beiden Jungen nicht ausstehen können), Max ist charakterschwach und lässt sich von Edgar sehr einfach zu übertriebenem Alkoholkonsum verführen, und Steffens Besserwisserei nervt besonders Edgar und Moritz. Die Konfliktbeziehungen zwischen

F. Gurski et al., *Exakte Algorithmen für schwere Graphenprobleme*, eXamen.press,
DOI 10.1007/978-3-642-04500-4_1, © Springer-Verlag Berlin Heidelberg 2010

diesen fünf[1] Schülern sind in Abb. 1.1 graphisch dargestellt: Zwei Schüler sind genau dann miteinander verbunden, wenn sie nach Meinung des Lehrers besser nicht gemeinsam ein Zimmer beziehen sollten.

Wie viele Herbergszimmer sollte der Lehrer nun reservieren, um alle konfliktträchtigen Kombinationen zu vermeiden? (Dabei nehmen wir an, dass alle Zimmer eine hinreichend große Zahl von Betten haben; in unserem kleinen Beispiel genügen Doppelzimmer.)

Abbildung 1.2 zeigt eine mögliche Zimmeraufteilung, die dem Lehrer das Leben einfacher macht: Paul geht in Zimmer 1, Max in Zimmer 2 und so weiter. Diese Zimmeraufteilung entspricht einer Färbung des Graphen, denn in dieser Jugendherberge haben alle Zimmer unterschiedliche Farben: Zimmer 1 ist rot, Zimmer 2 ist blau und Zimmer 3 ist grün. Das Problem des Lehrers ist also nichts anderes als ein Graphfärbbarkeitsproblem: Können alle Jungen, die im Graphen in Abb. 1.1 miteinander verbunden sind, auf möglichst wenige (verschiedenfarbige) Zimmer verteilt werden? Denn natürlich möchte der Lehrer mit möglichst wenigen Farben bzw. Zimmern auskommen, um möglichst sparsam mit dem für die Klassenfahrt gesammelten Geld umzugehen.

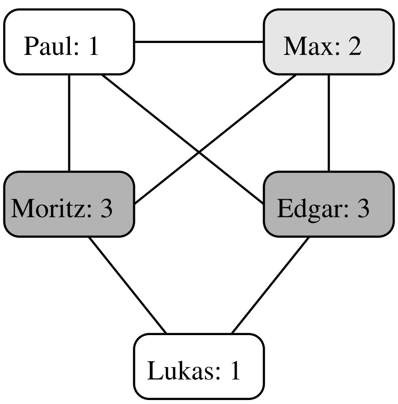


**Abb. 1.2.** Eine mögliche Zimmeraufteilung, die Konflikte vermeidet.

Der Graph in Abb. 1.2 zeigt, dass er sein Problem für diese fünf Schüler mit drei Farben bzw. Zimmern lösen kann – zwei wären jedoch zu wenig. Ist die Anzahl $k$ der Zimmer (bzw. Farben) vorgegeben, die der Lehrer höchstens reservieren (bzw. verwenden) kann, so erhält man statt des gerade beschriebenen Minimierungsproblems das zugehörige Entscheidungsproblem: Können alle Jungen, die im Graphen in Abb. 1.1 miteinander verbunden sind, auf höchstens $k$ (verschiedenfarbige) Zimmer verteilt werden? Wenn die Antwort auf diese Frage „ja“ ist, so nennt man den

[1] Der Einfachheit halber beschränken wir uns auf diese fünf und zeigen nicht den gesamten Graphen mit allen 32 Schülerinnen und Schülern der Klasse 8c des Goethe-Gymnasiums.

entsprechenden Graphen $k$-färbbar. Auf dieses Beispiel werden wir später im Buch (vor allem in Kapitel 7) noch zurückgreifen.

Das Problem, ob ein gegebener Graph $k$-färbbar (z. B. dreifärbbar) ist, ist eines der vielen Graphenprobleme, die in diesem Buch untersucht werden. Kleine Instanzen solcher Probleme (wie unser überschaubares Beispiel des Färbbarkeitsproblems für Graphen oben) lassen sich gerade noch so „im Kopf" lösen. Schon für etwas größere Probleminstanzen erweist sich eine Lösung als recht schwierig. Für noch größere Instanzen (etwa für die gesamte 32-köpfige Klasse des Lehrers in diesem Beispiel, und erst recht für Graphen mit mehreren hundert Knoten) benötigt man Computer, doch selbst dann übersteigt der Zeitaufwand zur Lösung schnell die Grenzen des Machbaren, wie wir später sehen werden. Die besten Algorithmen, die für solche Probleme bekannt sind, benötigen Exponentialzeit und sind daher asymptotisch ineffizient, d. h., sie haben eine Laufzeit, die mit wachsender Eingabegröße „explodiert" und sehr rasch astronomische Größen erreicht.

Dennoch möchte man solche wichtigen Probleme lösen. Dieses Buch befasst sich mit schweren Problemen auf Graphen, für die es vermutlich keine effizienten Algorithmen gibt, und stellt verschiedene Methoden vor, wie man mit der algorithmischen Härte solcher Probleme umgehen kann.

Dieses Buch ist in drei Teile gegliedert. Im ersten Teil werden die Grundlagen behandelt. Insbesondere werden die elementaren Begriffe und Bezeichnungen für die Aufwandsabschätzung von Algorithmen eingeführt und die benötigten Fundamente der Graphentheorie, Logik und Komplexitätstheorie gelegt. Neben der klassischen Komplexitätstheorie gehen wir insbesondere auch auf die parametrisierte Komplexität von Problemen ein.

Im zweiten Teil werden (uneingeschränkte) schwere Graphenprobleme algorithmisch untersucht. Zunächst werden Fest-Parameter-Algorithmen behandelt, die im Sinne der parametrisierten Komplexität effizient sind. Solche Algorithmen fangen die einem schweren Problem innewohnende Berechnungshärte, die mit wachsender Eingabegröße zu einem explosiven Anwachsen des Zeitaufwands bei der Lösung führt, in einem oder mehreren geeigneten Parametern ein und machen sie dort unschädlich: Für eine feste Wahl des Parameters wird das exponentielle Verhalten zu einem konstanten Faktor in der Laufzeitanalyse degradiert, und wenn der feste Parameter in „typischen" Eingaben nicht zu groß ist, ist ein solcher Fest-Parameter-Algorithmus sogar effizient.

Auch wenn diese Methoden nicht anwendbar sind, können die vorhandenen exakten Exponentialzeit-Algorithmen für solche schweren Probleme oft verbessert werden. Deshalb behandelt der zweite Teil auch Exponentialzeit-Algorithmen, die zwar nicht als effizient betrachtet werden können, aber dennoch deutlich besser als der „naive" Algorithmus für die entsprechenden schweren Graphenprobleme sind. Unter dem naiven Algorithmus versteht man dabei einfach das sture, systematische Ausprobieren aller Lösungsmöglichkeiten. In der Praxis können auch Exponentialzeit-Algorithmen für moderate Eingabegrößen nützlich sein – nämlich dann, wenn es gelingt, die exponentielle Zeitschranke entsprechend zu verbessern. Findet man etwa einen Algorithmus, der nicht wie der naive Algorithmus für dieses Problem in der Zeit $2^n$, sondern dank einer raffinierten Idee in der Zeit $\sqrt{2}^n \approx 1.414^n$

läuft, so kann man mit demselben Zeitaufwand doppelt so große Eingaben bearbeiten. Ob man ein Graphenproblem statt für Graphen mit 30 sogar für solche mit 60 Knoten in einem gerade noch so vertretbaren Aufwand bewältigen kann, spielt in der Praxis oft eine entscheidende Rolle.

Im dritten Teil schließlich werden Algorithmen vorgestellt, die im Allgemeinen nur schwer lösbare Graphenprobleme auf geeignet eingeschränkten Graphen effizient lösen können. Einerseits kann man effiziente Algorithmen entwerfen, die sich eine geeignete Baumstruktur der Graphen zunutze machen. Andererseits erlauben Fest-Parameter-Algorithmen eine effiziente Lösung, wenn gewisse Graphparameter klein sind. Dies betrifft spezielle Graphklassen wie zum Beispiel Bäume, Co-Graphen sowie baumweite- und cliquenweitebeschränkte Graphen.

Durch die leicht verständliche Darstellung, viele erklärende Abbildungen, Beispiele und Übungsaufgaben sowie die durchdachte Auswahl von Resultaten und Techniken ist dieses Buch besonders gut für den Einsatz in der Lehre geeignet, vor allem im Masterstudium Informatik und in den höheren Semestern des Bachelorstudiums Informatik. Gleichzeitig führt es den Leser unmittelbar an die Fronten der aktuellen Forschung in diesem neuen Teilgebiet der Algorithmik heran.