SECOND EDITION

INTRODUCTION TO GRAPH THEORY

DOUGLAS B. WEST

ALWAYS LEARNING

PEARSON

Introduction to
Graph Theory

Second Edition

Douglas B. West
University of Illinois — Urbana

PEARSON

ISBN 978-93-325-4965-4

First Impression, 2015

[redacted]

Published by Pearson India Education Services Pvt.Ltd, CIN: U72200TN2005PTC057128.
Formerly known as TutorVista Global Pvt Ltd, licensees of Pearson Education in South Asia

Head Office: 7th Floor, knowledge Boulevard, A-8(A) Sector-62, Noida (U.P) 201309, India

Registered Office: Module G4, Ground Floor, Elnet Software City, TS -140, Block 2 & 9
Rajiv Gandhi Salai, Taramani, Chennai, Tamil Nadu 600113.
Fax: 080-30461003, Phone: 080-30461060, www.pearson.co.in email id: CS.India@pearson.com

Printed in India by Shree Maitrey Printech Pvt. Ltd., Noida.

For my dear wife Ching

and for all lovers of graph theory

Contents

Preface xi

Chapter 1 Fundamental Concepts 1

1.1 What Is a Graph? 1
The Definition, 1
Graphs as Models, 3
Matrices and Isomorphism, 6
Decomposition and Special Graphs, 11
Exercises, 14

1.2 Paths, Cycles, and Trails 19
Connection in Graphs, 20
Bipartite Graphs, 24
Eulerian Circuits, 26
Exercises, 31

1.3 Vertex Degrees and Counting 34
Counting and Bijections, 35
Extremal Problems, 38
Graphic Sequences, 44
Exercises, 47

1.4 Directed Graphs 53
Definitions and Examples, 53
Vertex Degrees, 58
Eulerian Digraphs, 60
Orientations and Tournaments, 61
Exercises, 63

Chapter 2 Trees and Distance **67**

2.1 Basic Properties **67**
Properties of Trees, 68
Distance in Trees and Graphs, 70
Disjoint Spanning Trees (optional), 73
Exercises, 75

2.2 Spanning Trees and Enumeration **81**
Enumeration of Trees, 81
Spanning Trees in Graphs, 83
Decomposition and Graceful Labelings, 87
Branchings and Eulerian Digraphs (optional), 89
Exercises, 92

2.3 Optimization and Trees **95**
Minimum Spanning Tree, 95
Shortest Paths, 97
Trees in Computer Science (optional), 100
Exercises, 103

Chapter 3 Matchings and Factors **107**

3.1 Matchings and Covers **107**
Maximum Matchings, 108
Hall's Matching Condition, 110
Min-Max Theorems, 112
Independent Sets and Covers, 113
Dominating Sets (optional), 116
Exercises, 118

3.2 Algorithms and Applications **123**
Maximum Bipartite Matching, 123
Weighted Bipartite Matching, 125
Stable Matchings (optional), 130
Faster Bipartite Matching (optional), 132
Exercises, 134

3.3 Matchings in General Graphs **136**
Tutte's 1-factor Theorem, 136
f-factors of Graphs (optional), 140
Edmonds' Blossom Algorithm (optional), 142
Exercises, 145

Chapter 4 Connectivity and Paths 149

4.1 Cuts and Connectivity **149**

Connectivity, 149
Edge-connectivity, 152
Blocks, 155
Exercises, 158

4.2 *k*-connected Graphs **161**

2-connected Graphs, 161
Connectivity of Digraphs, 164
k-connected and *k*-edge-connected Graphs, 166
Applications of Menger's Theorem, 170
Exercises, 172

4.3 Network Flow Problems **176**

Maximum Network Flow, 176
Integral Flows, 181
Supplies and Demands (optional), 184
Exercises, 188

Chapter 5 Coloring of Graphs 191

5.1 Vertex Colorings and Upper Bounds **191**

Definitions and Examples, 191
Upper Bounds, 194
Brooks' Theorem, 197
Exercises, 199

5.2 Structure of *k*-chromatic Graphs **204**

Graphs with Large Chromatic Number, 205
Extremal Problems and Turán's Theorem 207
Color-Critical Graphs, 210
Forced Subdivisions, 212
Exercises, 214

5.3 Enumerative Aspects **219**

Counting Proper Colorings, 219
Chordal Graphs, 224
A Hint of Perfect Graphs, 226
Counting Acyclic Orientations (optional), 228
Exercises, 229

Chapter 6 Planar Graphs 233

6.1 Embeddings and Euler's Formula **233**

Drawings in the Plane, 233
Dual Graphs, 236
Euler's Formula, 241 255
Exercises, 243

6.2 Characterization of Planar Graphs **246**

Preparation for Kuratowski's Theorem, 247
Convex Embeddings, 248
Planarity Testing (optional), 252
Exercises, 255

6.3 Parameters of Planarity **257**

Coloring of Planar Graphs, 257
Crossing Number, 261
Surfaces of Higher Genus (optional), 266
Exercises, 269

Chapter 7 Edges and Cycles 273

7.1 Line Graphs and Edge-coloring **273**

Edge-colorings, 274
Characterization of Line Graphs (optional), 279
Exercises, 282

7.2 Hamiltonian Cycles **286**

Necessary Conditions, 287
Sufficient Conditions, 288
Cycles in Directed Graphs (optional), 293
Exercises, 294

7.3 Planarity, Coloring, and Cycles **299**

Tait's Theorem, 300
Grinberg's Theorem, 302
Snarks (optional), 304
Flows and Cycle Covers (optional), 307
Exercises, 314

Chapter 8 Additional Topics (optional) 319

8.1 Perfect Graphs 319

The Perfect Graph Theorem, 320
Chordal Graphs Revisited, 323
Other Classes of Perfect Graphs, 328
Imperfect Graphs, 334
The Strong Perfect Graph Conjecture, 340
Exercises, 344

8.2 Matroids 349

Hereditary Systems and Examples, 349
Properties of Matroids, 354
The Span Function, 358
The Dual of a Matroid, 360
Matroid Minors and Planar Graphs, 363
Matroid Intersection, 366
Matroid Union, 369
Exercises, 372

8.3 Ramsey Theory 378

The Pigeonhole Principle Revisited, 378
Ramsey's Theorem, 380
Ramsey Numbers, 383
Graph Ramsey Theory, 386
Sperner's Lemma and Bandwidth, 388
Exercises, 392

8.4 More Extremal Problems 396

Encodings of Graphs, 397
Branchings and Gossip, 404
List Coloring and Choosability, 408
Partitions Using Paths and Cycles, 413
Circumference, 416
Exercises, 422

8.5 Random Graphs 425

Existence and Expectation, 426
Properties of Almost All Graphs, 430
Threshold Functions, 432
Evolution and Graph Parameters, 436
Connectivity, Cliques, and Coloring, 439
Martingales, 442
Exercises, 448

8.6 Eigenvalues of Graphs **452**
The Characteristic Polynomial, 453
Linear Algebra of Real Symmetric Matrices, 456
Eigenvalues and Graph Parameters, 458
Eigenvalues of Regular Graphs, 460
Eigenvalues and Expanders, 463
Strongly Regular Graphs, 464
Exercises, 467

Appendix A Mathematical Background **471**

Sets, 471
Quantifiers and Proofs, 475
Induction and Recurrence, 479
Functions, 483
Counting and Binomial Coefficients, 485
Relations, 489
The Pigeonhole Principle, 491

Appendix B Optimization and Complexity **493**

Intractability, 493
Heuristics and Bounds, 496
NP-Completeness Proofs, 499
Exercises, 505

Appendix C Hints for Selected Exercises **507**

General Discussion, 507
Supplemental Specific Hints, 508

Appendix D Glossary of Terms **515**

Appendix E Supplemental Reading **533**

Appendix F References **567**

Author Index **569**

Subject Index **575**

Preface

Graph theory is a delightful playground for the exploration of proof techniques in discrete mathematics, and its results have applications in many areas of the computing, social, and natural sciences. The design of this book permits usage in a one-semester introduction at the undergraduate or beginning graduate level, or in a patient two-semester introduction. No previous knowledge of graph theory is assumed. Many algorithms and applications are included, but the focus is on understanding the structure of graphs and the techniques used to analyze problems in graph theory.

Many textbooks have been written about graph theory. Due to its emphasis on both proofs and applications, the initial model for this book was the elegant text by J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications* (Macmillan/North-Holland [1976]). Graph theory is still young, and no consensus has emerged on how the introductory material should be presented. Selection and order of topics, choice of proofs, objectives, and underlying themes are matters of lively debate. Revising this book dozens of times has taught me the difficulty of these decisions. This book is my contribution to the debate.

The Second Edition

The revision for the second edition emphasizes making the text easier for the students to learn from and easier for the instructor to teach from. There have not been great changes in the overall content of the book, but the presentation has been modified to make the material more accessible, especially in the early parts of the book. Some of the changes are discussed in more detail later in this preface; here I provide a brief summary.

- Optional material within non-optional sections is now designated by $(*)$; such material is not used later and can be skipped. Most of it is *intended* to be skipped in a one-semester course. When a subsection is marked "optional", the entire subsection is optional, and hence no individual items are starred.
- For less-experienced students, Appendix A has been added as a reference summary of helpful material on sets, logical statements, induction, counting arguments, binomial coefficients, relations, and the pigeonhole principle.

- Many proofs have been reworded in more patient language with additional details, and more examples have been added.
- More than 350 exercises have been added, mostly easier exercises in Chapters 1–7. There are now more than 1200 exercises.
- More than 100 illustrations have been added; there are now more than 400. In illustrations showing several types of edges, the switch to bold and solid edges instead of solid and dashed edges has increased clarity.
- Easier problems are now grouped at the beginning of each exercise section, usable as warm-ups. Statements of some exercises have been clarified.
- In addition to hints accompanying the exercise statements, there is now an appendix of supplemental hints.
- For easier access, terms being defined are in bold type, and the vast majority of them appear in Definition items.
- For easier access, the glossary of notation has been placed on the inside covers.
- Material involving Eulerian circuits, digraphs, and Turán's Theorem has been relocated to facilitate more efficient learning.
- Chapters 6 and 7 have been switched to introduce the idea of planarity earlier, and the section on complexity has become an appendix.
- The glossary has been improved to eliminate errors and to emphasize items more directly related to the text.

Features

Various features of this book facilitate students' efforts to understand the material. There is discussion of proof techniques, more than 1200 exercises of varying difficulty, more than 400 illustrations, and many examples. Proofs are presented in full in the text.

Many undergraduates begin a course in graph theory with little exposure to proof techniques. Appendix A provides background reading that will help them get started. Students who have difficulty understanding or writing proofs in the early material should be encouraged to read this appendix in conjunction with Chapter 1. Some discussion of proof techniques still appears in the early sections of the text (especially concerning induction), but an expanded treatment of the basic background (especially concerning sets, functions, relations, and elementary counting) is now in Appendix A.

Most of the exercises require proofs. Many undergraduates have had little practice at presenting explanations, and this hinders their appreciation of graph theory and other mathematics. The intellectual discipline of justifying an argument is valuable independently of mathematics; I hope that students will appreciate this. In writing solutions to exercises, students should be careful in their use of language ("say what you mean"), and they should be intellectually honest ("mean what you say").

Although many terms in graph theory suggest their definitions, the quantity of terminology remains an obstacle to fluency. Mathematicians like to gather definitions at the start, but most students succeed better if they use a concept before receiving the next. This, plus experience and requests from reviewers, has led me to postpone many definitions until they are needed. For example, the definition of cartesian product appears in Section 5.1 with coloring problems. Line graphs are defined in Section 4.2 with Menger's Theorem and in Section 7.1 with edge-coloring. The definitions of induced subgraph and join have now been postponed to Section 1.2 and Section 3.1, respectively.

I have changed the treatment of digraphs substantially by postponing their introduction to Section 1.4. Introducing digraphs at the same time as graphs tends to confuse or overwhelm students. Waiting to the end of Chapter 1 allows them to become comfortable with basic concepts in the context of a single model. The discussion of digraphs then reinforces some of those concepts while clarifying the distinctions. The two models are still discussed together in the material on connectivity.

This book contains more material than most introductory texts in graph theory. Collecting the advanced material as a final optional chapter of "additional topics" permits usage at different levels. The undergraduate introduction consists of the first seven chapters (omitting most optional material), leaving Chapter 8 as topical reading for interested students. A graduate course can treat most of Chapters 1 and 2 as recommended reading, moving rapidly to Chapter 3 in class and reaching some topics in Chapter 8. Chapter 8 can also be used as the basis for a second course in graph theory, along with material that was optional in earlier chapters.

Many results in graph theory have several proofs; illustrating this can increase students' flexibility in trying multiple approaches to a problem. I include some alternative proofs as remarks and others as exercises.

Many exercises have hints, some given with the exercise statement and others in Appendix C. Exercises marked "(−)" or "(+)" are easier or more difficult, respectively, than unmarked problems. Those marked "(+)" should *not* be assigned as homework in a typical undergraduate course. Exercises marked "(!)" are especially valuable, instructive, or entertaining. Those marked "(∗)" use material labeled optional in the text.

Each exercise section begins with a set of "(−)" exercises, ordered according to the material in the section and ending with a line of bullets. These exercises either check understanding of concepts or are immediate applications of results in the section. I recommend some of these to my class as "warmup" exercises to check their understanding before working the main homework problems, most of which are marked "(!)". Most problems marked "(−)" are good exam questions. When using other exercises on exams, it may be a good idea to provide hints from Appendix C.

Exercises that relate several concepts appear when the last is introduced. Many pointers to exercises appear in the text where relevant concepts are discussed. An exercise in the current section is cited by giving only its item number among the exercises of that section. Other cross-references are by Chapter.Section.Item.

Organization and Modifications

In the first edition, I sought a development that was intellectually coherent and displayed a gradual (not monotonic) increase in difficulty of proofs and in algorithmic complexity.

Carrying this further in the second edition, Eulerian circuits and Hamiltonian cycles are now even farther apart. The simple characterization of Eulerian circuits is now in Section 1.2 with material closely related to it. The remainder of the former Section 2.4 has been dispersed to relevant locations in other sections, with Fleury's Algorithm dropped.

Chapter 1 has been substantially rewritten. I continue to avoid the term "multigraph"; it causes more trouble than it resolves, because many students assume that a multigraph *must* have multiple edges. It is less distracting to append the word "simple" where needed and keep "graph" as the general object, with occasional statements that in particular contexts it makes sense to consider only simple graphs.

The treatment of definitions in Chapter 1 has been made more friendly and precise, particularly those involving paths, trails, and walks. The informal groupings of basic definitions in Section 1.1 have been replaced by Definition items that help students find definitions more easily.

In addition to the material on isomorphism, Section 1.1 now has a more precise treatment of the Petersen graph and an explicit introduction of the notions of decomposition and girth. This provides language that facilitates later discussion in various places, and it permits interesting explicit questions other than isomorphism.

Sections 1.2–1.4 have become more coherent. The treatment of Eulerian circuits motivates and completes Section 1.2. Some material has been removed from Section 1.3 to narrow its focus to degrees and counting, and this section has acquired the material on vertex degrees that had been in Section 1.4. Section 1.4 now provides the introduction to digraphs and can be treated lightly.

Trees and distance appear together in Chapter 2 due to the many relations between these topics. Many exercises combine these notions, and algorithms to compute distances produce or use trees.

Most graph theorists agree that the König-Egerváry Theorem deserves an independent proof without network flow. Also, students have trouble distinguishing "k-connected" from "connectivity k", which have the same relationship as "k-colorable" and "chromatic number k". I therefore treat matching first and later use matching to prove Menger's Theorem. Both matching and connectivity are used in the coloring material.

In response to requests from a number of users, I have added a short optional subsection on dominating sets at the end of Section 3.1. The material on weighted bipartite matching has been clarified by emphasis on vertex cover instead of augmenting path and by better use of examples.

Turán's Theorem uses only elementary ideas about vertex degrees and induction and hence appeared in Chapter 1 in the first edition. This caused some difficulties, because it was the most abstract item up to that point and students felt somewhat overwhelmed by it. Thus I have kept the simple triangle-free case (Mantel's Theorem) in Section 1.3 and have moved the full theorem to Section 5.2 under the viewpoint of extremal problems related to coloring.

The chapter on planarity now comes before that on "Edges and Cycles". When an instructor is short of time, planarity is more important to reach than the material on edge-coloring and Hamiltonian cycles. The questions involved in planarity appeal intuitively to students due to their visual aspects, and many students have encountered these questions before. Also, the ideas involved in discussing planar graphs seem more intellectually broadening in relation to the earlier material of the course than the ideas used to prove the basic results on edge-coloring and Hamiltonian cycles.

Finally, discussing planarity first makes the material of Chapter 7 more coherent. The new arrangement permits a more thorough discussion of the relationships among planarity, edge-coloring, and Hamiltonian cycles, leading naturally beyond the Four Color Theorem to the optional new material on nowhere-zero flows as a dual concept to coloring.

When students discover that the coloring and Hamiltonian cycle problems lack good algorithms, many become curious about NP-completeness. Appendix B satisfies this curiosity. Presentation of NP-completeness via formal languages can be technically abstract, so some students appreciate a presentation in the context of graph problems. NP-completeness proofs also illustrate the variety and usefulness of "graph transformation" arguments.

The text explores relationships among fundamental results. Petersen's Theorem on 2-factors (Chapter 3) uses Eulerian circuits and bipartite matching. The equivalence between Menger's Theorem and the Max Flow-Min Cut Theorem is explored more fully than in the first edition, and the "Baseball Elimination" application is now treated in more depth. The $k-1$-connectedness of k-color-critical graphs (Chapter 5) uses bipartite matching. Section 5.3 offers a brief introduction to perfect graphs, emphasizing chordal graphs. Additional features of this text in comparison to some others include the algorithmic proof of Vizing's Theorem and the proof of Kuratowski's Theorem by Thomassen's methods.

There are various other additions and improvements in the first seven chapters. There is now a brief discussion of Heawood's Formula and the Robertson–Seymour Theorem at the end of Chapter 6. In Section 7.1, a proof of Shannon's bound on the edge-chromatic number has been added. In Section 5.3, the characterization of chordal graphs is somewhat simpler than before by proving a stronger result about simplicial vertices. In Section 6.3, the proof of the reducibility of the Birkhoff diamond has been eliminated, but a brief discussion of discharging has been added. The material discussing issues in the proof of the theorem is optional, and the aim is to give the flavor of the approach without getting into detailed arguments. From this viewpoint the reducibility proof seemed out of focus.

Chapter 8 contains highlights of advanced material and is not intended for an undergraduate course. It assumes more sophistication than earlier chapters and is written more tersely. Its sections are independent; each selects appealing results from a large topic that merits a chapter of its own. Some of these sections become more difficult near the end; an instructor may prefer to sample early material in several sections rather than present one completely.

There may be occasional relationships between items in Chapter 8 and items marked optional in the first seven chapters, but generally cross-references indicate the connections. The material of Chapter 8 has not changed substantially since the first edition, although many corrections have been made and the presentation has been clarified in many places.

I will treat advanced graph theory more thoroughly in *The Art of Combinatorics*. Volume I is devoted to extremal graph theory and Volume II to structure of graphs. Volume III has chapters on matroids and integer programming (including network flow). Volume IV emphasizes methods in combinatorics and discusses various aspects of graphs, especially random graphs.

Design of Courses

I intend the 22 sections in Chapters 1–7 for a pace of slightly under two lectures per section when most optional material (starred items and optional subsections) is skipped. When I teach the course I spend eight lectures on Chapter 1, six lectures each on Chapters 4 and 5, and five lectures on each of Chapters 2, 3, 6, and 7. This presents the fundamental material in about 40 lectures. Some instructors may want to spend more time on Chapter 1 and omit more material from later chapters.

In chapters after the first, the most fundamental material is concentrated in the first section. Emphasizing these sections (while skipping the optional items) still illustrates the scope of graph theory in a slower-paced one-semester course. From the second sections of Chapters 2, 4, 5, 6, and 7, it would be beneficial to include Cayley's Formula, Menger's Theorem, Mycielski's construction, Kuratowski's Theorem, and Dirac's Theorem (spanning cycles), respectively.

Some optional material is particularly appealing to present in class. For example, I always present the optional subsections on Disjoint Spanning Trees (in Section 2.1) and Stable Matchings (in Section 3.2), and I usually present the optional subsection on f-factors (in Section 3.3). Subsections are marked optional when no later material in the first seven chapters requires them and they are not part of the central development of elementary graph theory, but these are nice applications that engage students' interest. In one sense, the "optional" marking indicates to students that the final examination is unlikely to have questions on these topics.

Graduate courses skimming the first two chapters might include from them such topics as graphic sequences, kernels of digraphs, Cayley's Formula, the Matrix Tree Theorem, and Kruskal's algorithm.

Courses that introduce graph theory in one term under the quarter system must aim for highlights; I suggest the following rough syllabus: 1.1: adjacency matrix, isomorphism, Petersen graph. 1.2: all. 1.3: degree-sum formula, large bipartite subgraphs. 1.4: up to strong components, plus tournaments. 2.1: up to centers of trees. 2.2: up to statement of Matrix Tree Theorem. 2.3: Kruskal's algorithm. 3.1: almost all. 3.2: none. 3.3: statement of Tutte's Theorem, proof of Petersen's results. 4.1: up to definition of blocks, omitting Harary graphs. 4.2: up to open ear decomposition, plus statement of Menger's Theorem(s). 4.3: duality between flows and cuts, statement of Max-flow = Min-cut. 5.1: up to Szekeres-Wilf theorem. 5.2: Mycielski's construction, possibly Turán's Theorem. 5.3: up to chromatic recurrence, plus perfection of chordal graphs. 6.1: non-planarity of K_5 and $K_{3,3}$, examples of dual graphs, Euler's formula with applications. 6.2: statement and examples of Kuratowski's Theorem and Tutte's Theorem. 6.3: 5-color Theorem, plus the idea of crossing number. 7.1: up to Vizing's Theorem. 7.2: up to Ore's condition, plus the Chvátal-Erdös condition. 7.3: Tait's Theorem, Grinberg's Theorem.

Further Pedagogical Aspects

In the revision I have emphasized some themes that arise naturally from the material; underscoring these in lecture helps provide continuity.

More emphasis has been given to the theme of TONCAS—"The obvious necessary condition is also sufficient." Explicit mention has been added that many of the fundamental results can be viewed in this way. This both provides a theme for the course and clarifies the distinction between the easy direction and the hard direction in an equivalence.

Another theme that underlies much of Chapters 3–5 and Section 7.1 is that of dual maximization and minimization problems. In a graph theory course one does not want to delve deeply into the nature of duality in linear optimization. It suffices to say that two optimization problems form a dual pair when every feasible solution to the maximization problem has value at most the value of every feasible solution to the minimization problem. When feasible solutions with the same value are given for the two problems, this duality implies that both solutions are optimal. A discussion of the linear programming context appears in Section 8.1.

Other themes can be identified among the proof techniques. One is the use of extremality to give short proofs and avoid the use of induction. Another is the paradigm for proving conditional statements by induction, as described explicitly in Remark 1.3.25.

The development leading to Kuratowski's Theorem is somewhat long. Nevertheless, it is preferable to present the proof in a single lecture. The preliminary lemmas reducing the problem to the 3-connected case can be treated lightly to save time. Note that the induction paradigm leads naturally to the two lemmas proved for the 3-connected case. Note also that the proof uses the notion of S-lobe defined in Section 5.2.

The first lecture in Chapter 6 should not belabor technical definitions of drawings and regions. These are better left as intuitive notions unless students ask for details; the precise statements appear in the text.

The motivating applications of digraphs in Section 1.4 have been marked optional because they are not needed in the rest of the text, but they help clarify that the appropriate model (graph or digraph) depends on the application.

Due to its reduced emphasis on numerical computation and increased emphasize on techniques of proof and clarity of explanations, graph theory is an excellent subject in which to encourage students to improve their habits of communication, both written and oral. In addition to assigning written homework that requires carefully presented arguments, I have found it productive to organize optional "collaborative study" sessions in which students can work together on problems while I circulate, listen, and answer questions. It should not be forgotten that one of the best ways to discover whether one understands a proof is to try to explain it to someone else. The students who participate find these sessions very beneficial.

Acknowledgments

This text benefited from classroom testing of gradually improving prepublication versions at many universities. Instructors who used the text on this experimental basis were, roughly in chronological order, Ed Scheinerman (Johns Hopkins), Kathryn Fraughnaugh (Colorado-Denver), Paul Weichsel / Paul Schupp / Xiaoyun Lu (Illinois), Dean Hoffman / Pete Johnson / Chris Rodger (Auburn), Dan Ullman (George Washington), Zevi Miller / Dan Pritikin (Miami-Ohio), David Matula (Southern Methodist), Pavol Hell (Simon Fraser), Grzegorz Kubicki (Louisville), Jeff Smith (Purdue), Ann Trenk (Wellesley), Ken Bogart (Dartmouth), Kirk Tolman (Brigham Young), Roger Eggleton (Illinois State), Herb Kasube (Bradley), and Jeff Dinitz (Vermont). Many of these (or their students) provided suggestions or complaints that led to improvements.

I thank George Lobell at Prentice Hall for his continuing commitment to this project and for finding diligent reviewers. Reviewers Paul Edelman, Renu Laskar, Gary MacGillivray, Joseph Neggers, Joseph Malkevitch, James Oxley, Sam Stueckle, and Barry Tesman made helpful comments. Reviewers for early versions of sections of Chapter 8 included Mike Albertson, Sanjoy Barvah, Dan Kleitman, James Oxley, Chris Rodger, and Alan Tucker. Reviewers for the second edition included Nate Dean, Dalibor Froncek, Renu Laskar, Michael Molloy, David Sumner, and Daniel Ullman.

Additional comments and corrections between the first and second editions came from many readers. These items range from typographical errors to additional exercises to simplifications in proofs. The accumulated effect of these contributions is substantial. Most of these suggestions don't lend themselves to attribution within the text, so I express my gratitude here for their willingness to contribute their observations, opinions, and expertise. These people include Troy Barcume, Stephan Brandt, Gerard Chang, Scott Clark, Dave Gunderson, Dean Hoffman, John D'Angelo, Charles Delzell, Thomas Emden-Weinert, Shimon Even, Fred Galvin, Alfio Giarlotta, Don Greenwell, Jing Huang, Garth

Isaak, Steve Kilner, Alexandr Kostochka, André Kündgen, Peter Kwok, Jean-Marc Lanlignel, Francois Margot, Alan Mehlenbacher, Joel Miller, Zevi Miller, Wendy Myrvold, Charles Parry, Robert Pratt, Dan Pritikin, Radhika Ramamurthi, Craig Rasmussen, Bruce Reznick, Jian Shen, Tom Shermer, Warren Shreve, Alexander Strehl, Tibor Szabó, Vitaly Voloshin, and C.Q. Zhang.

Several students found numerous typographical errors in the pre-publication version of the second edition (thereby earning extra credit!): Jaspreet Bagga, Brandon Bowersox, Mark Chabura, John Chuang, Greg Harfst, Shalene Melo, Charlie Pikscher, and Josh Reed.

The cover drawing for the first edition was produced by Ed Scheinerman using BRL-CAD, a product of the U.S. Army Ballistic Research Laboratory. For the second edition, the drawing was produced by Maria Muyot using CorelDraw.

Chris Hartman contributed vital assistance in preparing the bibliography for the first edition; additional references have now been added. Ted Harding helped resolve typesetting difficulties in other parts of the first edition.

Students who helped gather data for the index of the first edition included Maria Axenovich, Nicole Henley, André Kündgen, Peter Kwok, Kevin Leuthold, John Jozwiak, Radhika Ramamurthi, and Karl Schmidt. Raw data for the index of the second edition was gathered using scripts I wrote in `perl`; Maria Muyot and Radhika Ramamurthi assisted with processing of the index and the bibliography.

I prepared the second edition in TeX, the typesetting system for which the scientific world owes Donald E. Knuth eternal gratitude. The figures were generated using `gpic`, a product of the Free Software Foundation.

Feedback

I welcome corrections and suggestions, including comments on topics, attributions of results, updates, suggestions for exercises, typographical errors, omissions from the glossary or index, etc. Please send these to me at

west@math.uiuc.edu

In particular, I apologize in advance for missing references; please inform me of the proper citations! Also, no changes other than corrections of errors will be made between printings of this edition.

I maintain a web site containing a syllabus, errata, updates, and other material. Please visit!

http://www.math.uiuc.edu/~west/igt

I have corrected all typographical and mathematical errors known to me before the time of printing. Nevertheless, the robustness of the set of errors and the substantial rewriting and additions make me confident that some error remains. Please find it and tell me so I can correct it!

Douglas B. West
Urbana, Illinois

Chapter 1

Fundamental Concepts

1.1. What Is a Graph?

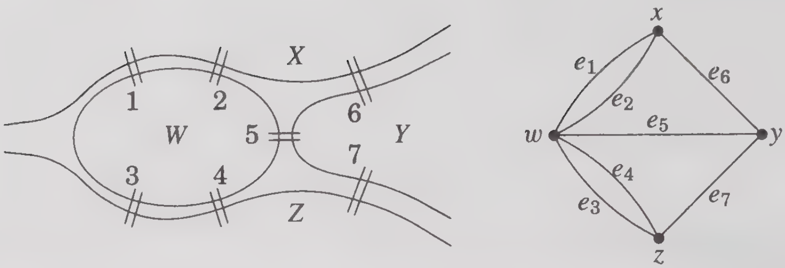
How can we lay cable at minimum cost to make every telephone reachable from every other? What is the fastest route from the national capital to each state capital? How can n jobs be filled by n people with maximum total utility? What is the maximum flow per unit time from source to sink in a network of pipes? How many layers does a computer chip need so that wires in the same layer don't cross? How can the season of a sports league be scheduled into the minimum number of weeks? In what order should a traveling salesman visit cities to minimize travel time? Can we color the regions of every map using four colors so that neighboring regions receive different colors?

These and many other practical problems involve graph theory. In this book, we develop the theory of graphs and apply it to such problems. Our starting point assumes the mathematical background in Appendix A, where basic objects and language of mathematics are discussed.

THE DEFINITION

The problem that is often said to have been the birth of graph theory will suggest our basic definition of a graph.

1.1.1. Example. *The Königsberg Bridge Problem.* The city of Königsberg was located on the Pregel river in Prussia. The city occupied two islands plus areas on both banks. These regions were linked by seven bridges as shown on the left below. The citizens wondered whether they could leave home, cross every bridge exactly once, and return home. The problem reduces to traversing the figure on the right, with heavy dots representing land masses and curves representing bridges.



The model on the right makes it easy to argue that the desired traversal does not exist. Each time we enter and leave a land mass, we use two bridges ending at it. We can also pair the first bridge with the last bridge on the land mass where we begin and end. Thus existence of the desired traversal requires that each land mass be involved in an even number of bridges. This necessary condition did not hold in Königsberg. ■

The Königsberg Bridge Problem becomes more interesting when we show in Section 1.2 which configurations have traversals. Meanwhile, the problem suggests a general model for discussing such questions.

1.1.2. Definition. A **graph** G is a triple consisting of a **vertex set** $V(G)$, an **edge set** $E(G)$, and a relation that associates with each edge two vertices (not necessarily distinct) called its **endpoints**.

We **draw** a graph on paper by placing each vertex at a point and representing each edge by a curve joining the locations of its endpoints.

1.1.3. Example. In the graph in Example 1.1.1, the vertex set is $\{x, y, z, w\}$, the edge set is $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$, and the assignment of endpoints to edges can be read from the picture.

Note that edges e_1 and e_2 have the same endpoints, as do e_3 and e_4. Also, if we had a bridge over an inlet, then its ends would be in the same land mass, and we would draw it as a curve with both ends at the same point. We have appropriate terms for these types of edges in graphs. ■

1.1.4. Definition. A **loop** is an edge whose endpoints are equal. **Multiple edges** are edges having the same pair of endpoints.

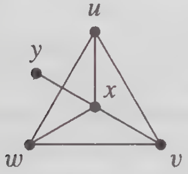
A **simple graph** is a graph having no loops or multiple edges. We specify a simple graph by its vertex set and edge set, treating the edge set as a set of unordered pairs of vertices and writing $e = uv$ (or $e = vu$) for an edge e with endpoints u and v.

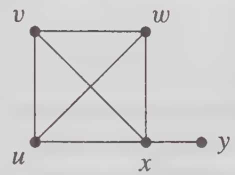
When u and v are the endpoints of an edge, they are **adjacent** and are **neighbors**. We write $u \leftrightarrow v$ for "u is adjacent to v".

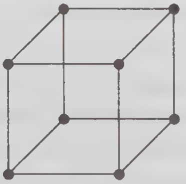
In many important applications, loops and multiple edges do not arise, and we restrict our attention to simple graphs. In this case an edge is determined by its endpoints, so we can *name* the edge by its endpoints, as stated in Definition 1.1.4. Thus in a *simple* graph we view an edge as an unordered pair of vertices and can ignore the formality of the relation associating endpoints to edges. This book emphasizes simple graphs.

1.1.5. Example. On the left below are two drawings of a simple graph. The vertex set is $\{u, v, w, x, y\}$, and the edge set is $\{uv, uw, ux, vx, vw, xw, xy\}$.

The terms "vertex" and "edge" arise from solid geometry. A cube has vertices and edges, and these form the vertex set and edge set of a graph. It is drawn on the right below, omitting the names of vertices and edges. ■







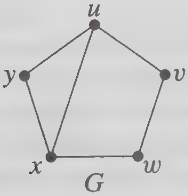
A graph is **finite** if its vertex set and edge set are finite. We adopt the convention that **every graph mentioned in this book is finite**, unless explicitly constructed otherwise.

1.1.6.* Remark. The **null graph** is the graph whose vertex set and edge set are empty. Extending general theorems to the null graph introduces unnecessary distractions, so we ignore it. All statements and exercises should be considered only for graphs with a nonempty set of vertices. ■

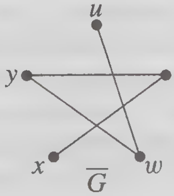
GRAPHS AS MODELS

Graphs arise in many settings. The applications suggest useful concepts and terminology about the structure of graphs.

1.1.7. Example. *Acquaintance relations and subgraphs.* Does every set of six people contain three mutual acquaintances or three mutual strangers? Since "acquaintance" is symmetric, we model it using a simple graph with a vertex for each person and an edge for each acquainted pair. The "nonacquaintance" relation on the same set yields another graph with the "complementary" set of edges. We introduce terms for these concepts. ■



G



$\overline{G}$

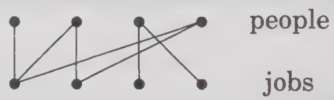
1.1.8. Definition. The **complement** $\overline{G}$ of a simple graph G is the simple graph with vertex set $V(G)$ defined by $uv \in E(\overline{G})$ if and only if $uv \notin E(G)$. A **clique** in a graph is a set of pairwise adjacent vertices. An **independent set** (or **stable set**) in a graph is a set of pairwise nonadjacent vertices.

In the graph G of Example 1.1.7, $\{u, x, y\}$ is a clique of size 3 and $\{u, w\}$ is an independent set of size 2, and these are the largest such sets. These values reverse in the complement $\overline{G}$, since cliques become independent sets (and vice versa) under complementation. The question in Example 1.1.7 asks whether it is true that every 6-vertex graph has a clique of size 3 or an independent set of size 3 (Exercise 29). Deleting edge ux from G yields a 5-vertex graph having no clique or independent set of size 3.

1.1.9. Example. *Job assignments and bipartite graphs.* We have m jobs and n people, but not all people are qualified for all jobs. Can we fill the jobs with qualified people? We model this using a simple graph H with vertices for the jobs and people; job j is adjacent to person p if p can do j.

Each job is to be filled by exactly one person, and each person can hold at most one of the jobs. Thus we seek m pairwise disjoint edges in H (viewing edges as pairs of vertices). Chapter 3 shows how to test for this; it can't be done in the graph below.

The use of graphs to model relations between two disjoint sets has many important applications. These are the graphs whose vertex sets can be partitioned into two independent sets; we need a name for them. ■

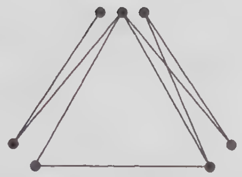


1.1.10. Definition. A graph G is **bipartite** if $V(G)$ is the union of two disjoint (possibly empty) independent sets called **partite sets** of G.

1.1.11. Example. *Scheduling and graph coloring.* Suppose we must schedule Senate committee meetings into designated weekly time periods. We cannot assign two committees to the same time if they have a common member. How many different time periods do we need?

We create a vertex for each committee, with two vertices adjacent when their committees have a common member. We must assign labels (time periods) to the vertices so the endpoints of each edge receive different labels. In the graph below, we can use one label for each of the three independent sets of vertices grouped closely together. The members of a clique must receive distinct labels, so in this example the minimum number of time periods is three.

Since we are only interested in partitioning the vertices, and the labels have no numerical value, it is convenient to call them **colors**. ■



1.1.12. Definition. The **chromatic number** of a graph G, written $\chi(G)$, is the minimum number of colors needed to label the vertices so that adjacent vertices receive different colors. A graph G is k-**partite** if $V(G)$ can be expressed as the union of k (possibly empty) independent sets.

This generalizes the idea of bipartite graphs, which are 2-partite. Vertices given the same color must form an independent set, so $\chi(G)$ is the minimum number of independent sets needed to partition $V(G)$. A graph is k-partite if and only if its chromatic number is at most k. We use the term "partite set" when referring to a set in a partition into independent sets.

We study chromatic number and graph colorings in Chapter 5. The most (in)famous problem in graph theory involves coloring of "maps".

1.1.13. Example. *Maps and coloring.* Roughly speaking, a **map** is a partition of the plane into connected regions. Can we color the regions of every map using at most four colors so that neighboring regions have different colors?

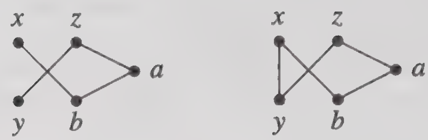
To relate map coloring to graph coloring, we introduce a vertex for each region and an edge for regions sharing a boundary. The map question asks whether the resulting graph must have chromatic number at most 4. The graph can be drawn in the plane without crossing edges; such graphs are **planar**. The graph before Definition 1.1.12 is planar; that drawing has a crossing, but another drawing has no crossings. We study planar graphs in Chapter 6. ■

1.1.14. Example. *Routes in road networks.* We can model a road network using a graph with edges corresponding to road segments between intersections. We can assign edge weights to measure distance or travel time. In this context edges do represent physical links. How can we find the shortest route from x to y? We show how to compute this in Chapter 2.

If the vertices of the graph represent our house and other places to visit, then we may want to follow a route that visits every vertex exactly once, so as to visit everyone without overstaying our welcome. We consider the existence of such a route in Chapter 7.

We need terms to describe these two types of routes in graphs. ■

1.1.15. Definition. A **path** is a simple graph whose vertices can be ordered so that two vertices are adjacent if and only if they are consecutive in the list. A **cycle** is a graph with an equal number of vertices and edges whose vertices can be placed around a circle so that two vertices are adjacent if and only if they appear consecutively along the circle.



Above we show a path and a cycle, as demonstrated by listing the vertices in the order x, b, a, z, y. Dropping one edge from a cycle produces a path. In studying the routes in road networks, we think of paths and cycles *contained* in the graph. Also, we hope that every vertex in the network can be reached from every other. The next definition makes these concepts precise.

1.1.16. Definition. A **subgraph** of a graph G is a graph H such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ and the assignment of endpoints to edges in H is the same as in G. We then write $H \subseteq G$ and say that "G **contains** H".

A graph G is **connected** if each pair of vertices in G belongs to a path; otherwise, G is **disconnected**.

The graph before Definition 1.1.12 has three subgraphs that are cycles. It is a connected graph, but the graph in Example 1.1.9 is not.

MATRICES AND ISOMORPHISM

How do we specify a graph? We can list the vertices and edges (with endpoints), but there are other useful representations. Saying that a graph is **loopless** means that multiple edges are allowed but loops are not.

1.1.17. Definition. Let G be a loopless graph with vertex set $V(G) = \{v_1, \dots, v_n\}$ and edge set $E(G) = \{e_1, \dots, e_m\}$. The **adjacency matrix** of G, written $A(G)$, is the n-by-n matrix in which entry $a_{i,j}$ is the number of edges in G with endpoints $\{v_i, v_j\}$. The **incidence matrix** $M(G)$ is the n-by-m matrix in which entry $m_{i,j}$ is 1 if v_i is an endpoint of e_j and otherwise is 0.

If vertex v is an endpoint of edge e, then v and e are **incident**. The **degree** of vertex v (in a loopless graph) is the number of incident edges.

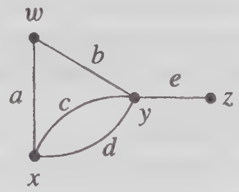
The appropriate way to define adjacency matrix, incidence matrix, or vertex degrees for graphs with loops depends on the application; Sections 1.2 and 1.3 discuss this.

1.1.18. Remark. An adjacency matrix is determined by a vertex ordering. Every adjacency matrix is **symmetric** ($a_{i,j} = a_{j,i}$ for all i, j). An adjacency matrix of a simple graph G has entries 0 or 1, with 0s on the diagonal. The degree of v is the sum of the entries in the row for v in either $A(G)$ or $M(G)$. ■

1.1.19. Example. For the loopless graph G below, we exhibit the adjacency matrix and incidence matrix that result from the vertex ordering w, x, y, z and the edge ordering a, b, c, d, e. The degree of y is 4, by viewing the graph or by summing the row for y in either matrix. ■

$$\begin{array}{c c} & \begin{matrix} w & x & y & z \end{matrix} \\ \begin{matrix} w \\ x \\ y \\ z \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

$A(G)$



G

$$\begin{array}{c c} & \begin{matrix} a & b & c & d & e \end{matrix} \\ \begin{matrix} w \\ x \\ y \\ z \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

$M(G)$

Presenting an adjacency matrix for a graph implicitly names the vertices by the order of the rows; the ith vertex corresponds to the ith row and column. Storing a graph in a computer requires naming the vertices.

Nevertheless, we want to study properties (like connectedness) that do not depend on these names. Intuitively, the structural properties of G and H will be the same if we can rename the vertices of G using the vertices in H so that G will actually become H. We make the definition precise for simple graphs. The renaming is a function from $V(G)$ to $V(H)$ that assigns each element of $V(H)$ to one element of $V(G)$, thus pairing them up. Such a function is a *one-to-one correspondence* or **bijection** (see Appendix A). Saying that the renaming turns G into H is saying that the vertex bijection preserves the adjacency relation.

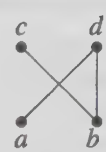
1.1.20. Definition. An **isomorphism** from a simple graph G to a simple graph H is a bijection $f: V(G) \to V(H)$ such that $uv \in E(G)$ if and only if $f(u)f(v) \in E(H)$. We say "G is **isomorphic to** H", written $G \cong H$, if there is an isomorphism from G to H.

1.1.21. Example. The graphs G and H drawn below are 4-vertex paths. Define the function $f: V(G) \to V(H)$ by $f(w) = a$, $f(x) = d$, $f(y) = b$, $f(z) = c$. To show that f is an isomorphism, we check that f preserves edges and non-edges. Note that rewriting $A(G)$ by placing the rows in the order w, y, z, x and the columns also in that order yields $A(H)$, as illustrated below; this verifies that f is an isomorphism.

Another isomorphism maps w, x, y, z to c, b, d, a, respectively. ■

G

H



$$\begin{array}{c c} & \begin{matrix} w & x & y & z \end{matrix} \\ \begin{matrix} w \\ x \\ y \\ z \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \quad \begin{array}{c c} & \begin{matrix} w & y & z & x \end{matrix} \\ \begin{matrix} w \\ y \\ z \\ x \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{array} \quad \begin{array}{c c} & \begin{matrix} a & b & c & d \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

1.1.22. Remark. *Finding isomorphisms.* As suggested in Example 1.1.21, presenting the adjacency matrices with vertices ordered so that the matrices are identical is one way to prove that two graphs are isomorphic. Applying a permutation σ to both the rows and the columns of $A(G)$ has the effect of reordering the vertices of G. If the new matrix equals $A(H)$, then the permutation yields an isomorphism. One can also verify preservation of the adjacency relation without writing out the matrices.

In order for an explicit vertex bijection to be an isomorphism from G to H, the image in H of a vertex v in G must behave in H as v does in G. For example, they must have the same degree. ∎

1.1.23.* Remark. *Isomorphism for non-simple graphs.* The definition of isomorphism extends to graphs with loops and multiple edges, but the precise statement needs the language of Definition 1.1.2.

> An **isomorphism** from G to H is a bijection f that maps $V(G)$ to $V(H)$ and $E(G)$ to $E(H)$ such each edge of G with endpoints u and v is mapped to an edge with endpoints $f(u)$ and $f(v)$.

This technicality will not concern us, because we will study isomorphism only in the context of simple graphs. ∎

Since H is isomorphic to G whenever G is isomorphic to H, we often say "G and H are isomorphic" (meaning to each other). The adjective "isomorphic" applies only to pairs of graphs; "G is isomorphic" by itself has no meaning (we respond, "isomorphic to what?"). Similarly, we may say that a set of graphs is "pairwise isomorphic" (taken two at a time), but it doesn't make sense to say "this set of graphs is isomorphic".

A **relation** on a set S is a collection of ordered pairs from S. An **equivalence relation** is a relation that is reflexive, symmetric, and transitive (see Appendix A). For example, the adjacency relation on the set of vertices of a graph is symmetric, but it is not reflexive and rarely is transitive. On the other hand, the **isomorphism relation**, consisting of the set of ordered pairs (G, H) such that G is isomorphic to H, does have all three properties.

1.1.24. Proposition. The isomorphism relation is an equivalence relation on the set of (simple) graphs.

Proof: *Reflexive property.* The identity permutation on $V(G)$ is an isomorphism from G to itself. Thus $G \cong G$.

Symmetric property. If $f\colon V(G) \to V(H)$ is an isomorphism from G to H, then f^{-1} is an isomorphism from H to G, because the statement "$uv \in E(G)$ if and only if $f(u)f(v) \in E(H)$" yields "$xy \in E(H)$ if and only if $f^{-1}(x)f^{-1}(y) \in E(H)$". Thus $G \cong H$ implies $H \cong G$.

Transitive property. Suppose that $f\colon V(F) \to V(G)$ and $g\colon V(G) \to V(H)$ are isomorphisms. We are given "$uv \in E(F)$ if and only if $f(u)f(v) \in E(G)$" and "$xy \in E(G)$ if and only if $g(x)g(y) \in E(H)$". Since f is an isomorphism, for every $xy \in E(G)$ we can find $uv \in E(F)$ such that $f(u) = x$ and $f(v) = y$. This yields "$uv \in E(F)$ if and only if $g(f(u))g(f(v)) \in E(H)$". Thus the composition $g \circ f$ is an isomorphism from F to H. We have proved that $F \cong G$ and $G \cong H$ together imply $F \cong H$. ∎

An equivalence relation partitions a set into **equivalence classes**; two elements satisfy the relation if and only if they lie in the same class.

1.1.25. Definition. An **isomorphism class** of graphs is an equivalence class of graphs under the isomorphism relation.

Paths with n vertices are pairwise isomorphic; the set of all n-vertex paths forms an isomorphism class.

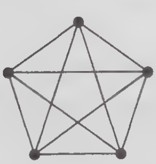
1.1.26. Remark. *"Unlabeled" graphs and isomorphism classes.* When discussing a graph G, we have a fixed vertex set, but our structural comments apply also to every graph isomorphic to G. Our conclusions are independent of the names (labels) of the vertices. Thus, we use the informal expression "unlabeled graph" to mean an isomorphism class of graphs.

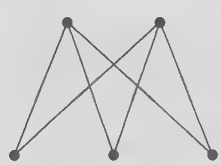
When we draw a graph, its vertices are named by their physical locations, even if we give them no other names. Hence a drawing of a graph is a member of its isomorphism class, and we just call it a graph. When we redraw a graph to display some structural aspect, we have chosen a more convenient member of the isomorphism class, still discussing the same "unlabeled graph". ∎

When discussing structure of graphs, it is convenient to have names and notation for important isomorphism classes. We want the flexibility to refer to the isomorphism class or to any representative of it.

1.1.27. Definition. The (unlabeled) path and cycle with n vertices are denoted P_n and C_n, respectively; an **n-cycle** is a cycle with n vertices. A **complete graph** is a simple graph whose vertices are pairwise adjacent; the (unlabeled) complete graph with n vertices is denoted K_n. A **complete bipartite graph** or **biclique** is a simple bipartite graph such that two vertices are adjacent if and only if they are in different partite sets. When the sets have sizes r and s, the (unlabeled) biclique is denoted $K_{r,s}$.

K_5





$K_{2,3}$

1.1.28.* Remark. We have defined a *complete graph* as a graph whose vertices are pairwise adjacent, while a *clique* is a set of pairwise adjacent vertices in a graph. Many authors use the terms interchangeably, but the distinction allows us to discuss cliques in the same language as independent sets.

In the bipartite setting, we simply use "biclique" to abbreviate "complete bipartite graph". The alternative name "biclique" is a reminder that a complete bipartite graph is generally *not* a complete graph (Exercise 1). ■

1.1.29. Remark. *A graph by any other name . . .* When we name a graph without naming its vertices, we often mean its isomorphism class. Technically, "H is a subgraph of G" means that some subgraph of G is isomorphic to H (we say "G contains a **copy** of H"). Thus C_3 is a subgraph of K_5 (every complete graph with 5 vertices has 10 subgraphs isomorphic to C_3) but not of $K_{2,3}$.

Similarly, asking whether G "is" C_n means asking whether G is isomorphic to a cycle with n vertices. ■

The structural properties of a graph are determined by its adjacency relation and hence are preserved by isomorphism. We can prove that G and H are *not* isomorphic by finding some structural property in which they differ. If they have different number of edges, or different subgraphs, or different complements, etc., then they cannot be isomorphic.

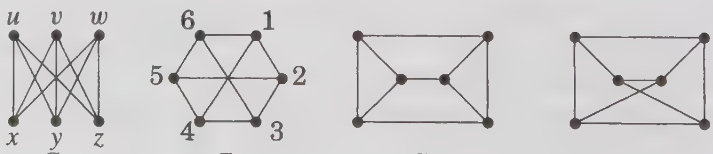
On the other hand, checking that a few structural properties are the same does not imply that $G \cong H$. To prove that $G \cong H$, we must present a bijection $f\colon V(G) \to V(H)$ that preserves the adjacency relation.

1.1.30. Example. *Isomorphic or not?* Each graph below has six vertices and nine edges and is connected, but these graphs are not pairwise isomorphic.

To prove that $G_1 \cong G_2$, we give names to the vertices, specify a bijection, and check that it preserves the adjacency relation. As labeled below, the bijection that sends u, v, w, x, y, z to $1, 3, 5, 2, 4, 6$, respectively, is an isomorphism from G_1 to G_2. The map sending u, v, w, x, y, z to $6, 4, 2, 1, 3, 5$, respectively, is another isomorphism.

Both G_1 and G_2 are bipartite; they are drawings of $K_{3,3}$ (as is G_4). The graph G_3 contains K_3, so its vertices cannot be partitioned into two independent sets. Thus G_3 is not isomorphic to the others.

Sometimes we can test isomorphism quickly using the complements. Simple graphs G and H are isomorphic if and only if their complements are isomorphic (Exercise 4). Here $\overline{G}_1, \overline{G}_2, \overline{G}_4$ all consist of two disjoint 3-cycles and are not connected, but $\overline{G}_3$ is a 6-cycle and is connected. ■

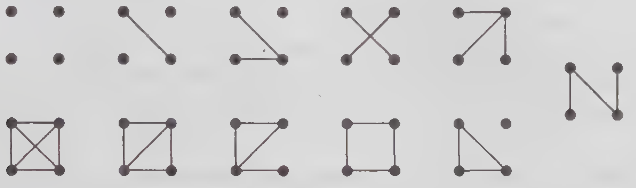


G_1 G_2 G_3 G_4

1.1.31. Example. *The number of n-vertex graphs.* When choosing two vertices from a set of size n, we can pick one and then the other but don't care about the order, so the number of ways is $n(n-1)/2$. (The notation for the number of ways to choose k elements from n elements is $\binom{n}{k}$, read "n choose k". These numbers are called **binomial coefficients**; see Appendix A for further background.)

In a simple graph with a vertex set X of size n, each vertex pair may form an edge or may not. Making the choice for each pair specifies the graph, so the number of simple graphs with vertex set X is $2^{\binom{n}{2}}$.

For example, there are 64 simple graphs on a fixed set of four vertices. These graphs form only 11 isomorphism classes. The classes appear below in complementary pairs; only P_4 is isomorphic to its complement. Isomorphism classes have different sizes, so we cannot count the isomorphism classes of n-vertex simple graphs by dividing $2^{\binom{n}{2}}$ by the size of a class. ■



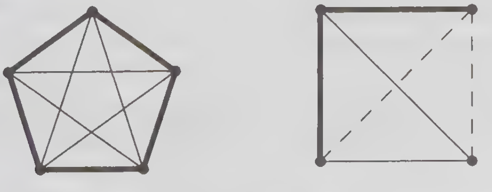
DECOMPOSITION AND SPECIAL GRAPHS

The example $P_4 \cong \overline{P}_4$ suggests a family of graph problems.

1.1.32. Definition. A graph is **self-complementary** if it is isomorphic to its complement. A **decomposition** of a graph is a list of subgraphs such that each edge appears in exactly one subgraph in the list.

An n-vertex graph H is self-complementary if and only if K_n has a decomposition consisting of two copies of H.

1.1.33. Example. We can decompose K_5 into two 5-cycles, and thus the 5-cycle is self-complementary. Any n-vertex graph and its complement decompose K_n. Also $K_{1,n-1}$ and K_{n-1} decompose K_n, even though one of these subgraphs omits a vertex. On the right below we show a decomposition of K_4 using three copies of P_3. Exercises 31–39 consider graph decompositions. ■



1.1.34.* Example. The question of which complete graphs decompose into copies of K_3 is a fundamental question in the theory of combinatorial designs.

On the left below we suggest such a decomposition for K_7. Rotating the triangle through seven positions uses each edge exactly once.

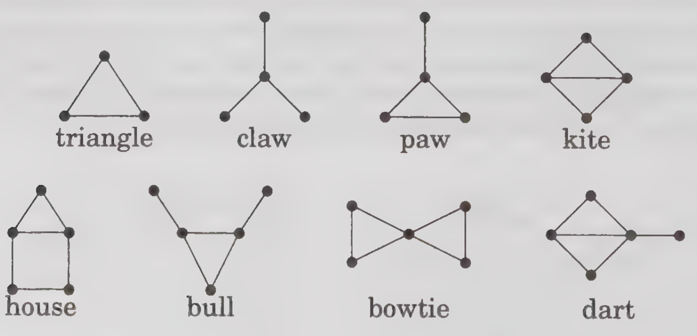
On the right we suggest a decomposition of K_6 into copies of P_4. Placing one vertex in the center groups the edges into three types: the outer 5-cycle, the inner (crossing) 5-cycle on those vertices, and the edges involving the central vertex. Each 4-vertex path in the decomposition uses one edge of each type; we rotate the picture to get the next path. ∎

We referred to a copy of K_3 as a *triangle*. Short names for graphs that arise frequently in structural discussions can be convenient.

1.1.35. Example. *The Graph Menagerie.* A catchy "name" for a graph usually comes from some drawing of the graph. We also use such a name for all other drawings, and hence it is best viewed as a name for the isomorphism class. Below we give names to several graphs with at most five vertices.

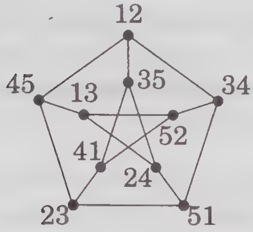
Among these the most important are the **triangle** (K_3) and the **claw** ($K_{1,3}$). We also sometimes discuss the **paw** ($K_{1,3} + e$)and the **kite** ($K_4 - e$); the others arise less frequently.

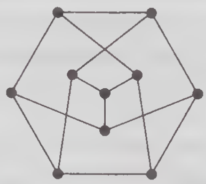
The complements of the graphs in the first row are disconnected. The complement of the house is P_5, and the bull is self-complementary. Exercise 39 asks which of these graphs can be used to decompose K_6. ∎

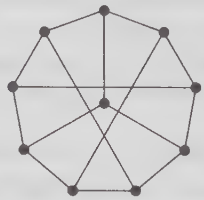


In order to decompose H into copies of G, the number of edges of G must divide the number of edges of H. This is not sufficient, since K_5 does not decompose into two copies of the kite.

1.1.36. Definition. The **Petersen graph** is the simple graph whose vertices are the 2-element subsets of a 5-element set and whose edges are the pairs of disjoint 2-element subsets.







We have drawn the Petersen graph in three ways above. It is a useful example so often that an entire book was devoted to it (Holton–Sheehan [1993]). Its properties follow from the statement of its adjacency relation that we have used as the definition.

1.1.37. Example. *Structure of the Petersen graph.* Using $[5] = \{1, 2, 3, 4, 5\}$ as our 5-element set, we write the pair $\{a, b\}$ as ab or ba. Since 12 and 34 are disjoint, they are adjacent vertices when we form the graph, but 12 and 23 are not. For each 2-set ab, there are three ways to pick a 2-set from the remaining three elements of [5], so every vertex has degree 3.

The Petersen graph consists of two disjoint 5-cycles plus edges that pair up vertices on the two 5-cycles. The disjointness definition tells us that 12, 34, 51, 23, 45 in order are the vertices of a 5-cycle, and similarly this holds for the remaining vertices 13, 52, 41, 35, 24. Also 13 is adjacent to 45, and 52 is adjacent to 34, and so on, as shown on the left above.

We use this name even when we do not specify the vertex labeling; in essence, we use "Petersen graph" to name an isomorphism class. To show that the graphs above are pairwise isomorphic, it suffices to name the vertices of each using the 2-element subsets of [5] so that in each case the adjacency relation is disjointness (Exercise 24). ■

1.1.38. Proposition. If two vertices are nonadjacent in the Petersen graph, then they have exactly one common neighbor.

Proof: Nonadjacent vertices are 2-sets sharing one element; their union S has size 3. A vertex adjacent to both is a 2-set disjoint from both. Since the 2-sets are chosen from $\{1, 2, 3, 4, 5\}$, there is exactly one 2-set disjoint from S. ■

1.1.39. Definition. The **girth** of a graph with a cycle is the length of its shortest cycle. A graph with no cycle has infinite girth.

1.1.40. Corollary. The Petersen graph has girth 5.

Proof: The graph is simple, so it has no 1-cycle or 2-cycle. A 3-cycle would require three pairwise-disjoint 2-sets, which can't occur among 5 elements.

A 4-cycle in the absence of 3-cycles would require nonadjacent vertices with two common neighbors, which Proposition 1.1.38 forbids. Finally, the vertices 12, 34, 51, 23, 45 yield a 5-cycle, so the girth is 5. ■

The Petersen graph is highly symmetric. Every permutation of $\{1, 2, 3, 4, 5\}$ generates a permutation of the 2-subsets that preserves the disjointness relation. Thus there are at least $5! = 120$ isomorphisms from the Petersen graph to itself. Exercise 43 confirms that there are no others.

1.1.41.* Definition. An **automorphism** of G is an isomorphism from G to G. A graph G is **vertex-transitive** if for every pair $u, v \in V(G)$ there is an automorphism that maps u to v.

The automorphisms of G are the permutations of $V(G)$ that can be applied to both the rows and the columns of $A(G)$ without changing $A(G)$.

1.1.42.* Example. *Automorphisms.* Let G be the path with vertex set $\{1, 2, 3, 4\}$ and edge set $\{12, 23, 34\}$. This graph has two automorphisms: the identity permutation and the permutation that switches 1 with 4 and switches 2 with 3. Interchanging vertices 1 and 2 is not an automorphism of G, although G is isomorphic to the graph with vertex set $\{1, 2, 3, 4\}$ and edge set $\{21, 13, 34\}$.

In $K_{r,s}$, permuting the vertices of one partite set does not change the adjacency matrix; this leads to $r!s!$ automorphisms. When $r = s$, we can also interchange the partite sets; $K_{t,t}$ has $2(t!)^2$ automorphisms.

The biclique $K_{r,s}$ is vertex-transitive if and only if $r = s$. If $n > 2$, then P_n is not vertex-transitive, but every cycle is vertex-transitive. The Petersen graph is vertex-transitive. ■

We can prove a statement for every vertex in a vertex-transitive graph by proving it for one vertex. Vertex-transitivity guarantees that the graph "looks the same" from each vertex.

EXERCISES

Solutions to problems generally require clear explanations written in sentences. The designations on problems have the following meanings:

"(−)" = easier or shorter than most,

"(+)" = harder or longer than most,

"(!)" = particularly useful or instructive,

"(∗)" = involves concepts marked optional in the text.

The exercise sections begin with easier problems to check understanding, ending with a line of dots. The remaining problems roughly follow the order of material in the text.

1.1.1. (−) Determine which complete bipartite graphs are complete graphs.

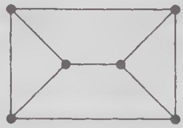
1.1.2. (−) Write down all possible adjacency matrices and incidence matrices for a 3-vertex path. Also write down an adjacency matrix for a path with six vertices and for a cycle with six vertices.

1.1.3. (−) Using rectangular blocks whose entries are all equal, write down an adjacency matrix for $K_{m,n}$.

1.1.4. (−) From the definition of isomorphism, prove that $G \cong H$ if and only if $\overline{G} \cong \overline{H}$.

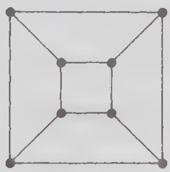
1.1.5. (−) Prove or disprove: If every vertex of a simple graph G has degree 2, then G is a cycle.

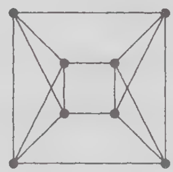
1.1.6. (−) Determine whether the graph below decomposes into copies of P_4.



1.1.7. (−) Prove that a graph with more than six vertices of odd degree cannot be decomposed into three paths.

1.1.8. (−) Prove that the 8-vertex graph on the left below decomposes into copies of $K_{1,3}$ and also into copies of P_4.

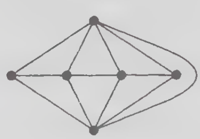




1.1.9. (−) Prove that the graph on the right above is isomorphic to the complement of the graph on the left.

1.1.10. (−) Prove or disprove: The complement of a simple disconnected graph must be connected.

1.1.11. Determine the maximum size of a clique and the maximum size of an independent set in the graph below.



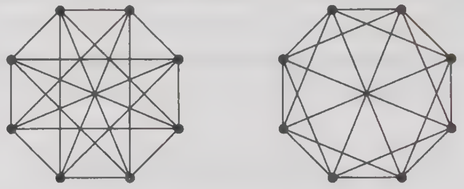
1.1.12. Determine whether the Petersen graph is bipartite, and find the size of its largest independent set.

1.1.13. Let G be the graph whose vertex set is the set of k-tuples with coordinates in $\{0, 1\}$, with x adjacent to y when x and y differ in exactly one position. Determine whether G is bipartite.

1.1.14. (!) Prove that removing opposite corner squares from an 8-by-8 checkerboard leaves a subboard that cannot be partitioned into 1-by-2 and 2-by-1 rectangles. Using the same argument, make a general statement about all bipartite graphs.

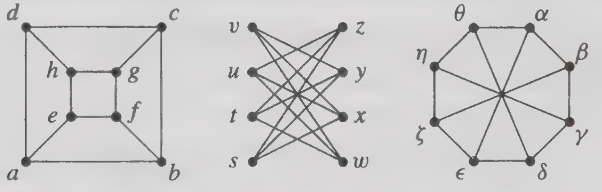
1.1.15. Consider the following four families of graphs: $A = \{\text{paths}\}$, $B = \{\text{cycles}\}$, $C = \{\text{complete graphs}\}$, $D = \{\text{bipartite graphs}\}$. For each pair of these families, determine all isomorphism classes of graphs that belong to both families.

1.1.16. Determine whether the graphs below are isomorphic.

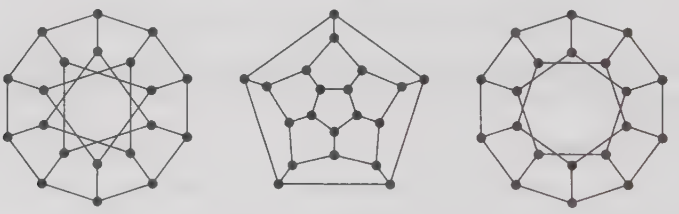


1.1.17. Determine the number of isomorphism classes of simple 7-vertex graphs in which every vertex has degree 4.

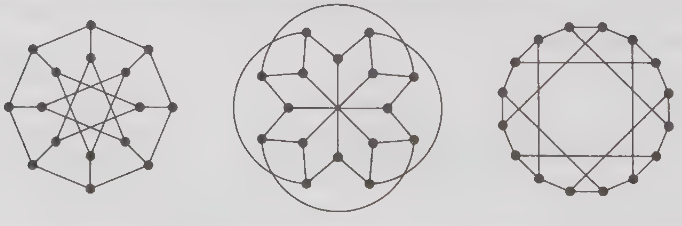
1.1.18. Determine which pairs of graphs below are isomorphic.



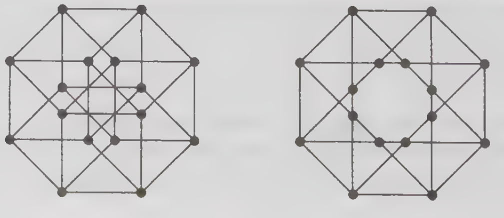
1.1.19. Determine which pairs of graphs below are isomorphic.



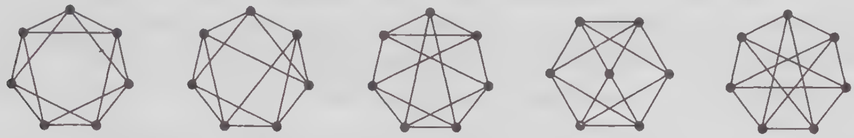
1.1.20. Determine which pairs of graphs below are isomorphic.



1.1.21. Determine whether the graphs below are bipartite and whether they are isomorphic. (The graph on the left appears on the cover of Wilson–Watkins [1990].)



1.1.22. (!) Determine which pairs of graphs below are isomorphic, presenting the proof by testing the smallest possible number of pairs.

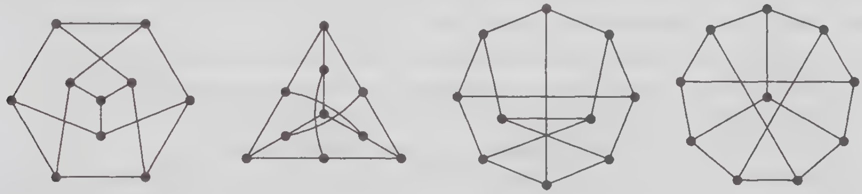


1.1.23. In each class below, determine the smallest n such that there exist nonisomorphic n-vertex graphs having the same list of vertex degrees.

(a) all graphs, (b) loopless graphs, (c) simple graphs.

(Hint: Since each class contains the next, the answers form a nondecreasing triple. For part (c), use the list of isomorphism classes in Example 1.1.31.)

1.1.24. (!) Prove that the graphs below are all drawings of the Petersen graph (Definition 1.1.36). (Hint: Use the disjointness definition of adjacency.)



1.1.25. (!) Prove that the Petersen graph has no cycle of length 7.

1.1.26. (!) Let G be a graph with girth 4 in which every vertex has degree k. Prove that G has at least $2k$ vertices. Determine all such graphs with exactly $2k$ vertices.

1.1.27. (!) Let G be a graph with girth 5. Prove that if every vertex of G has degree at least k, then G has at least $k^2 + 1$ vertices. For $k = 2$ and $k = 3$, find one such graph with exactly $k^2 + 1$ vertices.

1.1.28. (+) *The Odd Graph* O_k. The vertices of the graph O_k are the k-element subsets of $\{1, 2, \ldots, 2k + 1\}$. Two vertices are adjacent if and only if they are disjoint sets. Thus O_2 is the Petersen graph. Prove that the girth of O_k is 6 if $k \geq 3$.

1.1.29. Prove that every set of six people contains (at least) three mutual acquaintances or three mutual strangers.

1.1.30. Let G be a simple graph with adjacency matrix A and incidence matrix M. Prove that the degree of v_i is the ith diagonal entry in A^2 and in MM^T. What do the entries in position (i, j) of A^2 and MM^T say about G?

1.1.31. (!) Prove that a self-complementary graph with n vertices exists if and only if n or $n - 1$ is divisible by 4. (Hint: When n is divisible by 4, generalize the structure of P_4 by splitting the vertices into four groups. For $n \equiv 1 \bmod 4$, add one vertex to the graph constructed for $n - 1$.)

1.1.32. Determine which bicliques decompose into two isomorphic subgraphs.

1.1.33. For $n = 5$, $n = 7$, and $n = 9$, decompose K_n into copies of C_n.

1.1.34. (!) Decompose the Petersen graph into three connected subgraphs that are pairwise isomorphic. Also decompose it into copies of P_4.

1.1.35. (!) Prove that K_n decomposes into three pairwise-isomorphic subgraphs if and only if $n+1$ is not divisible by 3. (Hint: For the case where n is divisible by 3, split the vertices into three sets of equal size.)

1.1.36. Prove that if K_n decomposes into triangles, then $n-1$ or $n-3$ is divisible by 6.

1.1.37. Let G be a graph in which every vertex has degree 3. Prove that G has no decomposition into paths that each have at least 5 vertices.

1.1.38. (!) Let G be a simple graph in which every vertex has degree 3. Prove that G decomposes into claws if and only if G is bipartite.

1.1.39. (+) Determine which of the graphs in Example 1.1.35 can be used to form a decomposition of K_6 into pairwise-isomorphic subgraphs. (Hint: Each graph that is not excluded by some divisibility condition works.)

1.1.40. (∗) Count the automorphisms of P_n, C_n, and K_n.

1.1.41. (∗) Construct a simple graph with six vertices that has only one automorphism. Construct a simple graph that has exactly three automorphisms. (Hint: Think of a rotating triangle with appendages to prevent flips.)

1.1.42. (∗) Verify that the set of automorphisms of G has the following properties:
a) The composition of two automorphisms is an automorphism.
b) The identity permutation is an automorphism.
c) The inverse of an automorphism is also an automorphism.
d) Composition of automorphisms satisfies the associative property.
(Comment: Thus the set of automorphisms satisfies the defining properties for a *group*.)

1.1.43. (∗) *Automorphisms of the Petersen graph.* Consider the Petersen graph as defined by disjointness of 2-sets in $\{1,2,3,4,5\}$. Prove that every automorphism maps the 5-cycle with vertices $12, 34, 51, 23, 45$ to a 5-cycle with vertices ab, cd, ea, bc, de determined by a permutation of $\{1,2,3,4,5\}$ taking elements 1,2,3,4,5 to a, b, c, d, e, respectively. (Comment: This implies that there are only 120 automorphisms.)

1.1.44. (∗) The Petersen graph has even more symmetry than vertex-transitivity. Let $P = (u_0, u_1, u_2, u_3)$ and $Q = (v_0, v_1, v_2, v_3)$ be paths with three edges in the Petersen graph. Prove that there is exactly one automorphism of the Petersen graph that maps u_i into v_i for $i = 0, 1, 2, 3$. (Hint: Use the disjointness description.)

1.1.45. (∗) Construct a graph with 12 vertices in which every vertex has degree 3 and the only automorphism is the identity.

1.1.46. (∗) *Edge-transitivity.* A graph G is **edge-transitive** if for all $e, f \in E(G)$ there is an automorphism of G that maps the endpoints of e to the endpoints of f (in either order). Prove that the graphs of Exercise 1.1.21 are vertex-transitive and edge-transitive. (Comment: Complete graphs, bicliques, and the Petersen graph are edge-transitive.)

1.1.47. (∗) *Edge-transitive versus vertex-transitive.*

a) Let G be obtained from K_n with $n \geq 4$ by replacing each edge of K_n with a path of two edges through a new vertex of degree 2. Prove that G is edge-transitive but not vertex-transitive.

b) Suppose that G is edge-transitive but not vertex-transitive and has no vertices of degree 0. Prove that G is bipartite.

c) Prove that the graph in Exercise 1.1.6 is vertex-transitive but not edge-transitive.

1.2. Paths, Cycles, and Trails

In this section we return to the Königsberg Bridge Problem, determining when it is possible to traverse all the edges of a graph. We also we develop useful properties of connection, paths, and cycles.

Before embarking on this, we review an important technique of proof. Many statements in graph theory can be proved using the principle of induction. Readers unfamiliar with induction should read the material on this proof technique in Appendix A. Here we describe the form of induction that we will use most frequently, in order to familiarize the reader with a template for proof.

1.2.1. Theorem. (Strong Principle of Induction). Let $P(n)$ be a statement with an integer parameter n. If the following two conditions hold, then $P(n)$ is true for each positive integer n.
1) $P(1)$ is true.
2) For all $n > 1$, "$P(k)$ is true for $1 \leq k < n$" implies "$P(n)$ is true".

Proof: We ASSUME the **Well Ordering Property** for the positive integers: every nonempty set of positive integers has a least element. Given this, suppose that $P(n)$ fails for some n. By the Well Ordering Property, there is a least n such that $P(n)$ fails. Statement (1) ensures that this value cannot be 1. Statement (2) ensures that this value cannot be greater than 1. The contradiction implies that $P(n)$ holds for every positive integer n. ■

In order to apply induction, we verify (1) and (2) for our sequence of statements. Verifying (1) is the **basis step** of the proof; verifying (2) is the **induction step**. The statement "$P(k)$ is true for all $k < n$" is the **induction hypothesis**, because it is the hypothesis of the implication proved in the induction step. The variable that indexes the sequence of statements is the **induction parameter**.

The induction parameter may be any integer function of the instances of our problem, such as the number of vertices or edges in a graph. We say that we are using "induction on n" when the induction parameter is n.

There are many ways to phrase inductive proofs. We can start at 0 to prove a statement for nonnegative integers. When our proof of $P(n)$ in the induction step makes use only of $P(n-1)$ from the induction hypothesis, the technique is called "ordinary" induction; making use of all previous statements is "strong" induction. We seldom distinguish between strong induction and ordinary induction; they are equivalent (see Appendix A).

Most students first learn ordinary induction in the following phrasing: 1) verify that $P(n)$ is true when $n = 1$, and 2) prove that if $P(n)$ is true when n is k, then $P(n)$ is also true when n is $k+1$. Proving $P(k+1)$ from $P(k)$ for $k \geq 1$ is equivalent to proving $P(n)$ from $P(n-1)$ for $n > 1$.

When we focus on proving the statement for the parameter value n in the induction step, we need not decide at the outset whether we are using strong induction or ordinary induction. The language is also simpler, since we avoid introducing a new name for the parameter. In Section 1.3 we will explain why this phrasing is also less prone to error.

CONNECTION IN GRAPHS

As defined in Definition 1.1.15, paths and cycles are graphs; a path *in* a graph G is a subgraph of G that is a path (similarly for cycles). We introduce further definitions to model other movements in graphs. A tourist wandering in a city (or a Königsberg pedestrian) may want to allow vertex repetitions but avoid edge repetitions. A mail carrier delivers mail to houses on both sides of the street and hence traverses each edge twice.

1.2.2. Definition. A **walk** is a list $v_0, e_1, v_1, \ldots, e_k, v_k$ of vertices and edges such that, for $1 \le i \le k$, the edge e_i has endpoints v_{i-1} and v_i. A **trail** is a walk with no repeated edge. A u, v**-walk** or u, v**-trail** has first vertex u and last vertex v; these are its **endpoints**. A u, v**-path** is a path whose vertices of degree 1 (its **endpoints**) are u and v; the others are **internal vertices**.

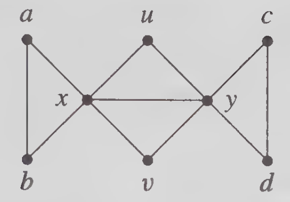
The **length** of a walk, trail, path, or cycle is its number of edges. A walk or trail is **closed** if its endpoints are the same.

1.2.3. Example. In the Königsberg graph (Example 1.1.1), the list $x, e_2, w, e_5, y, e_6, x, e_1, w, e_2, x$ is a closed walk of length 5; it repeats edge e_2 and hence is not a trail. Deleting the last edge and vertex yields a trail of length 4; it repeats vertices but not edges. The subgraph consisting of edges e_1, e_5, e_6 and vertices x, w, y is a cycle of length 3; deleting one of its edges yields a path. Two edges with the same endpoints (such as e_1 and e_2) form a cycle of length 2. A loop is a cycle of length 1. ■

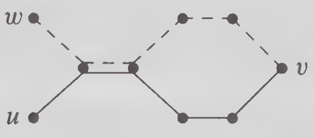
The reason for listing the edges in a walk is to distinguish among multiple edges when a graph is not simple. In a simple graph, a walk (or trail) is completely specified by its ordered list of vertices. We usually name a path, cycle, trail, or walk in a simple graph by listing only its vertices in order, even though it consists of both vertices and edges. When discussing a cycle, we can start at any vertex and do not repeat the first vertex at the end. We can use parentheses to clarify that this is a cycle and not a path.

1.2.4. Example. We illustrate the simplified notation in a simple graph. In the graph below, $a, x, a, x, u, y, c, d, y, v, x, b, a$ specifies a closed walk of length 12. Omitting the first two steps yields a closed trail.

The graph has five cycles: (a, b, x), (c, y, d), (u, x, y), (x, y, v), (u, x, v, y). The u, v-trail u, y, c, d, y, x, v contains the edges of the u, v-path u, y, x, v, but not of the u, v-path u, y, v. ■



Suppose we follow a path from u to v in a graph and then follow a path from v to w. The result need not be a u, w-path, because the u, v-path and v, w-path may have a common internal vertex. Nevertheless, the list of vertices and edges that we visit does form a u, w-walk. In the illustration below, the u, w-walk contains a u, w-path. Saying that a walk W **contains** a path P means that the vertices and edges of P occur as a sublist of the vertices and edges of W, in order but not necessarily consecutive.

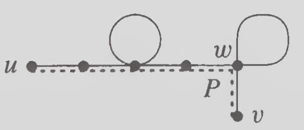


1.2.5. Lemma. Every u, v-walk contains a u, v-path.

Proof: We prove the statement by induction on the length l of a u, v-walk W.

Basis step: $l = 0$. Having no edge, W consists of a single vertex ($u = v$). This vertex is a u, v-path of length 0.

Induction step: $l \geq 1$. We suppose that the claim holds for walks of length less than l. If W has no repeated vertex, then its vertices and edges form a u, v-path. If W has a repeated vertex w, then deleting the edges and vertices between appearances of w (leaving one copy of w) yields a shorter u, v-walk W' contained in W. By the induction hypothesis, W' contains a u, v-path P, and this path P is contained in W. ■



Exercise 13b develops a shorter proof. We apply the lemma to properties of connection.

1.2.6. Definition. A graph G is **connected** if it has a u, v-path whenever $u, v \in V(G)$ (otherwise, G is **disconnected**). If G has a u, v-path, then u is **connected to** v in G. The **connection relation** on $V(G)$ consists of the ordered pairs (u, v) such that u is connected to v.

"Connected" is an adjective we apply only to graphs and to pairs of vertices (we never say "v is disconnected" when v is a vertex). The phrase "u is connected to v" is convenient when writing proofs, but in adopting it we must clarify the distinction between connection and adjacency:

G has a u,v-path	$uv \in E(G)$
u and v are connected	u and v are adjacent
u is connected to v	u is joined to v
	u is adjacent to v

1.2.7. Remark. By Lemma 1.2.5, we can prove that a graph is connected by showing that from each vertex there is a walk to one particular vertex.

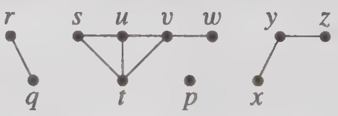
By Lemma 1.2.5, the connection relation is transitive: if G has a u,v-path and a v,w-path, then G has a u,w-path. It is also reflexive (paths of length 0) and symmetric (paths are reversible), so it is an equivalence relation. ■

Our next definition leads us to the equivalence classes of the connection relation. A *maximal* connected subgraph of G is a subgraph that is connected and is not contained in any other connected subgraph of G.

1.2.8. Definition. The **components** of a graph G are its maximal connected subgraphs. A component (or graph) is **trivial** if it has no edges; otherwise it is **nontrivial**. An **isolated vertex** is a vertex of degree 0.

The equivalence classes of the connection relation on $V(G)$ are the vertex sets of the components of G. An isolated vertex forms a trivial component, consisting of one vertex and no edge.

1.2.9. Example. The graph below has four components, one being an isolated vertex. The vertex sets of the components are $\{p\}$, $\{q,r\}$, $\{s,t,u,v,w\}$, and $\{x,y,z\}$; these are the equivalence classes of the connection relation. ■



1.2.10. Remark. Components are pairwise disjoint; no two share a vertex. Adding an edge with endpoints in distinct components combines them into one component. Thus adding an edge decreases the number of components by 0 or 1, and deleting an edge increases the number of components by 0 or 1. ■

1.2.11. Proposition. Every graph with n vertices and k edges has at least $n-k$ components.

Proof: An n-vertex graph with no edges has n components. By Remark 1.2.10, each edge added reduces this by at most 1, so when k edges have been added the number of components is still at least $n-k$. ■

Deleting a vertex or an edge can increase the number of components. Although deleting an edge can only increase the number of components by 1, deleting a vertex can increase it by many (consider the biclique $K_{1,m}$). When we obtain a subgraph by deleting a vertex, it must be a graph, so deleting the vertex also deletes all edges incident to it.

1.2.12. Definition. A **cut-edge** or **cut-vertex** of a graph is an edge or vertex whose deletion increases the number of components. We write $G - e$ or $G - M$ for the subgraph of G obtained by deleting an edge e or set of edges M. We write $G - v$ or $G - S$ for the subgraph obtained by deleting a vertex v or set of vertices S. An **induced subgraph** is a subgraph obtained by deleting a set of vertices. We write $G[T]$ for $G - \overline{T}$, where $\overline{T} = V(G) - T$; this is the subgraph of G **induced by** T.

When $T \subseteq V(G)$, the induced subgraph $G[T]$ consists of T and all edges whose endpoints are contained in T. The full graph is itself an induced subgraph, as are individual vertices. A set S of vertices is an independent set if and only if the subgraph induced by it has no edges.

1.2.13. Example. The graph of Example 1.2.9 has cut-vertices v and y. Its cut-edges are qr, vw, xy, and yz. (When we delete an edge, its endpoints remain.)

This graph has C_4 and P_5 as subgraphs but *not* as induced subgraphs. The subgraph induced by $\{s, t, u, v\}$ is a kite; the 4-vertex paths on these vertices are not induced subgraphs. The graph P_4 does occur as an induced subgraph; it is the subgraph induced by $\{s, t, v, w\}$ (also by $\{s, u, v, w\}$). ■

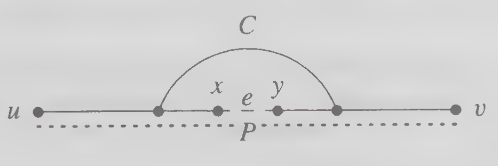
Next we characterize cut-edges in terms of cycles.

1.2.14. Theorem. An edge is a cut-edge if and only if it belongs to no cycle.

Proof: Let e be an edge in a graph G (with endpoints x, y), and let H be the component containing e. Since deletion of e affects no other component, it suffices to prove that $H - e$ is connected if and only if e belongs to a cycle.

First suppose that $H - e$ is connected. This implies that $H - e$ contains an x, y-path, and this path completes a cycle with e.

Now suppose that e lies in a cycle C. Choose $u, v \in V(H)$. Since H is connected, H has a u, v-path P. If P does not contain e, then P exists in $H - e$. If P contains e, suppose by symmetry that x is between u and y on P. Since $H - e$ contains a u, x-path along P, an x, y-path along C, and a y, v-path along P, the transitivity of the connection relation implies that $H - e$ has a u, v-path. We did this for all $u, v \in V(H)$, so $H - e$ is connected. ■



BIPARTITE GRAPHS

Our next goal is to characterize bipartite graphs using cycles. Characterizations are equivalence statements, like Theorem 1.2.14. When two conditions are equivalent, checking one also yields the other for free.

Characterizing a class $\mathbf{G}$ by a condition P means proving the equivalence "$G \in \mathbf{G}$ if and only if G satisfies P". In other words, P is both a **necessary** and a **sufficient** condition for membership in $\mathbf{G}$.

Necessity	Sufficiency
$G \in \mathbf{G}$ *only if* G satisfies P	$G \in \mathbf{G}$ *if* G satisfies P
$G \in \mathbf{G} \Rightarrow G$ satisfies P	G satisfies P $\Rightarrow G \in \mathbf{G}$

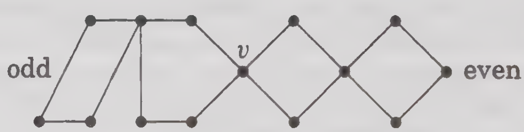
Recall that a loop is a cycle of length 1; also two distinct edges with the same endpoints form a cycle of length 2. A walk is **odd** or **even** as its length is odd or even. As in Lemma 1.2.5, a closed walk **contains** a cycle C if the vertices and edges of C occur as a sublist of W, in cyclic order but not necessarily consecutive. We can think of a closed walk or a cycle as starting at any vertex; the next lemma requires this viewpoint.

1.2.15. Lemma. Every closed odd walk contains an odd cycle.

Proof: We use induction on the length l of a closed odd walk W.

Basis step: $l = 1$. A closed walk of length 1 traverses a cycle of length 1.

Induction step: $l > 1$. Assume the claim for closed odd walks shorter than W. If W has no repeated vertex (other than first = last), then W itself forms a cycle of odd length. If vertex v is repeated in W, then we view W as starting at v and break W into two v, v-walks. Since W has odd length, one of these is odd and the other is even. The odd one is shorter than W. By the induction hypothesis, it contains an odd cycle, and this cycle appears in order in W. ■



1.2.16. Remark. A closed even walk need not contain a cycle; it may simply repeat. Nevertheless, if an edge e appears *exactly once* in a closed walk W, then W does contain a cycle through e. Let x, y be the endpoints of e. Deleting e from W leaves an x, y-walk that avoids e. By Lemma 1.2.5, this walk contains an x, y-path, and this path completes a cycle with e. (See Exercises 15–16.) ■

Lemma 1.2.15 will help us characterize bipartite graphs.

1.2.17. Definition. A **bipartition** of G is a specification of two disjoint independent sets in G whose union is $V(G)$. The statement "Let G be a bipartite graph with bipartition X, Y" specifies one such partition. An **X, Y-bigraph** is a bipartite graph with bipartition X, Y.

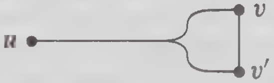
The sets of a bipartition are partite sets (Definition 1.1.10). A disconnected bipartite graph has more than one bipartition. A connected bipartite graph has only one bipartition, except for interchanging the two sets (Exercise 7).

1.2.18. Theorem. (König [1936]) A graph is bipartite if and only if it has no odd cycle.

Proof: *Necessity.* Let G be a bipartite graph. Every walk alternates between the two sets of a bipartition, so every return to the original partite set happens after an even number of steps. Hence G has no odd cycle.

Sufficiency. Let G be a graph with no odd cycle. We prove that G is bipartite by constructing a bipartition of each nontrivial component. Let u be a vertex in a nontrivial component H. For each $v \in V(H)$, let $f(v)$ be the minimum length of a u, v-path. Since H is connected, $f(v)$ is defined for each $v \in V(H)$.

Let $X = \{v \in V(H)\colon f(v) \text{ is even}\}$ and $Y = \{v \in V(H)\colon f(v) \text{ is odd}\}$. An edge v, v' within X or Y would create a closed odd walk using a shortest u, v-path, the edge vv', and the reverse of a shortest u, v'-path. By Lemma 1.2.15, such a walk must contain an odd cycle, which contradicts our hypothesis. Hence X and Y are independent sets. Also $X \cup Y = V(H)$, so H is an X, Y-bigraph. ■

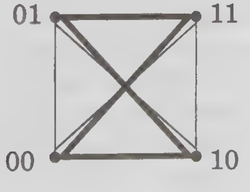


1.2.19. Remark. *Testing whether a graph is bipartite.* Theorem 1.2.18 implies that whenever a graph G is not bipartite, we can prove this statement by presenting an odd cycle in G. This is much easier than examining all possible bipartitions to prove that none work. When we want to prove that G *is* bipartite, we define a bipartition and prove that the two sets are independent; this is easier than examining all cycles. ■

We consider one application.

1.2.20. Definition. The **union** of graphs $G_1, \ldots, G_k$, written $G_1 \cup \cdots \cup G_k$, is the graph with vertex set $\bigcup_{i=1}^{k} V(G_i)$ and edge set $\bigcup_{i=1}^{k} E(G_i)$.

1.2.21. Example. Below we show K_4 as the union of two 4-cycles. When a graph G is expressed as the union of two or more subgraphs, an edge of G can belong to many of them. This distinguishes union from decomposition, where each edge belongs to only one subgraph in the list. ■



1.2.22. Example. Consider an air traffic system with k airlines. Suppose that
1) direct service between two cities means round-trip direct service, and
2) each pair of cities has direct service from at least one airline.

Suppose also that no airline can schedule a cycle through an odd number of cities. In terms of k, what is the maximum number of cities in the system?

By Theorem 1.2.18, we seek the largest n such that K_n can be expressed as the union of k bipartite graphs, one for each airline. The answer is 2^k. ■

1.2.23. Theorem. The complete graph K_n can be expressed as the union of k bipartite graphs if and only if $n \le 2^k$.

Proof: We use induction on k. Basis step: $k = 1$. Since K_3 has an odd cycle and K_2 does not, K_n is itself a bipartite graph if and only if $n \le 2$.

Induction step: $k > 1$. We prove each implication using the induction hypothesis. Suppose first that $K_n = G_1 \cup \cdots \cup G_k$, where each G_i is bipartite. We partition the vertex set into two sets X, Y such that G_k has no edge within X or within Y. The union of the other $k-1$ bipartite subgraphs must cover the complete subgraphs induced by X and by Y. Applying the induction hypothesis to each yields $|X| \le 2^{k-1}$ and $|Y| \le 2^{k-1}$, so $n \le 2^{k-1} + 2^{k-1} = 2^k$.

Conversely, suppose that $n \le 2^k$. We partition the vertex set into subsets X, Y, each of size at most 2^{k-1}. By the induction hypothesis, we can cover the complete subgraph induced by either subset with $k-1$ bipartite subgraphs. The union of the ith such subgraph on X with the ith such subgraph on Y is a bipartite graph. Hence we obtain $k-1$ bipartite graphs whose union consists of the complete subgraphs induced by X and Y. The remaining edges are those of the biclique with bipartition X, Y. Letting this be the kth bipartite subgraph completes the construction. ■

This theorem can also be proved without induction by encoding the vertices as binary k-tuples (Exercise 31).

EULERIAN CIRCUITS

We return to our analysis of the Königsberg Bridge Problem. What the people of Königsberg wanted was a closed trail containing all the edges in a graph. As we have observed, a necessary condition for existence of such a trail is that all vertex degrees be even. Also it is necessary that all edges belong to the same component of the graph.

The Swiss mathematician Leonhard Euler (pronounced "oiler") stated [1736] that these conditions are also sufficient. In honor of his contribution, we associate his name with such graphs. Euler's paper appeared in 1741 but gave no proof that the obvious necessary conditions are sufficient. Hierholzer [1873] gave the first complete published proof. The graph we drew in Example 1.1.1 to model the city did not appear in print until 1894 (see Wilson [1986] for a discussion of the historical record).

1.2.24. Definition. A graph is **Eulerian** if it has a closed trail containing all edges. We call a closed trail a **circuit** when we do not specify the first vertex but keep the list in cyclic order. An **Eulerian circuit** or **Eulerian trail** in a graph is a circuit or trail containing all the edges.

An **even graph** is a graph with vertex degrees all even. A vertex is **odd** **[even]** when its degree is odd [even].

Our discussion of Eulerian circuits applies also to graphs with loops; we extend the notion of vertex degree to graphs with loops by letting each loop contribute 2 to the degree of its vertex. This does not change the parity of the degree, and the presence of a loop does not affect whether a graph has an Eulerian circuit unless it is a loop in a component with one vertex.

Our proof of the characterization of Eulerian graphs uses a lemma. A **maximal path** in a graph G is a path P in G that is not contained in a longer path. When a graph is finite, no path can extend forever, so maximal (non-extendible) paths exist.

1.2.25. Lemma. If every vertex of a graph G has degree at least 2, then G contains a cycle.

Proof: Let P be a maximal path in G, and let u be an endpoint of P. Since P cannot be extended, every neighbor of u must already be a vertex of P. Since u has degree at least 2, it has a neighbor v in $V(P)$ via an edge not in P. The edge uv completes a cycle with the portion of P from v to u. ■

Note the importance of finiteness. If $V(G) = \mathbb{Z}$ and $E(G) = \{ij\colon |i - j| = 1\}$, then every vertex of G has degree 2, but G has no cycle (and no non-extendible path). We avoid such examples by assuming that all graphs in this book are finite, with rare explicit exceptions.

1.2.26. Theorem. A graph G is Eulerian if and only if it has at most one nontrivial component and its vertices all have even degree.

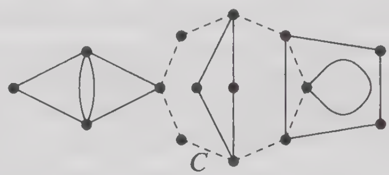
Proof: *Necessity.* Suppose that G has an Eulerian circuit C. Each passage of C through a vertex uses two incident edges, and the first edge is paired with the last at the first vertex. Hence every vertex has even degree. Also, two edges can be in the same trail only when they lie in the same component, so there is at most one nontrivial component.

Sufficiency. Assuming that the condition holds, we obtain an Eulerian circuit using induction on the number of edges, m.

Basis step: $m = 0$. A closed trail consisting of one vertex suffices.

Induction step: $m > 0$. With even degrees, each vertex in the nontrivial component of G has degree at least 2. By Lemma 1.2.25, the nontrivial component has a cycle C. Let G' be the graph obtained from G by deleting $E(C)$.

Since C has 0 or 2 edges at each vertex, each component of G' is also an even graph. Since each component also is connected and has fewer than m edges, we can apply the induction hypothesis to conclude that each component of G' has an Eulerian circuit. To combine these into an Eulerian circuit of G, we traverse C, but when a component of G' is entered for the first time we detour along an Eulerian circuit of that component. This circuit ends at the vertex where we began the detour. When we complete the traversal of C, we have completed an Eulerian circuit of G. ■



Perhaps as important as the characterization of Eulerian graphs is what the method of proof says about even graphs.

1.2.27. Proposition. Every even graph decomposes into cycles.

Proof: In the proof of Theorem 1.2.26, we noted that every even nontrivial graph has a cycle, and that the deletion of a cycle leaves an even graph. Thus this proposition follows by induction on the number of edges. ■

In the characterization of Eulerian circuits, the necessity of the condition is easy to see. This also holds for the characterization of bipartite graphs by absence of odd cycles and for many other characterizations. Nash-Williams and others popularized a mnemonic for such theorems: **TONCAS**, meaning "The Obvious Necessary Conditions are Also Sufficient".

The proof of Lemma 1.2.25 is an example of an important technique of proof in graph theory that we call **extremality**. When considering structures of a given type, choosing an example that is extreme in some sense may yield useful additional information. For example, since a maximal path P cannot be extended, we obtain the extra information that every neighbor of an endpoint of P belongs to $V(P)$.

In a sense, making an extremal choice goes directly to the important case. In Lemma 1.2.25, we could start with any path. If it is extendible, then we extend it. If not, then something important happens. We illustrate the technique with several examples, and Exercises 37–42 also use extremality. We begin by strengthening Lemma 1.2.25 for simple graphs.

1.2.28. Proposition. If G is a simple graph in which every vertex has degree at least k, then G contains a path of length at least k. If $k \ge 2$, then G also contains a cycle of length at least $k + 1$.

Proof: Let u be an endpoint of a maximal path P in G. Since P does not extend, every neighbor of u is in $V(P)$. Since u has at least k neighbors and G is simple,

P therefore has at least k vertices other than u and has length at least k. If $k \geq 2$, then the edge from u to its farthest neighbor v along P completes a sufficiently long cycle with the portion of P from v to u. ■

1.2.29. Proposition. Every graph with a nonloop edge has at least two vertices that are not cut-vertices.

Proof: If u is an endpoint of a maximal path P in G, then the neighbors of u lie on P. Since $P - u$ is connected in $G - u$, the neighbors of u belong to a single component of $G - u$, and u is not a cut-vertex. ■

1.2.30. Remark. Note the difference between "maximal" and "maximum". As adjectives, **maximum** means "maximum-sized", and **maximal** means "no larger one contains this one". Every maximum path is a maximal path, but maximal paths need not have maximum length. Similarly, the biclique $K_{r,s}$ has two maximal independent sets, but when $r \neq s$ it has only one maximum independent set. When describing numbers rather than containment, the meanings are the same; maximum vertex degree = maximal vertex degree.

Besides maximal or maximum paths or independent sets, other extremal aspects include vertices of minimum or maximum degree, the first vertex where two paths diverge, maximal connected subgraphs (components), etc. In a connected graph G with disjoint sets $S, T \subset V(G)$, we can obtain a path from S to T having only its endpoints in $S \cup T$ by choosing a shortest path from S to T; Exercise 40 applies this. Exercise 37 uses extremality for a short proof of the transitivity of the connection relation. ■

Many proofs using induction can be phrased using extremality, and many proofs using extremality can be done by induction. To underscore the interplay, we reprove the characterization of Eulerian graphs using extremality directly.

1.2.31. Lemma. In an even graph, every maximal trail is closed.

Proof: Let T be a maximal trail in an even graph. Every passage of T through a vertex v uses two edges at v, none repeated. Thus when arriving at a vertex v other than its initial vertex, T has used an odd number of edges incident to v. Since v has even degree, there remains an edge on which T can continue.

Hence T can only end at its initial vertex. In a finite graph, T must indeed end. We conclude that a maximal trail must be closed. ■

1.2.32. Theorem 1.2.26—Second Proof. We prove TONCAS. In a graph G satisfying the conditions, let T be a trail of maximum length; T must also be a maximal trail. By Lemma 1.2.31, T is closed.

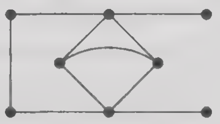
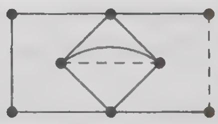
Suppose that T omits some edge e of G. Since G has only one nontrivial component, G has a shortest path from e to the vertex set of T. Hence some edge e' not in T is incident to some vertex v of T.

Since T is closed, there is a trail T' that starts and ends at v and uses the same edges as T. We now extend T' along e' to obtain a longer trail than T. This contradicts the choice of T, and hence T traverses all edges of G. ■

This proof and the resulting construction procedure (Exercise 12) are similar to those of Hierholzer [1873]. Exercise 35 develops another proof.

Later chapters contain several applications of the statement that every connected even graph has an Eulerian circuit. Here we give a simple one. When drawing a figure G on paper, how many times must we stop and move the pen? We are not allowed to repeat segments of the drawing, so each visit to the paper contributes a trail. Thus we seek a decomposition of G into the minimum number of trails. We may reduce the problem to connected graphs, since the number of trails needed to draw G is the sum of the number needed to draw each component.

For example, the graph G below has four odd vertices and decomposes into two trails. Adding the dashed edges on the right makes it Eulerian.

G


G′

1.2.33. Theorem. For a connected nontrivial graph with exactly $2k$ odd vertices, the minimum number of trails that decompose it is $\max\{k, 1\}$.

Proof: A trail contributes even degree to every vertex, except that a non-closed trail contributes odd degree to its endpoints. Therefore, a partition of the edges into trails must have some non-closed trail ending at each odd vertex. Since each trail has only two ends, we must use at least k trails to satisfy $2k$ odd vertices. We also need at least one trail since G has an edge, and Theorem 1.2.26 implies that one trail suffices when $k = 0$.

It remains to prove that k trails suffice when $k > 0$. Given such a graph G, we pair up the odd vertices in G (in any way) and form G' by adding for each pair an edge joining its two vertices, as illustrated above. The resulting graph G' is connected and even, so by Theorem 1.2.26 it has an Eulerian circuit C. As we traverse C in G', we start a new trail in G each time we traverse an edge of $G' - E(G)$. This yields k trails decomposing G. ■

We prove theorems in general contexts to avoid work. The proof of Theorem 1.2.33 illustrates this; by transforming G into a graph where Theorem 1.2.26 applies, we avoid repeating the basic argument of Theorem 1.2.26. Exercise 33 requests a proof of Theorem 1.2.33 directly by induction.

Note that Theorem 1.2.33 considers only graphs having an even number of vertices of odd degree. Our first result in the next section explains why.

EXERCISES

Most problems in this book require proofs. Words like "construct", "show", "obtain", "determine", etc., explicitly state that proof is required. Disproof by providing a counterexample requires confirming that it is a counterexample.

1.2.1. (−) Determine whether the statements below are true or false.
a) Every disconnected graph has an isolated vertex.
b) A graph is connected if and only if some vertex is connected to all other vertices.
c) The edge set of every closed trail can be partitioned into edge sets of cycles.
d) If a maximal trail in a graph is not closed, then its endpoints have odd degree.

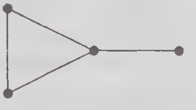
1.2.2. (−) Determine whether K_4 contains the following (give an example or a proof of non-existence).
a) A walk that is not a trail.
b) A trail that is not closed and is not a path.
c) A closed trail that is not a cycle.

1.2.3. (−) Let G be the graph with vertex set $\{1, \ldots, 15\}$ in which i and j are adjacent if and only if their greatest common factor exceeds 1. Count the components of G and determine the maximum length of a path in G.

1.2.4. (−) Let G be a graph. For $v \in V(G)$ and $e \in E(G)$, describe the adjacency and incidence matrices of $G - v$ and $G - e$ in terms of the corresponding matrices for G.

1.2.5. (−) Let v be a vertex of a connected simple graph G. Prove that v has a neighbor in every component of $G - v$. Conclude that no graph has a cut-vertex of degree 1.

1.2.6. (−) In the graph below (the paw), find all the maximal paths, maximal cliques, and maximal independent sets. Also find all the maximum paths, maximum cliques, and maximum independent sets.



1.2.7. (−) Prove that a bipartite graph has a unique bipartition (except for interchanging the two partite sets) if and only if it is connected.

1.2.8. (−) Determine the values of m and n such that $K_{m,n}$ is Eulerian.

1.2.9. (−) What is the minimum number of trails needed to decompose the Petersen graph? Is there a decomposition into this many trails using only paths?

1.2.10. (−) Prove or disprove:
a) Every Eulerian bipartite graph has an even number of edges.
b) Every Eulerian simple graph with an even number of vertices has an even number of edges.

1.2.11. (−) Prove or disprove: If G is an Eulerian graph with edges e, f that share a vertex, then G has an Eulerian circuit in which e, f appear consecutively.

1.2.12. (−) Convert the proof at 1.2.32 to an procedure for finding an Eulerian circuit in a connected even graph.

• • • • •

1.2.13. *Alternative proofs that every u, v-walk contains a u, v-path (Lemma 1.2.5).*

a) (ordinary induction) Given that every walk of length $l - 1$ contains a path from its first vertex to its last, prove that every walk of length l also satisfies this.

b) (extremality) Given a u, v-walk W, consider a shortest u, v-walk contained in W.

1.2.14. Prove or disprove the following statements about simple graphs. (Comment: "Distinct" does not mean "disjoint".)

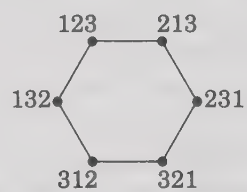
a) The union of the edge sets of distinct u, v-walks must contain a cycle.

b) The union of the edge sets of distinct u, v-paths must contain a cycle.

1.2.15. (!) Let W be a closed walk of length at least 1 that does not contain a cycle. Prove that some edge of W repeats immediately (once in each direction).

1.2.16. Let e be an edge appearing an odd number of times in a closed walk W. Prove that W contains the edges of a cycle through e.

1.2.17. (!) Let G_n be the graph whose vertices are the permutations of $\{1, \dots, n\}$, with two permutations $a_1, \dots, a_n$ and $b_1, \dots, b_n$ adjacent if they differ by interchanging a pair of adjacent entries (G_3 shown below). Prove that G_n is connected.



1.2.18. (!) Let G be the graph whose vertex set is the set of k-tuples with elements in $\{0, 1\}$, with x adjacent to y if x and y differ in exactly two positions. Determine the number of components of G.

1.2.19. Let r and s be natural numbers. Let G be the simple graph with vertex set $v_0, \dots, v_{n-1}$ such that $v_i \leftrightarrow v_j$ if and only if $|j - i| \in \{r, s\}$. Prove that S has exactly k components, where k is the greatest common divisor of $\{n, r, s\}$.

1.2.20. (!) Let v be a cut-vertex of a simple graph G. Prove that $\overline{G} - v$ is connected.

1.2.21. Let G be a self-complementary graph. Prove that G has a cut-vertex if and only if G has a vertex of degree 1. (Akiyama–Harary [1981])

1.2.22. Prove that a graph is connected if and only if for every partition of its vertices into two nonempty sets, there is an edge with endpoints in both sets.

1.2.23. For each statement below, determine whether it is true for every connected simple graph G that is not a complete graph.

a) Every vertex of G belongs to an induced subgraph isomorphic to P_3.

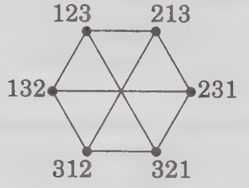
b) Every edge of G belongs to an induced subgraph isomorphic to P_3.

1.2.24. Let G be a simple graph having no isolated vertex and no induced subgraph with exactly two edges. Prove that G is a complete graph.

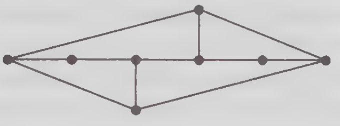
1.2.25. (!) Use ordinary induction on the number of edges to prove that absence of odd cycles is a sufficient condition for a graph to be bipartite.

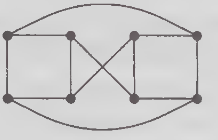
1.2.26. (!) Prove that a graph G is bipartite if and only if every subgraph H of G has an independent set consisting of at least half of $V(H)$.

1.2.27. Let G_n be the graph whose vertices are the permutations of $\{1, \ldots, n\}$, with two permutations $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ adjacent if they differ by switching two entries. Prove that G_n is bipartite (G_3 shown below). (Hint: For each permutation a, count the pairs i, j such that $i < j$ and $a_i > a_j$; these are called **inversions**.)



1.2.28. (!) In each graph below, find a bipartite subgraph with the maximum number of edges. Prove that this is the maximum, and determine whether this is the only bipartite subgraph with this many edges.





1.2.29. (!) Let G be a connected simple graph not having P_4 or C_3 as an induced subgraph. Prove that G is a biclique (complete bipartite graph).

1.2.30. Let G be a simple graph with vertices $v_1, \ldots, v_n$. Let A^k denote the kth power of the adjacency matrix of G under matrix multiplication. Prove that entry i, j of A^k is the number of v_i, v_j-walks of length k in G. Prove that G is bipartite if and only if, for the odd integer r nearest to n, the diagonal entries of A^r are all 0. (Reminder: A walk is an **ordered** list of vertices and edges.)

1.2.31. (!) *Non-inductive proof of Theorem 1.2.23* (see Example 1.2.21).

a) Given $n \le 2^k$, encode the vertices of K_n as distinct binary k-tuples. Use this to construct k bipartite graphs whose union is K_n.

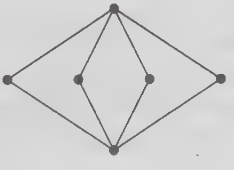
b) Given that K_n is a union of bipartite graphs $G_1, \ldots, G_k$, encode the vertices of K_n as distinct binary k-tuples. Use this to prove that $n \le 2^k$.

1.2.32. The statement below is false. Add a hypothesis to correct it, and prove the corrected statement.

"Every maximal trail in an even graph is an Eulerian circuit."

1.2.33. Use ordinary induction on k or on the number of edges (one by one) to prove that a connected graph with $2k$ odd vertices decomposes into k trails if $k > 0$. Does this remain true without the connectedness hypothesis?

1.2.34. Two Eulerian circuits are *equivalent* if they have the same unordered pairs of consecutive edges, viewed cyclically (the starting point and direction are unimportant). A cycle, for example, has only one equivalence class of Eulerian circuits. How many equivalence classes of Eulerian circuits are there in the graph drawn below?



1.2.35. *Tucker's Algorithm.* Let G be a connected even graph. At each vertex, partition the incident edges into pairs (each edge appears in a pair for each of its endpoints). Starting along a given edge e, form a trail by leaving each vertex along the edge paired with the edge just used to enter it, ending with the edge paired with e. This decomposes G into closed trails. As long as there is more than one trail in the decomposition, find two trails with a common vertex and combine them into a longer trail by changing the pairing at a common vertex. Prove that this procedure works and produces an Eulerian circuit as its final trail. (Tucker [1976])

1.2.36. (+) *Alternative characterization of Eulerian graphs.*

a) Prove that if G is Eulerian and $G' = G - uv$, then G' has an odd number of u, v-trails that visit v only at the end. Prove also that the number of the trails in this list that are not paths is even. (Toida [1973])

b) Let v be a vertex of odd degree in a graph. For each edge e incident to v, let $c(e)$ be the number of cycles containing e. Use $\sum_e c(e)$ to prove that $c(e)$ is even for some e incident to v. (McKee [1984])

c) Use part (a) and part (b) to conclude that a nontrivial connected graph is Eulerian if and only if every edge belongs to an odd number of cycles.

1.2.37. (!) Use extremality to prove that the connection relation is transitive. (Hint: Given a u, v-path P and a v, w-path Q, consider the first vertex of P in Q.)

1.2.38. (!) Prove that every n-vertex graph with at least n edges contains a cycle.

1.2.39. Suppose that every vertex of a loopless graph G has degree at least 3. Prove that G has a cycle of even length. (Hint: Consider a maximal path.) (P. Kwok)

1.2.40. (!) Let P and Q be paths of maximum length in a connected graph G. Prove that P and Q have a common vertex.

1.2.41. Let G be a connected graph with at least three vertices. Prove that G has two vertices x, y such that 1) $G - \{x, y\}$ is connected and 2) x, y are adjacent or have a common neighbor. (Hint: Consider a longest path.) (Chung [1978a])

1.2.42. Let G be a connected simple graph that does not have P_4 or C_4 as an induced subgraph. Prove that G has a vertex adjacent to all other vertices. (Hint: Consider a vertex of maximum degree.) (Wolk [1965])

1.2.43. (+) Use induction on k to prove that every connected simple graph with an even number of edges decomposes into paths of length 2. Does the conclusion remain true if the hypothesis of connectedness is omitted?

1.3. Vertex Degrees and Counting

The degrees of the vertices are fundamental parameters of a graph. We repeat the definition in order to introduce important notation.

1.3.1. Definition. The **degree** of vertex v in a graph G, written $d_G(v)$ or $d(v)$, is the number of edges incident to v, except that each loop at v counts twice. The maximum degree is $\Delta(G)$, the minimum degree is $\delta(G)$, and G is **regular** if $\Delta(G) = \delta(G)$. It is **k-regular** if the common degree is k. The **neighborhood** of v, written $N_G(v)$ or $N(v)$, is the set of vertices adjacent to v.

1.3.2. Definition. The **order** of a graph G, written $n(G)$, is the number of vertices in G. An **n-vertex graph** is a graph of order n. The **size** of a graph G, written $e(G)$, is the number of edges in G. For $n \in \mathbb{N}$, the notation $[n]$ indicates the set $\{1, \dots, n\}$.

Since our graphs are finite, $n(G)$ and $e(G)$ are well-defined nonnegative integers. We also often use "e" by itself to denote an edge. When e denotes a particular edge, it is not followed by the name of a graph in parentheses, so the context indicates the usage. We have used "n-cycle" to denote a cycle with n vertices; this is consistent with "n-vertex graph".

COUNTING AND BIJECTIONS

We begin with counting problems about subgraphs in a graph. The first such problem is to count the edges; we do this using the vertex degrees. The resulting formula is an essential tool of graph theory, sometimes called the "First Theorem of Graph Theory" or the "Handshaking Lemma".

1.3.3. Proposition. (Degree-Sum Formula) If G is a graph, then

$$\textstyle\sum_{v \in V(G)} d(v) = 2e(G).$$

Proof: Summing the degrees counts each edge twice, since each edge has two ends and contributes to the degree at each endpoint. ■

The proof holds even when G has loops, since a loop contributes 2 to the degree of its endpoint. For a loopless graph, the two sides of the formula count the set of pairs (v, e) such that v is an endpoint of e, grouped by vertices or grouped by edges. "Counting two ways" is an elegant technique for proving integer identities (see Exercise 31 and Appendix A).

The degree-sum formula has several immediate corollaries. Corollary 1.3.5 applies in Exercises 9–13 and in many arguments of later chapters.

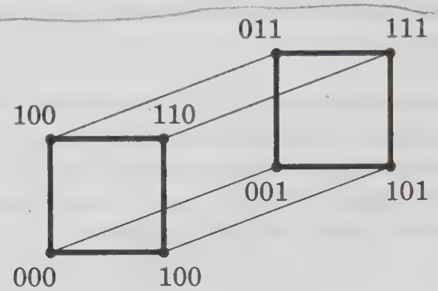
1.3.4. Corollary. In a graph G, the average vertex degree is $\frac{2e(G)}{n(G)}$, and hence $\delta(G) \le \frac{2e(G)}{n(G)} \le \Delta(G)$. ■

1.3.5. Corollary. Every graph has an even number of vertices of odd degree. No graph of odd order is regular with odd degree. ■

1.3.6. Corollary. A k-regular graph with n vertices has $nk/2$ edges. ■

We next introduce an important family of graphs.

1.3.7. Definition. The **k-dimensional cube** or **hypercube** Q_k is the simple graph whose vertices are the k-tuples with entries in $\{0, 1\}$ and whose edges are the pairs of k-tuples that differ in exactly one position. A **j-dimensional subcube** of Q_k is a subgraph of Q_k isomorphic to Q_j.



Above we show Q_3. The hypercube is a natural computer architecture. Processors can communicate directly if they correspond to adjacent vertices in Q_k. The k-tuples that name the vertices serve as addresses for the processors.

1.3.8. Example. *Structure of hypercubes.* The parity of a vertex in Q_k is the parity of the number of 1s in its name, even or odd. Each edge of Q_k has an even vertex and an odd vertex as endpoints. Hence the even vertices form an independent set, as do the odd vertices, and Q_k is bipartite.

Each position in the k-tuples can be specified in two ways, so $n(Q_k) = 2^k$. A neighbor of a vertex is obtained by changing one of the k positions in its name to the other value. Thus Q_k is k-regular. By Corollary 1.3.6, $e(Q_k) = k2^{k-1}$.

The bold edges above show two subgraphs of Q_3 isomorphic to Q_2, formed by keeping the last coordinate fixed at 0 or at 1. We can form a j-dimensional subcube by keeping any $k - j$ coordinates fixed and letting the values in the remaining j coordinates range over all 2^j possible j-tuples. The subgraph induced by such a set of vertices is isomorphic to Q_j. Since there are $\binom{k}{j}$ ways to pick j coordinates to vary and 2^{k-j} ways to specify the values in the fixed coordinates, this specifies $\binom{k}{j}2^{k-j}$ such subcubes. In fact, there are no other j-dimensional subcubes (Exercise 29).

The copies of Q_1 are simply the edges in Q_k. Our formula reduces to $k2^{k-1}$ when $j = 1$, so we have found another counting argument to compute $e(Q_k)$.

When $j = k - 1$, our discussion suggests a recursive description of Q_k. Append 0 to the vertex names in a copy of Q_{k-1}; append 1 in another copy. Add edges joining vertices from the two copies whose first $k - 1$ coordinates are equal. The result is Q_k. The basis of the construction is the 1-vertex graph Q_0. This description leads to inductive proofs for many properties of hypercubes, including $e(Q_k) = k2^{k-1}$ (Exercise 23). ■

A hypercube is a regular bipartite graph. A simple counting argument proves a fundamental observation about such graphs.

1.3.9. Proposition. If $k > 0$, then a k-regular bipartite graph has the same number of vertices in each partite set.

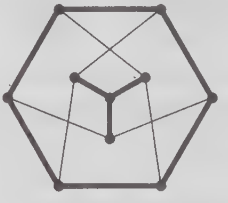
Proof: Let G be an X, Y-bigraph. Counting the edges according to their endpoints in X yields $e(G) = k\,|X|$. Counting them by their endpoints in Y yields $e(G) = k\,|Y|$. Thus $k\,|X| = k\,|Y|$, which yields $|X| = |Y|$ when $k > 0$. ■

Another technique for counting a set is to establish a bijection from it to a set of known size. Our next example uses this approach. Other examples of combinatorial arguments for counting problems appear in Appendix A. Exercises 18–35 involve counting.

?

1.3.10. Example. *The Petersen graph has ten 6-cycles.* Let G be the Petersen graph. Being 3-regular, G has ten claws (copies of $K_{1,3}$). We establish a one-to-one correspondence between the 6-cycles and the claws.

Since G has girth 5, every 6-cycle F is an induced subgraph. Each vertex of F has one neighbor outside F. Since nonadjacent vertices have exactly one common neighbor (Proposition 1.1.38), opposite vertices on F have a common neighbor outside F. Since G is 3-regular, the resulting three vertices outside F are distinct. Thus deleting $V(F)$ leaves a subgraph with three vertices of degree 1 and one vertex of degree 3; it is a claw.



We show that each claw H in G arises exactly once in this way. Let S be the set of vertices with degree 1 in H; S is an independent set. The central vertex of H is already a common neighbor, so the six other edges from S reach distinct vertices. Thus $G - V(H)$ is 2-regular. Since G has girth 5, $G - V(H)$ must be a 6-cycle. This 6-cycle yields H when its vertices are deleted. ■

We present one more counting argument related to a long-standing conjecture. Subgraphs obtained by deleting a single vertex are called **vertex-deleted subgraphs**. These subgraphs need not all be distinct; for example, the n vertex-deleted subgraphs of C_n are all isomorphic to P_{n-1}.

1.3.11.* Proposition. For a simple graph G with vertices $v_1, \ldots, v_n$ and $n \geq 3$,

$$e(G) = \frac{\Sigma e(G - v_i)}{n-2} \qquad \text{and} \qquad d_G(v_j) = \frac{\Sigma e(G - v_i)}{n-2} - e(G - v_j).$$

Proof: An edge e of G appears in $G - v_i$ if and only if v_i is not an endpoint of e. Thus $\sum(G - v_i)$ counts each edge exactly $n - 2$ times.

Once we know $e(G)$, the degree of v_j can be computed as the number of edges lost when deleting v_j to form $G - v_j$. ■

Typically, we are given the vertex-deleted subgraphs as unlabeled graphs; we know only the list of isomorphism classes, not which vertex of $G - v_i$ corresponds to which vertex in G. This can make it very difficult to tell what G is. For example, K_2 and its complement have the same list of vertex-deleted subgraphs. For larger graphs we have the **Reconstruction Conjecture**, formulated in 1942 by Kelly and Ulam.

1.3.12.* Conjecture. (Reconstruction Conjecture) If G is a simple graph with at least three vertices, then G is uniquely determined by the list of (isomorphism classes of) its vertex-deleted subgraphs. ■

The list of vertex-deleted subgraphs of G has $n(G)$ items. Proposition 1.3.11 shows that $e(G)$ and the list of vertex degrees can be reconstructed. The latter implies that regular graphs can be reconstructed (Exercise 37). We can also determine whether G is connected (Exercise 38); using this, disconnected graphs can be reconstructed (Exercise 39). Other sufficient conditions for reconstructibility are known, but the general conjecture remains open.

EXTREMAL PROBLEMS

An **extremal problem** asks for the maximum or minimum value of a function over a class of objects. For example, the maximum number of edges in a simple graph with n vertices is $\binom{n}{2}$.

1.3.13. Proposition. The minimum number of edges in a connected graph with n vertices is $n - 1$.

Proof: By Proposition 1.2.11, every graph with n vertices and k edges has at least $n-k$ components. Hence every n-vertex graph with fewer than $n-1$ edges has at least two components and is disconnected. The contrapositive of this is that every connected n-vertex graph has at least $n - 1$ edges. This lower bound is achieved by the path P_n. ■

1.3.14. Remark. Proving that β is the minimum of $f(G)$ for graphs in a class $\mathbf{G}$ requires showing two things:

1) $f(G) \geq \beta$ for all $G \in \mathbf{G}$.
2) $f(G) = \beta$ for some $G \in \mathbf{G}$.

The proof of the bound must apply to every $G \in \mathbf{G}$. For equality it suffices to obtain an example in $\mathbf{G}$ with the desired value of f.

Changing "$\geq$" to "$\leq$" yields the criteria for a maximum. ■

Next we solve a maximization problem that is not initially phrased as such.

1.3.15. Proposition. If G is a simple n-vertex graph with $\delta(G) \geq (n-1)/2$, then G is connected.

Proof: Choose $u, v \in V(G)$. It suffices to show that u, v have a common neighbor if they are not adjacent. Since G is simple, we have $|N(u)| \geq \delta(G) \geq (n-1)/2$, and similarly for v. When $u \not\leftrightarrow v$, we have $|N(u) \cup N(v)| \leq n-2$, since u and v are not in the union. Using Remark A.13 of Appendix A, we thus compute

$$|N(u) \cap N(v)| = |N(u)| + |N(v)| - |N(u) \cup N(v)| \geq \tfrac{n-1}{2} + \tfrac{n-1}{2} - (n-2) = 1. \blacksquare$$

We say that a result is **best possible** or **sharp** when there is some aspect of it that cannot be strengthened without the statement becoming false. As shown by the next example, this holds for Proposition 1.3.15; when $\delta(G)$ is smaller than $(n(G)-1)/2$, we cannot still conclude that G must be connected.

1.3.16. Example. Let G be the n-vertex graph with components isomorphic to $K_{\lfloor n/2 \rfloor}$ and $K_{\lceil n/2 \rceil}$, where the **floor** $\lfloor x \rfloor$ of x is the largest integer at most x and the **ceiling** $\lceil x \rceil$ of x is the smallest integer at least x. Since $\delta(G) = \lfloor n/2 \rfloor - 1$ and G is disconnected, the inequality in Proposition 1.3.15 is sharp.

We use the floor and ceiling functions here in order to describe a single family of graphs providing an example for each n. ■

By providing a family of examples to show that the bound is best possible, we have solved an extremal problem. Together, Proposition 1.3.15 and Example 1.3.16 prove "The minimum value of $\delta(G)$ that forces an n-vertex simple graph G to be connected is $\lfloor n/2 \rfloor$," or "The maximum value of $\delta(G)$ among disconnected n-vertex simple graphs is $\lfloor n/2 \rfloor - 1$."

We introduce compact notation to describe the graph of Example 1.3.16.

1.3.17. Definition. The graph obtained by taking the union of graphs G and H with disjoint vertex sets is the **disjoint union** or **sum**, written $G + H$. In general, mG is the graph consisting of m pairwise disjoint copies of G.

1.3.18. Example. If G and H are connected, then $G+H$ has components G and H, so the graph in Example 1.3.16 is $K_{\lfloor n/2 \rfloor} + K_{\lceil n/2 \rceil}$. This notation is convenient when we have not named the vertices. Note that $K_m + K_n = \overline{K}_{m,n}$.

The graph mK_2 consists of m pairwise disjoint edges. ■

In graph theory, we use "extremal problem" for finding an optimum over a class of graphs. When seeking extremes in a single graph, such as the maximum size of an independent set, or maximum size of a bipartite subgraph, we have a different problem for each graph. To distinguish these from the earlier type of problem, we call them **optimization problems**.

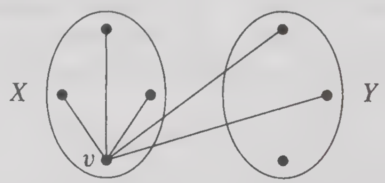
Since an optimization problem has an instance for each graph, we usually can't list all solutions. We may seek a solution procedure or bounds on the answer in terms of other aspects of the input graph. In this light we consider the problem of finding a large bipartite subgraph. It allows us to introduce the technique of constructive or "algorithmic" proof. (An **algorithm** is a procedure for performing some task.)

One way to prove that something exists is to build it. Such proofs can be viewed as algorithms. To complete an algorithmic proof, we must prove that the algorithm terminates and yields the desired result. This may involve induction,contradiction, finiteness, etc. We prove that every graph has a large bipartitesubgraph by providing an algorithm to find one. Exercises 45–49 arerelated to finding large bipartite subgraphs.

1.3.19. Theorem. Every loopless graph G has a bipartite subgraph with at least $e(G)/2$ edges.

Proof: We start with any partition of $V(G)$ into two sets X, Y. Using the edges having one endpoint in each set yields a bipartite subgraph H with bipartition X, Y. If H contains fewer than half the edges of G incident to a vertex v, then v has more edges to vertices in its own class than in the other class, as illustrated below. Moving v to the other class gains more edges of G than it loses.

We move one vertex in this way as long as the current bipartite subgraph captures less than half of the edges at some vertex. Each such switch increases the size of the subgraph, so the process must terminate. When it terminates, we have $d_H(v) \geq d_G(v)/2$ for every $v \in V(G)$. Summing this and applying the degree-sum formula yields $e(H) \geq e(G)/2$. ■

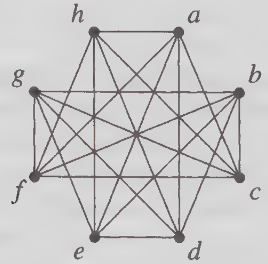


Algorithmic proofs often correspond to proofs by induction or extremality. Such proofs are shorter and may be easier to find, so we may seek such a proof and later convert it to an algorithm. For example, here is the proof of Theorem 1.3.19 in the language of extremality and contradiction; in effect, the extremal choice of H goes directly to the end of the algorithm:

> Let H be the bipartite subgraph of G that has the most edges. If $d_H(v) \geq d_G(v)/2$ for all $v \in V(G)$, then the degree-sum formula yields $e(H) \geq e(G)/2$. Otherwise, $d_H(v) < d_G(v)/2$ for some $v \in V(G)$, and then switching v in the bipartition contradicts the choice of H.

1.3.20. Example. *Local maximum.* The algorithm in Theorem 1.3.19 need not produce a bipartite subgraph with the most edges, merely one with at least half the edges. The graph below is 5-regular with 8 vertices and hence has 20 edges. The bipartition $X = \{a, b, c, d\}$ and $Y = \{e, f, g, h\}$ yields a 3-regular bipartite subgraph with 12 edges. The algorithm terminates here; switching one vertex would pick up two edges but lose three.

Nevertheless, the bipartition $X = \{a, b, g, h\}$ and $Y = \{c, d, e, f\}$ yields a 4-regular bipartite subgraph with 16 edges. An algorithm seeking the maximum by local changes may get stuck in a local maximum. ■



1.3.21. Remark. In a graph G, the (global) maximum number of edges in a bipartite subgraph is $e(G)$ minus the minimum number of edges needed to obtain at least one edge from every odd cycle. ■

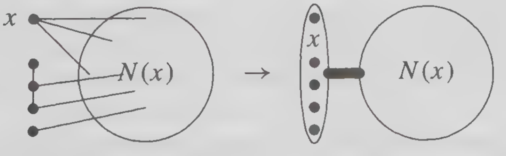
Our next extremal problem doesn't start with bipartite graphs, but it winds up there. In politics and warfare, seldom do two enemies have a common enemy; usually two of the three combine against the third. Given n factions, how many pairs of enemies can there be if no two enemies have a common enemy?

In the language of graphs, we are asking for the maximum number of edges in a simple n-vertex graph with no triangle. Bipartite graphs have no triangles, but also many non-bipartite graphs (such as the Petersen graph) have no triangles. Using extremality (by choosing a vertex of maximum degree), we will prove that the maximum is indeed achieved by a complete bipartite graph.

1.3.22. Definition. A graph G is H**-free** if G has no induced subgraph isomorphic to H.

1.3.23. Theorem. (Mantel [1907]) The maximum number of edges in an n-vertex triangle-free simple graph is $\lfloor n^2/4 \rfloor$.

Proof: Let G be an n-vertex triangle-free simple graph. Let x be a vertex of maximum degree, with $k = d(x)$. Since G has no triangles, there are no edges among neighbors of x. Hence summing the degrees of x and its nonneighbors counts at least one endpoint of every edge: $\sum_{v \notin N(x)} d(v) \geq e(G)$. We sum over $n - k$ vertices, each having degree at most k, so $e(G) \leq (n - k)k$.



Since $(n - k)k$ counts the edges in $K_{n-k,k}$, we have now proved that $e(G)$ is bounded by the size of some biclique with n vertices. Moving a vertex of $K_{n,k,k}$

from the set of size k to the set of size $n-k$ gains $k-1$ edges and loses $n-k$ edges. The net gain is $2k-1-n$, which is positive for $2k > n+1$ and negative for $2k < n+1$. Thus $e(K_{n-k,k})$ is maximized when k is $\lceil n/2 \rceil$ or $\lfloor n/2 \rfloor$. The product is then $n^2/4$ for even n and $(n^2-1)/4$ for odd n. Thus $e(G) \le \lfloor n^2/4 \rfloor$.

To prove that the bound is best possible, we exhibit a triangle-free graph with $\lfloor n^2/4 \rfloor$ edges: $K_{\lfloor n/2 \rfloor, \lceil n/2 \rceil}$. ■

Although $(n-k)k$ can be maximized over k using calculus, the discrete approach is preferable in some ways. It directly restricts k to be an integer and generalizes easily to more variables. The switching idea used is that of Theorem 1.3.19; here we have used it to find the largest bipartite subgraph of K_n. In Theorem 5.2.9 we generalize Theorem 1.3.23 to K_{r+1}-free graphs.

Mantel's result leads us to another reason for phrasing inductive proofs in the format that we have used. The reason is safety.

1.3.24. Example. *A failed proof.* Let us try to prove Theorem 1.3.23 by induction on n. Basis step: $n \le 2$. Here the complete graph K_n has the most edges and has no triangles.

Induction step: $n > 2$. We try "Suppose that the claim is true when $n = k$, so $K_{\lfloor k/2 \rfloor, \lceil k/2 \rceil}$ is the largest triangle-free graph with k vertices. We add a new vertex x to form a triangle-free graph with $k+1$ vertices. Making x adjacent to vertices from both partite sets would create a triangle. Hence we add the most edges by making x adjacent to all the vertices in the larger partite set of $K_{\lfloor k/2 \rfloor, \lceil k/2 \rceil}$. Doing so creates $K_{\lfloor (k+1)/2 \rfloor, \lceil (k+1)/2 \rceil}$. This completes the proof."

This argument is wrong, because we did not consider all triangle-free graphs with $k+1$ vertices. We considered only those containing the extremal k-vertex graph as an induced subgraph. This graph does appear in the extremal graph with $k+1$ vertices, but we cannot use that fact before proving it. It remains possible that the largest example with $k+1$ vertices arises by adding a new vertex of high degree to a non-maximal example with k vertices.

Exercise 51 develops a correct proof by induction on n. ■

The error in Example 1.3.24 was that our induction step did not consider all instances of the statement for the new larger value of the parameter. We call this error the **induction trap**. If the induction step grows an instance with the new value of the parameter from a smaller instance, then we must prove that all instances with the new value have been considered.

When there is only one instance for each value of the induction parameter (as in summation formulas), this does not cause trouble. With more than one instance, it is safer and simpler to start with an arbitrary instance for the larger parameter value. This explicitly considers each instance G for the larger value, so we don't need to prove that we have generated them all.

However, when we obtain from G a smaller instance, we must confirm that the induction hypothesis applies to it. For example, in the inductive proof of the characterization of Eulerian circuits (Theorem 1.2.26), we must apply the induction hypothesis to each component of the graph obtained by deleting the edges of a cycle, not to the entire graph at once.

1.3.25. Remark. *A template for induction.* Often the statement we want to prove by induction on n is an implication: $A(n) \Rightarrow B(n)$. We must prove that every instance G satisfying $A(n)$ also satisfies $B(n)$. Our induction step follows a typical format. From G we obtain some (smaller) G'. If we show that G' satisfies $A(n-1)$ (for ordinary induction), then the induction hypothesis implies that G' satisfies $B(n-1)$. Now we use the information that G' satisfies $B(n-1)$ to prove that G satisfies $B(n)$.

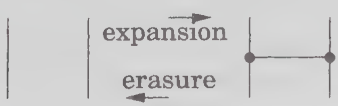
$$\begin{array}{ccc} G \text{ satisfies } A(n) & & G \text{ satisfies } B(n) \\ \Downarrow & & \Uparrow \\ G' \text{ satisfies } A(n-1) & \Rightarrow & G' \text{ satisfies } B(n-1) \end{array}$$

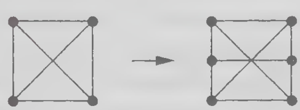
Here the central implication is the statement of the induction hypothesis, and the others are the work we must do. Our induction proofs have followed this format. ■

1.3.26.* Example. *The induction trap.* The induction trap can lead to a *false* conclusion. Let us try to prove by induction on the number of vertices that every 3-regular connected simple graph has no cut-edge.

By the degree-sum formula, every regular graph with odd degree has even order, so we consider graphs with $2m$ vertices. The smallest 3-regular simple graph, K_4, is connected and has no cut-edge; this proves the basis step with $m = 2$. Now consider the induction step.

Given a simple 3-regular graph G with $2k$ vertices, we can obtain a simple 3-regular graph G' with $2(k+1)$ vertices (the next larger possible order) by "expansion": take two edges of G, replace them by paths of length 2 through new vertices, and add an edge joining the two new vertices. As illustrated below, $K_{3,3}$ arises from K_4 by one expansion on two disjoint edges.

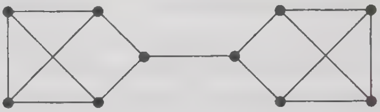




If G is connected, then the expanded graph G' is also connected: a path between old vertices that traversed a replaced edge has merely lengthened, and a path to a new vertex in G' is obtained from a path in G to a neighbor.

If G has no cut-edge, then every edge lies on a cycle (Theorem 1.2.14). These cycles remain in G' (those using replaced edges become longer). The edge joining the two new vertices in G' also lies on a cycle using a path in G between the edges that were replaced. Theorem 1.2.14 now implies that G' has no cut-edge.

We have proved that if G is connected and has no cut-edge, then the same holds for G'. We might think we have proved by induction on m that every 3-regular simple connected graph with $2m$ vertices has no cut-edge, but the graph below is a counterexample. The proof fails because we cannot build every 3-regular simple connected graph from K_4 by expansions. We cannot even obtain all those without cut-edges, as shown in Exercise 66. ■



Appendix A presents another example of the induction trap.

GRAPHIC SEQUENCES

Next we consider all the vertex degrees together.

1.3.27. Definition. The **degree sequence** of a graph is the list of vertex degrees, *usually* written in nonincreasing order, as $d_1 \geq \cdots \geq d_n$.

Every graph has a degree sequence, but which sequences occur? That is, given nonnegative integers $d_1, \ldots, d_n$, is there a graph with these as the vertex degrees? The degree-sum formula implies that $\sum d_i$ must be even. When we allow loops and multiple edges, TONCAS.

1.3.28. Proposition. The nonnegative integers $d_1, \ldots, d_n$ are the vertex degrees of some graph if and only if $\sum d_i$ is even.

Proof: *Necessity.* When some graph G has these numbers as its vertex degrees, the degree-sum formula implies that $\sum d_i = 2e(G)$, which is even.

Sufficiency. Suppose that $\sum d_i$ is even. We construct a graph with vertex set $v_1, \ldots, v_n$ and $d(v_i) = d_i$ for all i. Since $\sum d_i$ is even, the number of odd values is even. First form an arbitrary pairing of the vertices in $\{v_i: d_i \text{ is odd}\}$. For each resulting pair, form an edge having these two vertices as its endpoints. The remaining degree needed at each vertex is even and nonnegative; satisfy this for each i by placing $\lfloor d_i/2 \rfloor$ loops at v_i. ■

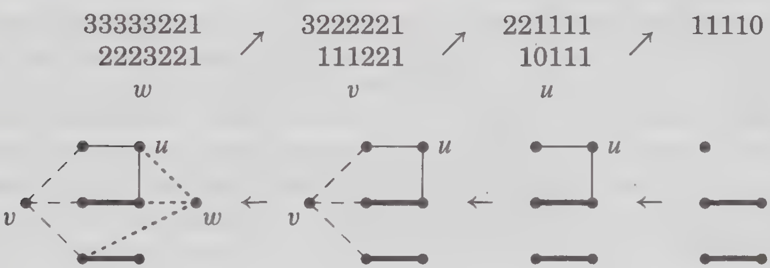
This proof is constructive; we could also use induction (Exercise 56). The construction is easy with loops available. Without them, $(2, 0, 0)$ is not realizable and the condition is not sufficient. Exercise 63 characterizes the degree sequences of loopless graphs. We next characterize degree sequences of simple graphs by a recursive condition that readily yields an algorithm. Many other characterizations are known; Sierksma–Hoogeveen [1991] lists seven.

1.3.29. Definition. A **graphic sequence** is a list of nonnegative numbers that is the degree sequence of some simple graph. A simple graph with degree sequence d "realizes" d.

1.3.30. Example. *A recursive condition.* The lists $2, 2, 1, 1$ and $1, 0, 1$ are graphic. The graph $K_2 + K_1$ realizes $1, 0, 1$. Adding a new vertex adjacent to vertices of degrees 1 and 0 yields a graph with degree sequence $2, 2, 1, 1$, as shown below. Conversely, if a graph realizing $2, 2, 1, 1$ has a vertex w with neighbors of degrees 2 and 1, then deleting w yields a graph with degrees $1, 0, 1$.

Similarly, to test 33333221, we seek a realization with a vertex w of degree 3 having three neighbors of degree 3. This exists if and only if 2223221 is graphic. We reorder this and test 3222221. We continue deleting and reordering until we can tell whether the remaining list is realizable. If it is, then we insert vertices with the desired neighbors to work back to a realization of the original list. The realization is not unique.

The next theorem implies that this recursive test works. ■



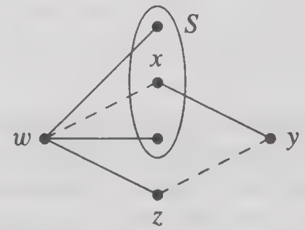
1.3.31. Theorem. (Havel [1955], Hakimi [1962]) For $n > 1$, an integer list d of size n is graphic if and only if d' is graphic, where d' is obtained from d by deleting its largest element Δ and subtracting 1 from its Δ next largest elements. The only 1-element graphic sequence is $d_1 = 0$.

Proof: For $n = 1$, the statement is trivial. For $n > 1$, we first prove that the condition is sufficient. Given d with $d_1 \geq \cdots \geq d_n$ and a simple graph G' with degree sequence d', we add a new vertex adjacent to vertices in G' with degrees $d_2 - 1, \ldots, d_{\Delta+1} - 1$. These d_i are the Δ largest elements of d after (one copy of) Δ itself, but $d_2 - 1, \ldots, d_{\Delta+1} - 1$ need not be the Δ largest numbers in d'.

To prove necessity, we begin with a simple graph G realizing d and produce a simple graph G' realizing d'. Let w be a vertex of degree Δ in G. Let S be a set of Δ vertices in G having the "desired degrees" $d_2, \ldots, d_{\Delta+1}$. If $N(w) = S$, then we delete w to obtain G'.

Otherwise, some vertex of S is missing from $N(w)$. In this case, we modify G to increase $|N(w) \cap S|$ without changing any vertex degree. Since $|N(w) \cap S|$ can increase at most Δ times, repeating this converts G into another graph G^* that realizes d and has S as the neighborhood of w. From G^* we then delete w to obtain the desired graph G' realizing d'.

To find the modification when $N(w) \neq S$, we choose $x \in S$ and $z \notin S$ so that $w \leftrightarrow z$ and $w \not\leftrightarrow x$. We want to add wx and delete wz, but we must preserve vertex degrees. Since $d(x) \geq d(z)$ and already w is a neighbor of z but not x, there must be a vertex y adjacent to x but not to z. Now we delete $\{wz, xy\}$ and add $\{wx, yz\}$ to increase $|N(w) \cap S|$. ■

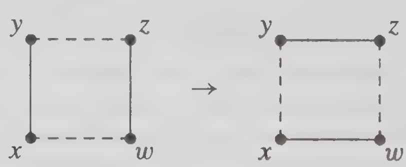


Theorem 1.3.31 tests a list of n numbers by testing a list of $n-1$ numbers; it yields a recursive algorithm to test whether d is graphic. The necessary condition "$\sum d_i$ even" holds implicitly: $\sum d_i' = (\sum d_i) - 2\Delta$ implies that $\sum d_i'$ and $\sum d_i$ have the same parity.

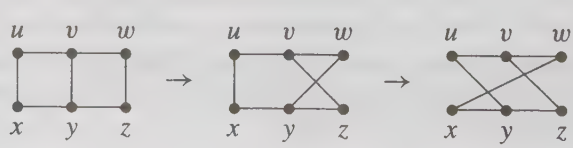
An algorithmic proof using "local change" pushes an object toward a desired condition. This can be phrased as proof by induction, where the induction parameter is the "distance" from the desired condition. In the proof of Theorem 1.3.31, this distance is the number of vertices in S that are missing from $N(w)$.

We used edge switches to transform an arbitrary graph with degree sequence d into a graph satisfying the desired condition. Next we will show that every simple graph with degree sequence d can be transformed by such switches into every other.

1.3.32. Definition. A **2-switch** is the replacement of a pair of edges xy and zw in a simple graph by the edges yz and wx, given that yz and wx did not appear in the graph originally.



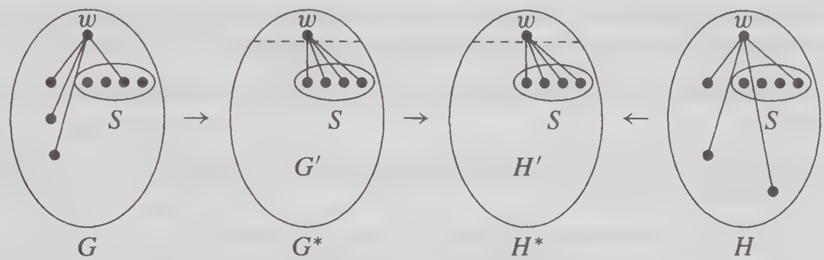
The dashed lines above indicate nonadjacent pairs. If $y \leftrightarrow z$ or $w \leftrightarrow x$, then the 2-switch cannot be performed, because the resulting graph would not be simple. A 2-switch preserves all vertex degrees. If some 2-switch turns H into H^*, then a 2-switch on the same four vertices turns H^* into H. Below we illustrate two successive 2-switches.



1.3.33.* Theorem. (Berge [1973, p153–154]) If G and H are two simple graphs with vertex set V, then $d_G(v) = d_H(v)$ for every $v \in V$ if and only if there is a sequence of 2-switches that transforms G into H.

Proof: Every 2-switch preserves vertex degrees, so the condition is sufficient. Conversely, when $d_G(v) = d_H(v)$ for all $v \in V$, we obtain an appropriate sequence of 2-switches by induction on the number of vertices, n. If $n \le 3$, then for each $d_1, \ldots, d_n$ there is at most one simple graph with $d(v_i) = d_i$. Hence we can use $n = 3$ as the basis step.

Consider $n \ge 4$, and let w be a vertex of maximum degree, Δ. Let $S = \{v_1, \ldots, v_\Delta\}$ be a fixed set of vertices with the Δ highest degrees other than w. As in the proof of Theorem 1.3.31, some sequence of 2-switches transforms G to a graph G^* such that $N_{G^*}(w) = S$, and some such sequence transforms H to a graph H^* such that $N_{H^*}(w) = S$.



Since $N_{G^*}(w) = N_{H^*}(w)$, deleting w leaves simple graphs $G' = G^* - w$ and $H' = H^* - w$ with $d_{G'}(v) = d_{H'}(v)$ for every vertex v. By the induction hypothesis, some sequence of 2-switches transforms G' to H'. Since these do not involve w, and w has the same neighbors in G^* and H^*, applying this sequence transforms G^* to H^*. Hence we can transform G to H by transforming G to G^*, then G^* to H^*, then (in reverse order) the transformation of H to H^*. ■

We could also phrase this using induction on the number of edges appearing in exactly one of G and H, which is 0 if and only if they are already the same. In this approach, it suffices to find a 2-switch in G that makes it closer to H or a 2-switch in H that makes it closer to G.

EXERCISES

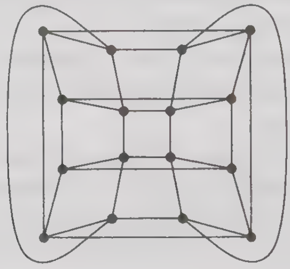
A statement with a parameter must be proved for all values of the parameter; it cannot be proved by giving examples. Counting a set includes providing proof.

1.3.1. (−) Prove or disprove: If u and v are the only vertices of odd degree in a graph G, then G contains a u,v-path.

1.3.2. (−) In a class with nine students, each student sends valentine cards to three others. Determine whether it is possible that each student receives cards from the same three students to whom he or she sent cards.

1.3.3. (−) Let u and v be adjacent vertices in a simple graph G. Prove that uv belongs to at least $d(u) + d(v) - n(G)$ triangles in G.

1.3.4. (−) Prove that the graph below is isomorphic to Q_4.



1.3.5. (−) Count the copies of P_3 and C_4 in Q_k.

1.3.6. (−) Given graphs G and H, determine the number of components and maximum degree of $G + H$ in terms of the those parameters for G and H.

1.3.7. (−) Determine the maximum number of edges in a bipartite subgraph of P_n, of C_n, and of K_n.

1.3.8. (−) Which of the following are graphic sequences? Provide a construction or a proof of impossibility for each.

a) (5,5,4,3,2,2,2,1), c) (5,5,5,3,2,2,1,1),
b) (5,5,4,4,2,2,1,1), d) (5,5,5,4,2,1,1,1).

• • • • •

1.3.9. In a league with two divisions of 13 teams each, determine whether it is possible to schedule a season with each team playing nine games against teams within its division and four games against teams in the other division.

1.3.10. Let l, m, n be nonnegative integers with $l + m = n$. Find necessary and sufficient conditions on l, m, n such that there exists a connected simple n-vertex graph with l vertices of even degree and m vertices of odd degree.

1.3.11. Let W be a closed walk in a graph G. Let H be the subgraph of G consisting of edges used an odd number of times in W. Prove that $d_H(v)$ is even for every $v \in V(G)$.

1.3.12. (!) Prove that an even graph has no cut-edge. For each $k \geq 1$, construct a $2k+1$-regular simple graph having a cut-edge.

1.3.13. (+) A **mountain range** is a polygonal curve from $(a, 0)$ to $(b, 0)$ in the upper half-plane. Hikers A and B begin at $(a, 0)$ and $(b, 0)$, respectively. Prove that A and B can meet by traveling on the mountain range in such a way that at all times their heights above the horizontal axis are the same. (Hint: Define a graph to model the movements, and use Corollary 1.3.5.) (Communicated by D.G. Hoffman.)

1.3.14. Prove that every simple graph with at least two vertices has two vertices of equal degree. Is the conclusion true for loopless graphs?

1.3.15. For each $k \geq 3$, determine the smallest n such that
a) there is a simple k-regular graph with n vertices.
b) there exist nonisomorphic simple k-regular graphs with n vertices.

1.3.16. (+) For $k \geq 2$ and $g \geq 2$, prove that there exists an k-regular graph with girth g. (Hint: To construct such a graph inductively, make use of an $k-1$-regular graph H with girth g and a graph with girth $\lceil g/2 \rceil$ that is $n(H)$-regular. Comment: Such a graph with minimum order is a (k, g)-**cage**.) (Erdős–Sachs [1963])

1.3.17. (!) Let G be a graph with at least two vertices. Prove or disprove:
a) Deleting a vertex of degree $\Delta(G)$ cannot increase the average degree.
b) Deleting a vertex of degree $\delta(G)$ cannot reduce the average degree.

1.3.18. (!) For $k \geq 2$, prove that a k-regular bipartite graph has no cut-edge.

1.3.19. Let G be a claw-free graph. Prove that if $\Delta(G) \geq 5$, then G has a 4-cycle. For all $n \in \mathbb{N}$, construct a 4-regular claw-free graph of order at least n that has no 4-cycle.

1.3.20. (!) Count the cycles of length n in K_n and the cycles of length $2n$ in $K_{n,n}$.

1.3.21. Count the 6-cycles in $K_{m,n}$.

1.3.22. (!) Let G be a nonbipartite graph with n vertices and minimum degree k. Let l be the minimum length of an odd cycle in G.

a) Let C be a cycle of length l in G. Prove that every vertex not in $V(C)$ has at most two neighbors in $V(C)$.

b) By counting the edges joining $V(C)$ and $G - V(C)$ in two ways, prove that $n \geq kl/2$ (and thus $l \leq 2n/k$). (Campbell–Staton [1991])

c) When k is even, prove that the inequality of part (b) is best possible. (Hint: form a graph having $k/2$ pairwise disjoint l-cycles.)

1.3.23. Use the recursive description of Q_k (Example 1.3.8) to prove that $e(Q_k) = k2^{k-1}$.

1.3.24. Prove that $K_{2,3}$ is not contained in any hypercube Q_k.

1.3.25. (!) Prove that every cycle of length $2r$ in a hypercube is contained in a subcube of dimension at most r. Can a cycle of length $2r$ be contained in a subcube of dimension less than r?

1.3.26. (!) Count the 6-cycles in Q_3. Prove that every 6-cycle in Q_k lies in exactly one 3-dimensional subcube. Use this to count the 6-cycles in Q_k for $k \geq 3$.

1.3.27. Given $k \in \mathbb{N}$, let G be the subgraph of Q_{2k+1} induced by the vertices in which the number of ones and zeros differs by 1. Prove that G is regular, and compute $n(G)$, $e(G)$, and the girth of G.

1.3.28. Let V be the set of binary k-tuples. Define a simple graph Q'_k with vertex set V by putting $u \leftrightarrow v$ if and only if u and v *agree* in exactly one coordinate. Prove that Q'_k is isomorphic to the hypercube Q_k if and only if k is even. (D.G. Hoffman)

1.3.29. (*+) *Automorphisms of the k-dimensional cube Q_k.*

a) Prove that every copy of Q_j in Q_k is a subgraph induced by a set of 2^j vertices having specified values on a fixed set of $k - j$ coordinates. (Hint: Prove that a copy of Q_j must have two vertices differing in j coordinates.)

b) Use part (a) to count the automorphisms of Q_k.

1.3.30. Prove that every edge in the Petersen graph belongs to exactly four 5-cycles, and use this to show that the Petersen graph has exactly twelve 5-cycles. (Hint: For the first part, extend the edge to a copy of P_4 and apply Proposition 1.1.38.)

1.3.31. (!) Use complete graphs and counting arguments (not algebra!) to prove that

a) $\binom{n}{2} = \binom{k}{2} + k(n-k) + \binom{n-k}{2}$ for $0 \le k \le n$.

b) If $\sum n_i = n$, then $\sum \binom{n_i}{2} \le \binom{n}{2}$.

1.3.32. (!) Prove that the number of simple even graphs with vertex set $[n]$ is $2^{\binom{n-1}{2}}$. (Hint: Establish a bijection to the set of all simple graphs with vertex set $[n-1]$.)

1.3.33. (+) Let G be a triangle-free simple n-vertex graph such that every pair of nonadjacent vertices has exactly two common neighbors.

a) Prove that $n(G) = 1 + \binom{d(x)}{2}$, where $x \in V(G)$. Conclude that G is regular.

b) When $k = 5$, prove that deleting any one vertex and its neighbors from G leaves the Petersen graph. (Comment: When $k = 5$, the graph G is in fact the graph obtained from Q_4 by adding edges joining complementary vertices.)

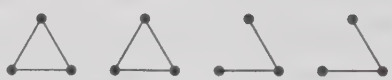
1.3.34. (+) Let G be a kite-free simple n-vertex graph such that every pair of nonadjacent vertices has exactly two common neighbors. Prove that G is regular. (Galvin)

1.3.35. (+) Let n and k be integers such that $1 < k < n - 1$. Let G be a simple n-vertex graph such that every k-vertex induced subgraph of G has m edges.

a) Let G' be an induced subgraph of G with l vertices, where $l > k$. Prove that $e(G') = m\binom{l}{k}/\binom{l-2}{k-2}$.

b) Use part (a) to prove that $G \in \{K_n, \overline{K}_n\}$. (Hint: Use part (a) to prove that the number of edges with endpoints u, v is independent of the choice of u and v.)

1.3.36. Let G be a 4-vertex graph whose list of subgraphs obtained by deleting one vertex appears below. Determine G.



1.3.37. Let H be a graph formed by deleting a vertex from a loopless regular graph G with $n(G) \ge 3$. Describe (and justify) a method for obtaining G from H.

1.3.38. Let G be a graph with at least 3 vertices. Prove that G is connected if and only if at least two of the subgraphs obtained by deleting one vertex of G are connected. (Hint: Use Proposition 1.2.29.)

1.3.39. (*+) Prove that every disconnected graph G with at least three vertices is reconstructible. (Hint: Having used Exercise 1.3.38 to determine that G is disconnected, use $G_1, \dots, G_n$ to find a component M of G that occurs the most times among the components with the maximum number of vertices, use Proposition 1.2.29 to choose v so that $L = M - v$ is connected, and reconstruct G by finding some $G - v_i$ in which a copy of M became a copy of L.)

1.3.40. (!) Let G be an n-vertex simple graph, where $n \ge 2$. Determine the maximum possible number of edges in G under each of the following conditions.

a) G has an independent set of size a.

b) G has exactly k components.

c) G is disconnected.

1.3.41. (!) Prove or disprove: If G is an n-vertex simple graph with maximum degree $\lceil n/2 \rceil$ and minimum degree $\lfloor n/2 \rfloor - 1$, then G is connected.

1.3.42. Let S be a set of vertices in a k-regular graph G such that no two vertices in S are adjacent or have a common neighbor. Use the pigeonhole principle to prove that $|S| \le \lfloor n(G)/(k+1) \rfloor$. Show that the bound is best possible for the cube Q_3. (Comment: The bound is not best possible for Q_4.)

1.3.43. (+) Let G be a simple graph with no isolated vertices, and let $a = 2e(G)/n(G)$ be the average degree in G. Let $t(v)$ denote the average of the degrees of the neighbors of v. Prove that $t(v) \ge a$ for some $v \in V(G)$. Construct an infinite family of connected graphs such that $t(v) > a$ for every vertex v. (Hint: For the first part, compute the average of $t(v)$, using that $x/y + y/x \ge 2$ when $x, y > 0$.) (Ajtai–Komlós–Szemerédi [1980])

1.3.44. (!) Let G be a loopless graph with average vertex degree $a = 2e(G)/n(G)$.

a) Prove that $G - x$ has average degree at least a if and only if $d(x) \le a/2$.

b) Use part (a) to give an algorithmic proof that if $a > 0$, then G has a subgraph with minimum degree greater than $a/2$.

c) Show that there is no constant c greater than 1/2 such that G must have a subgraph with minimum degree greater than ca; this proves that the bound in part (b) is best possible. (Hint: Use $K_{1,n-1}$.)

1.3.45. Determine the maximum number of edges in a bipartite subgraph of the Petersen graph.

1.3.46. Prove or disprove: Whenever the algorithm of Theorem 1.3.19 is applied to a bipartite graph, it finds the bipartite subgraph with the most edges (the full graph).

1.3.47. Use induction on $n(G)$ to prove that every nontrivial loopless graph G has a bipartite subgraph H such that H has *more* than $e(G)/2$ edges.

1.3.48. Construct graphs $G_1, G_2, \ldots$, with G_n having $2n$ vertices, such that $\lim_{n\to\infty} f_n = 1/2$, where f_n is the fraction of $E(G_n)$ belonging to the largest bipartite subgraph of G_n.

1.3.49. For each $k \in \mathbb{N}$ and each loopless graph G, prove that G has a k-partite subgraph H (Definition 1.1.12) such that $e(H) \ge (1 - 1/k)e(G)$.

1.3.50. (+) For $n \ge 3$, determine the minimum number of edges in a connected n-vertex graph in which every edge belongs to a triangle. (Erdős [1988])

1.3.51. (+) Let G be a simple n-vertex graph, where $n > 3$.

a) Use Proposition 1.3.11 to prove that if G has more than $n^2/4$ edges, then G has a vertex whose deletion leaves a graph with more than $(n-1)^2/4$ edges. (Hint: In every graph, the number of edges is an integer.)

b) Use part (a) to prove by induction that G contains a triangle if $e(G) > n^2/4$.

1.3.52. Prove that every n-vertex triangle-free simple graph with the maximum number of edges is isomorphic to $K_{\lfloor n/2 \rfloor, \lceil n/2 \rceil}$. (Hint: Strengthen the proof of Theorem 1.3.23.)

1.3.53. (!) Each game of *bridge* involves two teams of two partners each. Consider a club in which four players cannot play a game if two of them have previously been partners that night. Suppose that 15 members arrive, but one decides to study graph theory. The other 14 people play until each has been a partner with four others. Next they succeed in playing six more games (12 partnerships), but after that they cannot find four players containing no pair of previous partners. Prove that if they can convince the graph theorist to play, then at least one more game can be played. (Adapted from Bondy–Murty [1976, p111]).

1.3.54. (+) Let G be a simple graph with n vertices. Let $t(G)$ be the total number of triangles in G and $\overline{G}$ together.

a) Prove that $t(G) = \binom{n}{3} - (n-2)e(G) + \sum_{v\in V(G)} \binom{d(v)}{2}$ triangles. (Hint: Consider the contribution made to each side by each triple of vertices.

b) Prove that $t(G) \geq n(n-1)(n-5)/24$. (Hint: Use a lower bound on $\sum_{v\in V(G)} \binom{d(v)}{2}$ in terms of average degree.)

c) When $n-1$ is divisible by 4, construct a graph achieving equality in part (b). (Goodman [1959])

1.3.55. (+) *Maximum size with no induced P_4.*

a) Let G be the complement of a disconnected simple graph. Prove that $e(G) \leq \Delta(G)^2$, with equality only for $K_{\Delta(G),\Delta(G)}$.

b) Let G be a simple connected P_4-free graph with maximum degree k. Prove that $e(G) \leq k^2$. (Seinsche [1974], Chung–West [1993])

1.3.56. Use induction (on n or on $\sum d_i$) to prove that if $d_1, \ldots, d_n$ are nonnegative integers and $\sum d_i$ is even, then there is an n-vertex graph with vertex degrees $d_1, \ldots, d_n$. (Comment: This requests an alternative proof of Proposition 1.3.28.)

1.3.57. (!) Let n be a positive integer. Let d be a list of n nonnegative integers with even sum whose largest entry is less than n and differs from the smallest entry by at most 1. Prove that d is graphic. (Hint: Use the Havel–Hakimi Theorem. Example: 443333 is such a list, as is 33333322.)

1.3.58. *Generalization of Havel–Hakimi Theorem.* Given a nonincreasing list d of nonnegative integers, let d' be obtained by deleting d_k and subtracting 1 from the k largest elements remaining in the list. Prove that d is graphic if and only if d' is graphic. (Hint: Mimic the proof of Theorem 1.3.31.) (Wang–Kleitman [1973])

1.3.59. Define $d = (d_1, \ldots, d_{2k})$ by $d_{2i} = d_{2i-1} = i$ for $1 \leq i \leq k$. Prove that d is graphic. (Hint: Do not use the Havel–Hakimi Theorem.)

1.3.60. (+) Let d be a list of integers consisting of k copies of a and $n-k$ copies of b, with $a \geq b \geq 0$. Determine necessary and sufficient conditions for d to be graphic.

1.3.61. (!) Suppose that $G \cong \overline{G}$ and that $n(G) \equiv 1 \bmod 4$. Prove that G has at least one vertex of degree $(n(G)-1)/2$.

1.3.62. Suppose that n is congruent to 0 or 1 modulo 4. Construct an n-vertex simple graph G with $\frac{1}{2}\binom{n}{2}$ edges such that $\Delta(G) - \delta(G) \leq 1$.

1.3.63. (!) Let $d_1, \ldots, d_n$ be integers such that $d_1 \geq \cdots \geq d_n \geq 0$. Prove that there is a loopless graph (multiple edges allowed) with degree sequence $d_1, \ldots, d_n$ if and only if $\sum d_i$ is even and $d_1 \leq d_2 + \cdots + d_n$. (Hakimi [1962])

1.3.64. (!) Let $d_1 \leq \cdots \leq d_n$ be the vertex degrees of a simple graph G. Prove that G is connected if $d_j \geq j$ when $j \leq n-1-d_n$. (Hint: Consider a component that omits some vertex of maximum degree.)

1.3.65. (+) Let $a_1 < \cdots < a_k$ be distinct positive integers. Prove that there is a simple graph with $a_k + 1$ vertices whose **set** of distinct vertex degrees is $a_1, \ldots, a_k$. (Hint: Use induction on k to construct such a graph.) (Kapoor–Polimeni–Wall [1977])

1.3.66. (∗) *Expansion of 3-regular graphs* (see Example 1.3.26). For $n = 4k$, where $k \geq 2$, construct a connected 3-regular simple graph with n vertices that has no cut-edge but cannot be obtained from a smaller 3-regular simple graph by expansion. (Hint:

The desired graph must have no edge to which the inverse "erasure" operation can be applied to obtain a smaller simple graph.)

1.3.67. (*) *Construction of 3-regular simple graphs*

a) Prove that a 2-switch can be performed by performing a sequence of expansions and erasures; these operations are defined in Example 1.3.26. (Caution: Erasure is not allowed when it would produce multiple edges.)

b) Use part (a) to prove that every 3-regular simple graph can be obtained from K_4 by a sequence of expansions and erasures. (Batagelj [1984])

1.3.68. (*) Let G and H be two simple bipartite graphs, each with bipartition X, Y. Prove that $d_G(v) = d_H(v)$ for all $v \in X \cup Y$ if and only if there is a sequence of 2-switches that transforms G into H without ever changing the bipartition (each 2-switch replaces two edges joining X and Y by two other edges joining X and Y).

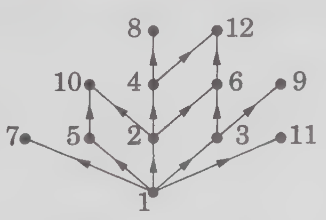
1.4. Directed Graphs

We have used graphs to model symmetric relations. Relation need not be symmetric; in general, a relation on S can be any set of ordered pairs in $S \times S$ (see Appendix A). For such relations, we need a more general model.

DEFINITIONS AND EXAMPLES

Seeking a graphical representation of the information in a general relation on S leads us to a model of directed graphs.

1.4.1. Example. For natural numbers x, y, we say that x is a "maximal divisor" of y if y/x is a prime number. For $S \subseteq \mathbb{N}$, the set $R = \{(x, y) \in S^2 \colon x \text{ is a maximal divisor of } y\}$ is a relation on S. To represent it graphically, we name a point in the plane for each element of S and draw an arrow from x to y whenever $(x, y) \in R$. Below we show the result when $S = [12]$. ■



1.4.2. Definition. A **directed graph** or **digraph** G is a triple consisting of a **vertex set** $V(G)$, an **edge set** $E(G)$, and a function assigning each edge an ordered pair of vertices. The first vertex of the ordered pair is the **tail** of the edge, and the second is the **head**; together, they are the **endpoints**. We say that an edge is an edge **from** its tail **to** its head.

The terms "head" and "tail" come from the arrows used to draw digraphs. As with graphs, we assign each vertex a point in the plane and each edge a curve joining its endpoints. When drawing a digraph, we give the curve a direction from the tail to the head.

When a digraph models a relation, each ordered pair is the (head, tail) pair for at most one edge. In this setting as with simple graphs, we ignore the technicality of a function assigning endpoints to edges and simply treat an edge as an ordered pair of vertices.

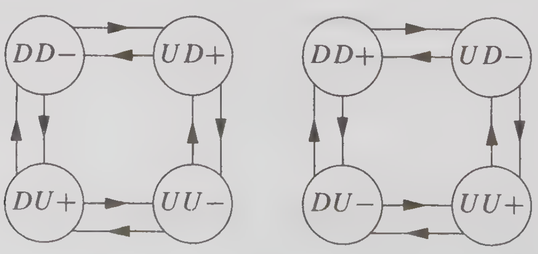
1.4.3. Definition. In a digraph, a **loop** is an edge whose endpoints are equal. **Multiple edges** are edges having the same ordered pair of endpoints. A digraph is **simple** if each ordered pair is the head and tail of at most one edge; one loop may be present at each vertex.

In a simple digraph, we write uv for an edge with tail u and head v. If there is an edge from u to v, then v is a **successor** of u, and u is a **predecessor** of v. We write $u \to v$ for "there is an edge from u to v".

1.4.4. Application. A **finite state machine** (also called **finite automaton** or **discrete system**) has a number of possible "states". Such a system can be modeled using a digraph in which vertices represent the states and edges represent the possible transitions between states.

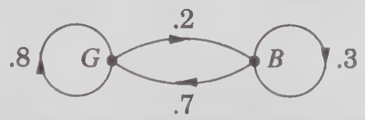
Transitions inherently move in one direction, so digraphs provide the appropriate model. Labels on the edges can be used to record the events that cause the transitions. When an event causes the system to remain in the same state, we have a loop. When two types of events can cause a particular transition, we might use multiple edges.

Consider a light controlled by two switches, often called a "three-way switch". The first switch can be up or down, the second switch can be up or down, and the light can be on (+) or off (−). Thus there are eight states. Transitions between states result by flipping switches. In the drawing below, the horizontal edges represent transitions caused by flipping the first switch, and the vertical edges represent transitions caused by flipping the second switch. (Drawing vertices large enough to put labels inside is not uncommon when discussing finite state machines, but we will stick with filled dots.) ■



1.4.5.* Application. Edge labels can be used to record transition probabilities when a system operates randomly. The probabilities on the edges leaving a vertex sum to 1, and the system is called a **Markov chain**. Methods of linear algebra can be used to compute the long-term fraction of time spent in each state.

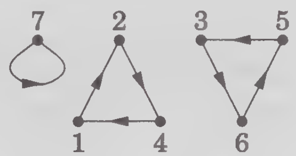
For example, suppose that weather has two states: good and bad. Air masses move slowly enough that tomorrow's weather tends to be like today's. In most places, storm systems don't linger long, so we might have transition probabilities as shown below. If we record states hourly instead of daily, then the probably of remaining in the same state is much higher. ■



1.4.6. Definition. A digraph is a **path** if it is a simple digraph whose vertices can be linearly ordered so that there is an edge with tail u and head v if and only if v immediately follows u in the vertex ordering. A **cycle** is defined similarly using an ordering of the vertices on a circle.

1.4.7. Example. *Functional digraphs.* We can study a function $f: A \to A$ using digraphs. The **functional digraph** of f is the simple digraph with vertex set A and edge set $\{(x, f(x): x \in A\}$. For each x, the single edge with tail x points to the image of x under f.

Following a path in a functional digraph corresponds to iterating the function. In a permutation, each element is the image of exactly one element, so the functional digraph has one head and one tail at each vertex. Hence the functional digraph of a permutation consists of disjoint cycles. Below we show the functional digraph for a permutation of [7]. ■

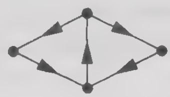


1.4.8.* Remark. We often use the same names for corresponding concepts in the graph and digraph models. Many authors replace "vertex" and "edge" with "node" and "arc" to discuss digraphs, but this obscures the analogies. Some results have the same statements and proofs; it would be wasteful to repeat them just to change terminology (especially in Chapter 4).

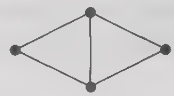
Also, a graph G can be modeled using a digraph D in which each edge $uv \in E(G)$ is replaced with $uv, vu \in E(D)$. In this way, results about digraphs can be applied to graphs. Since the notion of "edge" in digraphs extends the notion of "edge" in graphs, using the same name makes sense.

Some authors write "directed path" and "directed cycle" for our concepts of path and cycle in digraphs, but the distinction is unnecessary; for the "weak" version that does not follow the arrows, we can speak of a path or cycle in the graph obtained by ignoring the directions, which we define next. ■

1.4.9. Definition. The **underlying graph** of a digraph D is the graph G obtained by treating the edges of D as unordered pairs; the vertex set and edge set remain the same, and the endpoints of an edge are the same in G as in D, but in G they become an unordered pair.



D



G

Most ideas and methods of graph theory arise in the study of ordinary graphs. Digraphs can be a useful additional tool, especially in applications, as we have tried to suggest. We hope that describing the analogies and contrasts between graphs and digraphs will help clarify the concepts.

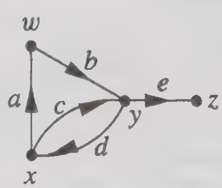
When comparing a digraph with a graph, we usually use G for the graph and D for the digraph. When discussing a single digraph, we often use G.

1.4.10. Definition. The definitions of **subgraph**, **isomorphism**, **decomposition**, and **union** are the same for graphs and digraphs. In the **adjacency matrix** $A(G)$ of a digraph G, the entry in position i, j is the number of edges from v_i to v_j. In the **incidence matrix** $M(G)$ of a loopless digraph G, we set $m_{i,j} = +1$ if v_i is the tail of e_j and $m_{i,j} = -1$ if v_i is the head of e_j.

1.4.11. Example. The underlying graph of the digraph below is the graph of Example 1.1.19; note the similarities and differences in their matrices. ■

$$\begin{array}{c} \\ w \\ x \\ y \\ z \end{array} \begin{array}{c} \begin{array}{cccc} w & x & y & z \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$A(G)$



G

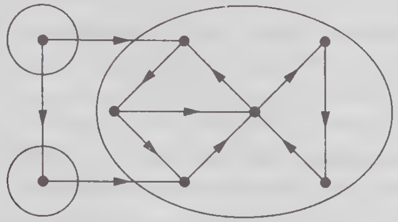
$$\begin{array}{c} \\ w \\ x \\ y \\ z \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} -1 & +1 & 0 & 0 & 0 \\ +1 & 0 & +1 & -1 & 0 \\ 0 & -1 & -1 & +1 & +1 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix} \end{array}$$

$M(G)$

To define connected digraphs, two options come to mind. We could require only that the underlying graph be connected. However, this does not capture the most useful sense of connection for digraphs.

1.4.12. Definition. A digraph is **weakly connected** if its underlying graph is connected. A digraph is **strongly connected** or **strong** if for each *ordered pair* u, v of vertices, there is a path from u to v. The **strong components** of a digraph are its maximal strong subgraphs.

1.4.13. Example. The 2-vertex digraph consisting only of the edge xy has an x, y-path but no y, x-path and is not strongly connected. As a digraph, an n-vertex path has n strong components, but a cycle has only one. In the digraph below, the three circled subdigraphs are the strong components. Properties of strong components are discussed in Exercises 10–13. ■



1.4.14.* Application. *Games.* Many games with two players can be described as finite state machines. The vertex set is the set of possible states of the game. There is an edge from state x to state y if some move can be made (by the player whose turn it is to play) to reach state y from state x.

Let W be the set of vertices for winning positions; a player who brings the game to such a state wins. No edges leave W. A player who brings the game to a state with an edge to W loses, since the other player then reaches W. One way to analyze the game is to seek a set S of pairwise nonadjacent vertices containing W such that every vertex outside S has an edge to a vertex in S. A player who can bring the game to a position in S wins, but one who must move from a position in S loses.

For example, consider a game with two piles of pennies. At his or her turn, each player can remove any portion of a single pile. The player who removes the last coin wins. The possible game positions are the nonnegative integer pairs (r, s). The definition of the game specifies $(0, 0)$ as the only winning position. However, the set S of desirable positions is $\{(r, r): r \geq 0\}$. Since only one coordinate can decrease on a move, there is no edge within S. For each vertex $(r, s) \notin S$, a player can remove $|r - s|$ from the larger pile to reach S.

The general game of Nim starts with an arbitrary number of piles with arbitrary sizes, but otherwise the rules of the game are the same as this. Exercise 18 guarantees that Nim always has a winning strategy set S, since the digraph for this game has no cycles. If the initial position is in S, then the second player wins (assuming optimal play). Otherwise, the first player wins. ■

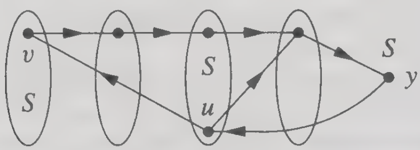
1.4.15.* Definition. A **kernel** in the digraph D is a set $S \subseteq V(D)$ such that S induces no edges and every vertex outside S has a successor in S.

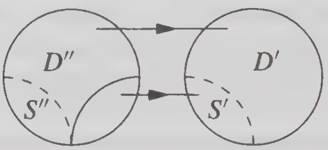
A digraph that is an odd cycle has no kernel (Exercise 17), but forbidding odd cycles as subdigraphs always yields a kernel. In proving this, all uses of paths, cycles, and walks are in the directed sense. We need several statements about movement in digraphs that hold by the same proofs as in graphs. For example, every u, v-walk in a digraph contains a u, v-path (Exercise 3), and every closed odd walk in a digraph contains an odd cycle (Exercise 4). The concept of **distance** from x to y will be explored more fully in Section 2.1; it is the least length of an x, y-path.

1.4.16.* Theorem. (Richardson [1953]) Every digraph having no odd cycle has a kernel.

Proof: Let D be such a digraph. We first consider the case that D is strongly connected; see the figure on the left below. Given an arbitrary vertex $y \in V(D)$, let S be the set of vertices with even distance to y. Every vertex with odd distance to y has a successor in S, as desired.

If the vertices of S are not pairwise nonadjacent, then there is an edge uv with $u, v \in S$. By the definition of S, there is a u, y-path P of even length and a v, y-path P' of even length. Adding uv at the start of P' yields a u, y-walk W of odd length. Because D is strong, D has a y, u-path Q. Combining Q with one of P or W yields a closed odd walk in D. This is impossible, since a closed odd walk contains an odd cycle. Thus S is a kernel in D.





For the general case, we use induction on $n(D)$.

Basis step: $n(D) = 1$. The only example is a single vertex with no loop. This vertex is a kernel by itself.

Induction step: $n(D) > 1$. Since we have already proved the claim for strong digraphs, we may assume that D is not strong. For some strong component D' of D, there is no edge from a vertex of D' to a vertex not in D' (Exercise 11). We have shown that D' has a kernel; let S' be a kernel of D'.

Let D'' be the subdigraph obtained from D by deleting D' and all the predecessors of S'. By the induction hypothesis, D'' has a kernel; let S'' be a kernel of D''. We claim that $S' \cup S''$ is a kernel of D. Since D'' has no predecessor of S', there is no edge within $S' \cup S''$. Every vertex in $D'' - S''$ has a successor in S'', and all other vertices not in $S' \cup S''$ have a successor in S'. ■

VERTEX DEGREES

In a digraph, we use the same notation for number of vertices and number of edges as in graphs. The notation for vertex degrees incorporates the distinction between heads and tails of edges.

1.4.17. Definition. Let v be a vertex in a digraph. The **outdegree** $d^+(v)$ is the number of edges with tail v. The **indegree** $d^-(v)$ is the number of edges with head v. The **out-neighborhood** or **successor set** $N^+(v)$ is $\{x \in V(G): v \to x\}$. The **in-neighborhood** or **predecessor set** $N^-(v)$ is $\{x \in V(G): x \to v\}$. The minimum and maximum indegree are $\delta^-(G)$ and $\Delta^-(G)$; for outdegree we use $\delta^+(G)$ and $\Delta^+(G)$.

The digraph analogue of the degree-sum formula for graphs is easy.

1.4.18. Proposition. In a digraph G, $\sum_{v \in V(G)} d^+(v) = e(G) = \sum_{v \in V(G)} d^-(v)$.

Proof: Every edge has exactly one tail and exactly one head. ■

The digraph analogue of degree sequence is the list of "degree pairs" $(d^+(v_i), d^-(v_i))$. When is a list of pairs realizable as the degree pairs of a digraph? As with graphs, this is easy when we allow multiple edges.

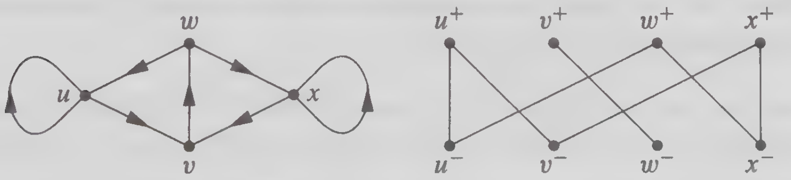
1.4.19.* Proposition. A list of pairs of nonnegative integers is realizable as the degree pairs of a digraph if and only if the sum of the first coordinates equals the sum of the second coordinates.

Proof: The condition is necessary because every edge has one tail and one head, contributing once to each sum.

For sufficiency, consider the pairs $\{(d_i^+, d_i^-) : 1 \le i \le n\}$ and vertices $v_1, \ldots, v_n$. Let $m = \sum d_i^+ = \sum d_j^-$. Consider m dots. Give the dots positive labels, with d_i^+ of them having label i. Also give the dots negative labels, with d_j^- of them having label $-j$. For each dot with labels i and $-j$, place an edge from v_i to v_j. This creates a digraph with $d^+(v_i) = d_i^+$ and $d^-(v_i) = d_i^-$. ■

The analogous question for simple digraphs is harder. The question can be rephrased in terms of bipartite graphs via a transformation that is useful in many problems about digraphs.

1.4.20.* Definition. The **split** of a digraph D is a bipartite graph G whose partite sets V^+, V^- are copies of $V(D)$. For each vertex $x \in V(D)$, there is one vertex $x^+ \in V^+$ and one vertex $x^- \in V^-$. For each edge from u to v in D, there is an edge with endpoints u^+, v^- in G.



1.4.21.* Remark. The degrees of the vertices in the split of D are the indegrees and outdegrees of the vertices in D.

Furthermore, an X, Y-bigraph G with $|X| = |Y| = n$ can be transformed into an n-vertex digraph D by putting an edge $v_i v_j$ in D for each edge $x_i y_j$ in G; now G is the split of D. (This is one reason to allow loops in simple digraphs.)

Thus there is a simple digraph with degree pairs $\{(d_i^+, d_i^-) : 1 \le i \le n\}$ if and only if there is a simple bipartite graph G in which the vertex degrees are $d_1^+, \ldots, d_n^+$ in one partite set and $d_1^-, \ldots, d_n^-$ in the other partite set. Exercise 32 obtains a recursive test for existence of such a bipartite graph. The statement and proof are like that of the Havel–Hakimi Theorem, so we leave further discussion to the exercise. ■

EULERIAN DIGRAPHS

The definitions of **trail**, **walk**, **circuit**, and the **connection relation** are the same in graphs and digraphs when we list edges as ordered pairs of vertices. In a digraph, the successive edges must "follow the arrows". In a walk $v_0, e_1, \ldots, e_k, v_k$, the edge e_i has tail v_{i-1} and head v_i.

1.4.22. Definition. An **Eulerian trail** in a digraph (or graph) is a trail containing all edges. An **Eulerian circuit** is a closed trail containing all edges. A digraph is **Eulerian** if it has an Eulerian circuit.

The characterization of Eulerian digraphs is analogous to the characterization of Eulerian graphs. The proof is essentially the same as for graphs, so we leave it to the exercises.

1.4.23. Lemma. If G is a digraph with $\delta^+(G) \geq 1$, then G contains a cycle. The same conclusion holds when $\delta^-(G) \geq 1$.

Proof: Let P be a maximal path in G, and let u be the last vertex of P. Since P cannot be extended, every successor of u must already be a vertex of P. Since $\delta^+(G) \geq 1$, u has a successor v on P. The edge uv completes a cycle with the portion of P from v to u. ■

1.4.24. Theorem. A digraph is Eulerian if and only if $d^+(v) = d^-(v)$ for each vertex v and the underlying graph has at most one nontrivial component.

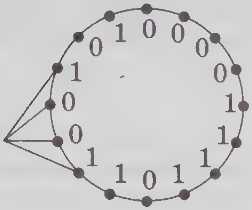
Proof: See Exercise 19 or Exercise 20. ■

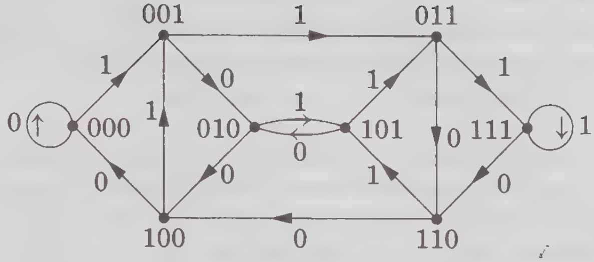
Every Eulerian digraph with no isolated vertices is strongly connected, although the characterization states that being weakly connected is sufficient.

1.4.25. Application. *de Bruijn cycles*. There are 2^n binary strings of length n. Is there a cyclic arrangement of 2^n binary digits such that the 2^n strings of n consecutive digits are all distinct? For $n = 4$, (0000111101100101) works.

We can use such an arrangement to keep track of the position of a rotating drum (Good [1946]). Our drum has 2^n rotational positions. A band around the circumference is split into 2^n portions that can be coded 0 or 1. Sensors read n consecutive portions. If the coding has the property specified above, then the position of the drum is determined by the string read by the sensors.

To obtain such a circular arrangement, define a digraph D_n whose vertices are the binary $(n-1)$-tuples. Put an edge from a to b if the last $n-2$ entries of a agree with the first $n-2$ entries of b. Label the edge with the last entry of b. Below we show D_4. We next prove that D_n is Eulerian and show how an Eulerian circuit yields the desired circular arrangement. ■





1.4.26. Theorem. The digraph D_n of Application 1.4.25 is Eulerian, and the edge labels on the edges in any Eulerian circuit of D_n form a cyclic arrangement in which the 2^n consecutive segments of length n are distinct.

Proof: We show first that D_n is Eulerian. Every vertex has outdegree 2, because we can append a 0 or a 1 to its name to obtain the name of a successor vertex. Similarly, every vertex has indegree 2, because the same argument applies when moving in reverse and putting a 0 or a 1 on the front of the name. Also, D_n is strongly connected, because we can reach the vertex $b = (b_1, \dots, b_{n-1})$ from any vertex by successively following the edges labeled $b_1, \dots, b_{n-1}$. Thus D_n satisfies the hypotheses of Theorem 1.4.24 and is Eulerian.

Let C be an Eulerian circuit of D_n. Arrival at vertex $a = (a_1, \dots, a_{n-1})$ must be along an edge with label a_{n-1}, because the label on an edge entering a vertex agrees with the last entry of the name of the vertex. Since we delete the front and shift the rest to obtain the rest of the name at the head, the successive earlier labels (looking backward) must have been $a_{n-2}, \dots, a_1$ in order. If C next uses an edge with label a_n, then the list consisting of the n most recent edge labels at that time is $a_1, \dots, a_n$.

Since the 2^{n-1} vertex labels are distinct, and the two edges leaving each vertex have distinct labels, and we traverse each edge from each vertex exactly once along C, we have shown that the 2^n strings of length n in the circular arrangement given by the edge labels along C are distinct. ■

The digraph D_n is the **de Bruijn graph** of order n on an alphabet of size 2. It is useful for other purposes, because it has many vertices and few edges (only twice the number of vertices) and yet we can reach each vertex from any other by a short path. We can reach any desired vertex in $n - 1$ steps by introducing the bits in its name in order from the current vertex.

ORIENTATIONS AND TOURNAMENTS

There are n^2 ordered pairs of elements that can be formed from a vertex set of size n. A simple digraph allows loops but uses each ordered pair at most once as an edge. Thus there are n^2 ordered pairs that may or may not be present as edges, and there are 2^{n^2} simple digraphs with vertex set $v_1, \dots, v_n$.

Sometimes we want to forbid loops.

1.4.27. Definition. An **orientation** of a graph G is a digraph D obtained from G by choosing an orientation ($x \to y$ or $y \to x$) for each edge $xy \in E(G)$. An **oriented graph** is an orientation of a simple graph. A **tournament** is an orientation of a complete graph.

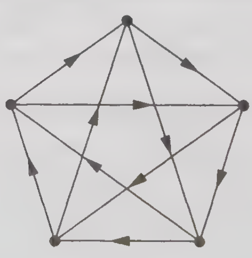
An oriented graph is the same thing as a loopless simple digraph. When the edges of a graph represent comparisons to be performed among items corresponding to the vertices, we can record the results by putting $x \to y$ when x does better than y in the comparison. The outcome is an orientation of G.

The number of oriented graphs with vertices $v_1, \ldots, v_n$ is $3^{\binom{n}{2}}$; the number of tournaments is $2^{\binom{n}{2}}$.

1.4.28. Example. Orientations of complete graphs model "round-robin tournaments". Consider an n-team league where each team plays every other exactly once. For each pair u, v, we include the edge uv if u wins or vu if v wins. At the end of the season we have an orientation of K_n. The "score" of a team is its outdegree, which equals its number of wins.

We therefore call the outdegree sequence of a tournament its **score sequence**. The outdegrees determine the indegrees, since $d^+(v) + d^-(v) = n - 1$ for every vertex v. It is easier to characterize the score sequences of tournaments than the degree sequences of simple graphs (Exercise 35). ■

A tournament may have more than one vertex with maximum outdegree, so there may be no clear "winner"—in the example below, every vertex has outdegree 2 and indegree 2. Choosing a champion when several teams have the maximum number of wins can be difficult. Although there need not be a clear winner, we show next that there must always be a team x such that, for every other team z, either x beats z or x beats some team that beats z.

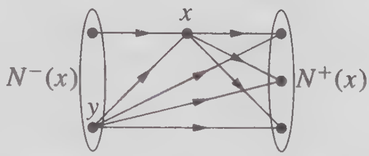


1.4.29. Definition. In a digraph, a **king** is a vertex from which every vertex is reachable by a path of length at most 2.

1.4.30. Proposition. (Landau [1953]) Every tournament has a king.

Proof: Let x be a vertex in a tournament T. If x is not a king, then some vertex y is not reachable from x by a path of length at most 2. Hence no successor of x is a predecessor of y. Since T is an orientation of a clique, every successor of x must therefore be a successor of y. Also $y \to x$. Hence $d^+(y) > d^+(x)$.

If y is not a king, then we repeat the argument to find z with yet larger outdegree. Since T is finite, we cannot forever obtain vertices of successively higher outdegree. The procedure must terminate, and it can terminate only when we have found a king. ■



In the language of extremality, we have proved that every vertex of maximum outdegree in a tournament is a king. Exercises 36–38 ask further questions about kings (see also Maurer [1980]). Exercise 39 generalizes the result to arbitrary digraphs.

EXERCISES

1.4.1. (−) Describe a relation in the real world whose digraph has no cycles. Describe another that has cycles but is not symmetric.

1.4.2. (−) In the lightswitch system of Application 1.4.4, suppose the first switch becomes disconnected from the wiring. Draw the digraph that models the resulting system.

1.4.3. (−) Prove that every u, v-walk in a digraph contains a u, v-path.

1.4.4. (−) Prove that every closed walk of odd length in a digraph contains the edges of an odd cycle. (Hint: Follow Lemma 1.2.15.)

1.4.5. (−) Let G be a digraph in which indegree equals outdegree at each vertex. Prove that G decomposes into cycles.

1.4.6. (−) Draw the de Bruijn graphs D_2 and D_4.

1.4.7. (−) Prove or disprove: If D is an orientation of a simple graph with 10 vertices, then the vertices of D cannot have distinct outdegrees.

1.4.8. (−) Prove that there is an n-vertex tournament with indegree equal to outdegree at every vertex if and only if n is odd.

• • • • •

1.4.9. For each $n \geq 1$, prove or disprove: Every simple digraph with n vertices has two vertices with the same outdegree or two vertices with the same indegree.

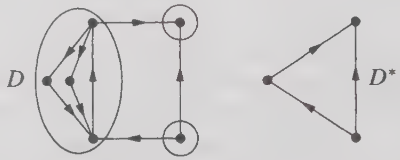
1.4.10. (!) Prove that a digraph is strongly connected if and only if for each partition of the vertex set into nonempty sets S and T, there is an edge from S to T.

1.4.11. (!) Prove that in every digraph, some strong component has no entering edges, and some strong component has no exiting edges.

1.4.12. Prove that in a digraph the connection relation is an equivalence relation, and its equivalence classes are the vertex sets of the strong components.

1.4.13. a) Prove that the strong components of a digraph are pairwise disjoint.

b) Let $D_1, \ldots, D_k$ be the strong components of a digraph D. Let D^* be the loopless digraph with vertices $v_1, \ldots, v_k$ such that $v_i \to v_j$ if and only if $i \neq j$ and D has an edge from D_i to D_j. Prove that D^* has no cycle.



1.4.14. (!) Let G be an n-vertex digraph with no cycles. Prove that the vertices of G can be ordered as $v_1, \ldots, v_n$ so that if $v_i v_j \in E(G)$, then $i < j$.

1.4.15. Let G be the simple digraph with vertex set $\{(i, j) \in \mathbb{Z}^2 \colon 0 \le i \le m \text{ and } 0 \le n\}$ and an edge from (i, j) to (i', j') if and only if (i', j') is obtained from (i, j) by adding 1 to one coordinate. Prove that the number of paths from $(0, 0)$ to (m, n) in G is $\binom{m+n}{n}$.

1.4.16. (+) *Fermat's Little Theorem.* Let $\mathbb{Z}_n$ denote the set of congruence classes of integers modulo n (see Appendix A). Let a be a natural number having no common prime factors with n; multiplication by a defines a permutation of $\mathbb{Z}_n$. Let l be the least natural number such that $a^l \equiv a \bmod n$.

a) Let G be the functional digraph with vertex set $\mathbb{Z}_n$ for the permutation defined by multiplication by a. Prove that all cycles in G (except the loop on n) have length $l-1$.

b) Conclude from part (a) that $a^{n-1} \equiv 1 \bmod n$.

1.4.17. (∗) Prove that a (directed) odd cycle is a digraph with no kernel. Construct a digraph that has an odd cycle as an induced subgraph but does have a kernel.

1.4.18. (∗) Prove that a digraph having no cycle has a unique kernel.

1.4.19. Use Lemma 1.4.23 and induction on the number of edges to prove the characterization of Eulerian digraphs (Theorem 1.4.24). (Hint: Follow Theorem 1.2.26.)

1.4.20. Prove the characterization of Eulerian digraphs (Theorem 1.4.24) using the notion of maximal trails. (Hint: Follow 1.2.32, the second proof of Theorem 1.2.26.)

1.4.21. Theorem 1.4.24 establishes necessary and sufficient conditions for a digraph to have an Eulerian circuit. Determine (with proof), the necessary and sufficient conditions for a digraph to have an Eulerian trail (Definition 1.4.22). (Good [1946])

1.4.22. Let D be a digraph with $d^-(v) = d^+(v)$ for every vertex v, except that $d^+(x) - d^-(x) = k = d^-(y) - d^+(y)$. Use the characterization of Eulerian digraphs to prove that D contains k pairwise edge-disjoint x, y-paths.

1.4.23. Prove that every graph G has an orientation D that is "balanced" at each vertex, meaning that $\left|d_D^+(v) - d_D^-(v)\right| \le 1$ for every $v \in V(G)$.

1.4.24. Prove or disprove: Every graph G has an orientation such that for every $S \subseteq V(G)$, the number of edges entering S and leaving S differ by at most 1.

1.4.25. (!) *Orientations and P_3-decomposition.*

a) Prove that every connected graph has an orientation in which the number of vertices with odd outdegree is at most 1. (Rotman [1991])

b) Use part (a) to conclude that a simple connected graph with an even number of edges can be decomposed into paths with two edges.

1.4.26. Arrange seven 0's and seven 1's cyclically so that the 14 strings of four consecutive bits are all the 4-digit binary strings other than 0101 and 1010.

1.4.27. *DeBruijn sequence for any alphabet and length.* Let A be an alphabet of size k. Prove that there exists a cyclic arrangement of k^l characters chosen from A such that the k^l strings of length l in the sequence are all distinct. (Good [1946], Rees [1946])

1.4.28. Let S be an alphabet of size m. Explain how to produce a cyclic arrangement of $m^4 - m$ letters from S such that all four-letter strings of consecutive letters are different and contain at least two distinct letters.

1.4.29. (!) Suppose that G is a graph and D is an orientation of G that is strongly connected. Prove that if G has an odd cycle, then D has an odd cycle. (Hint: Consider each pair $\{v_i, v_{i+1}\}$ in an odd cycle $(v_1, \ldots, v_k)$ of G.)

1.4.30. (+) Given a strong digraph D, let $f(D)$ be the length of the shortest closed walk visiting every vertex. Prove that the maximum value of $f(D)$ over all strong digraphs with n vertices is $\lfloor (n+1)^2/4 \rfloor$ if $n \geq 2$. (Cull [1980])

1.4.31. Determine the minimum n such that there is a pair of nonisomorphic n-vertex tournaments with the same list of outdegrees.

1.4.32. Let $p = p_1, \ldots, p_m$ and $q = q_1, \ldots, q_n$ be lists of nonnegative integers. The pair (p, q) is **bigraphic** if there is a simple bipartite graph in which $p_1, \ldots, p_m$ are the degrees for one partite set and $q_1, \ldots, q_n$ are the degrees for the other. When p has positive sum, prove that (p, q) is bigraphic if and only if (p', q') is bigraphic, where (p', q') is obtained from (p, q) by deleting the largest element Δ from p and subtracting 1 from each of the Δ largest elements of q. (Hint: Follow the method of Theorem 1.3.31.)

1.4.33. (*) Let A and B be two m by n matrices with entries in $\{0, 1\}$. An *exchange* operation substitutes a submatrix of the form $\binom{01}{10}$ for a submatrix of the form $\binom{10}{01}$ or vice versa. Prove that if A and B have the same list of row sums and have the same list of column sums, then A can be transformed into B by a sequence of exchange operations. Interpret this conclusion in the context of bipartite graphs. (Ryser [1957])

1.4.34. (!) Let G and H be two tournaments on a vertex set V. Prove that $d_G^+(v) = d_H^+(v)$ for all $v \in V$ if and only if G can be turned into H by a sequence of direction-reversals on cycles of length 3. (Hint: Consider a vertex of maximum outdegree in the subgraph of G consisting of edges oriented oppositely in H.) (Ryser [1964])

1.4.35. (+) Let $p_1, \ldots, p_n$ be nonnegative integers with $p_1 \leq \cdots \leq p_n$. Let $p_k' = \sum_{i=1}^k p_i$. Prove that there exists a tournament with outdegrees $p_1, \ldots, p_n$ if and only if $p_k' \geq \binom{k}{2}$ for $1 \leq k < n$ and $p_n' = \binom{n}{2}$. (Hint: Use induction on $\sum_{k=1}^n [p_k' - \binom{k}{2}]$.) (Landau [1953])

1.4.36. By Proposition 1.4.30, every tournament has a king. Let T be a tournament having no vertex with indegree 0.

a) Prove that if x is a king in T, then T has another king in $N^-(x)$.

b) Use part (a) to prove that T has at least three kings.

c) For each $n \geq 3$, construct a tournament T with $\delta^-(T) > 0$ and only 3 kings.

(Comment: There exists an n-vertex tournament having exactly k kings whenever $n \geq k \geq 1$ except when $k = 2$ and when $n = k = 4$.) (Maurer [1980])

1.4.37. Consider the following algorithm whose input is a tournament T.

1) Select a vertex x in T.

2) If x has indegree 0, call x a king of T and stop.

3) Otherwise, delete $\{x\} \cup N^+(x)$ from T to form T'.

4) Run the algorithm on T'; call the output a king in T and stop.

Prove that this algorithm terminates and produces a king in T.

1.4.38. (+) For $n \in \mathbb{N}$, prove that there is an n-vertex tournament in which every vertex is a king if and only if $n \notin \{2, 4\}$.

1.4.39. (+) Prove that every loopless digraph D has a set S of pairwise nonadjacent vertices such that every vertex outside S is reached from S by a path of length at most 2. (Hint: Use strong induction on $n(D)$. Comment: This generalizes Proposition 1.4.30.) (Chvátal–Lovász [1974])

1.4.40. A directed graph is **unipathic** if for every pair of vertices x, y there is at most one (directed) x, y-path. Let T_n be the tournament on n vertices with the edge between v_i and v_j directed toward the vertex with larger index. What is the maximum number of edges in a unipathic subgraph of T_n? How many unipathic subgraphs are there with the maximum number of edges? (Hint: Show that the underlying graph has no triangles.) (Maurer–Rabinovitch–Trotter [1980])

1.4.41. Let G be a tournament. Let L_0 be a listing of $V(G)$ in some order. If y immediately follows x in L_0 but $y \to x$ in G, then yx is a **reverse edge**. We can interchange x and y in the order when yx is a reverse edge (this may increase the number of reverse edges). Suppose that a sequence $L_0, L_1, \cdots$ is produced by successively switching one reverse edge in the current order. Prove that this always leads to a list with no reverse edges. Determine the maximum number of steps to termination. (Comment: In the special case where the vertices are numbers and each edge points to the higher number of the pair, the result says that successively switching adjacent numbers that are out of order always eventually sorts the list.) (Locke [1995])

1.4.42. (!) Given an ordering $\sigma = v_1, \ldots, v_n$ of the vertices of a tournament, let $f(\sigma)$ be the sum of the lengths of the feedback edges, meaning the sum of $j - i$ over edges $v_j v_i$ such that $j > i$. Prove that every ordering minimizing $f(\sigma)$ places the vertices in nonincreasing order of outdegree. (Hint: Determine how $f(\sigma)$ changes when consecutive elements of σ are exchanged.) (Kano–Sakamoto [1983], Isaak–Tesman [1991])

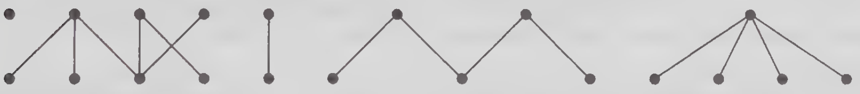
Chapter 2

Trees and Distance

2.1. Basic Properties

The word "tree" suggests branching out from a root and never completing a cycle. Trees as graphs have many applications, especially in data storage, searching, and communication.

2.1.1. Definition. A graph with no cycle is **acyclic**. A **forest** is an acyclic graph. A **tree** is a connected acyclic graph. A **leaf** (or **pendant vertex**) is a vertex of degree 1. A **spanning subgraph** of G is a subgraph with vertex set $V(G)$. A **spanning tree** is a spanning subgraph that is a tree.



2.1.2. Example. A tree is a connected forest, and every component of a forest is a tree. A graph with no cycles has no odd cycles; hence trees and forests are bipartite.

Paths are trees. A tree is a path if and only if its maximum degree is 2. A **star** is a tree consisting of one vertex adjacent to all the others. The n-vertex star is the biclique $K_{1,n-1}$.

A graph that is a tree has exactly one spanning tree; the full graph itself. A spanning subgraph of G need not be connected, and a connected subgraph of G need not be a spanning subgraph. For example:

If $n(G) > 1$, then the empty subgraph with vertex set $V(G)$ and edge set $\emptyset$ is spanning but not connected.

If $n(G) > 2$, then a subgraph consisting of one edge and its endpoints is connected but not spanning. ∎

PROPERTIES OF TREES

Trees have many equivalent characterizations, any of which could be taken as the definition. Such characterizations are useful because we need only verify that a graph satisfies any one of them to prove that it is a tree, after which we can use all the other properties.

We first prove that deleting a leaf from a tree yields a smaller tree.

2.1.3. Lemma. Every tree with at least two vertices has at least two leaves. Deleting a leaf from an n-vertex tree produces a tree with $n-1$ vertices.

Proof: A connected graph with at least two vertices has an edge. In an acyclic graph, an endpoint of a maximal nontrivial path has no neighbor other than its neighbor on the path. Hence the endpoints of a such a path are leaves.

Let v be a leaf of a tree G, and let $G' = G - v$. A vertex of degree 1 belongs to no path connecting two other vertices. Therefore, for $u, w \in V(G')$, every u, w-path in G is also in G'. Hence G' is connected. Since deleting a vertex cannot create a cycle, G' also is acyclic. Thus G' is a tree with $n-1$ vertices. ■

Lemma 2.1.3 implies that every tree with more than one vertex arises from a smaller tree by adding a vertex of degree 1 (all our graphs are finite). This rescues some proofs from the induction trap: growing an $n+1$-vertex tree from an arbitrary n-vertex tree by adding a new neighbor at an arbitrary old vertex generates all trees with $n+1$ vertices. The word "arbitrary" means that the discussion considers all ways of making the choice.

Our proof of equivalence of characterizations of trees uses induction, prior results, a counting argument, extremality, and contradiction.

2.1.4. Theorem. For an n-vertex graph G (with $n \geq 1$), the following are equivalent (and characterize the trees with n vertices).

A) G is connected and has no cycles.
B) G is connected and has $n-1$ edges.
C) G has $n-1$ edges and no cycles.
D) For $u, v \in V(G)$, G has exactly one u, v-path.

Proof: We first demonstrate the equivalence of A, B, and C by proving that any two of {connected, acyclic, $n-1$ edges} together imply the third.

A $\Rightarrow$ {B, C}. We use induction on n. For $n=1$, an acyclic 1-vertex graph has no edge. For $n>1$, we suppose that the implication holds for graphs with fewer than n vertices. Given an acyclic connected graph G, Lemma 2.1.3 provides a leaf v and states that $G' = G - v$ also is acyclic and connected (see figure above). Applying the induction hypothesis to G' yields $e(G') = n-2$. Since only one edge is incident to v, we have $e(G) = n-1$.

B $\Rightarrow$ {A, C}. Delete edges from cycles of G one by one until the resulting graph G' is acyclic. Since no edge of a cycle is a cut-edge (Theorem 1.2.14), G' is connected. Now the preceding paragraph implies that $e(G') = n - 1$. Since we are given $e(G) = n - 1$, no edges were deleted. Thus $G' = G$, and G is acyclic.

C $\Rightarrow$ {A, B}. Let $G_1, \ldots, G_k$ be the components of G. Since every vertex appears in one component, $\sum_i n(G_i) = n$. Since G has no cycles, each component satisfies property A. Thus $e(G_i) = n(G_i) - 1$. Summing over i yields $e(G) = \sum_i [n(G_i) - 1] = n - k$. We are given $e(G) = n - 1$, so $k = 1$, and G is connected.

A $\Rightarrow$ D. Since G is connected, each pair of vertices is connected by a path. If some pair is connected by more than one, we choose a shortest (total length) pair P, Q of distinct paths with the same endpoints. By this extremal choice, no internal vertex of P or Q can belong to the other path (see figure below). This implies that $P \cup Q$ is a cycle, which contradicts the hypothesis A.

D $\Rightarrow$ A. If there is a u, v-path for every $u, v \in V(G)$, then G is connected. If G has a cycle C, then G has two u, v-paths for $u, v \in V(C)$; hence G is acyclic (this also forbids loops). ■

2.1.5. Corollary. a) Every edge of a tree is a cut-edge.
b) Adding one edge to a tree forms exactly one cycle.
c) Every connected graph contains a spanning tree.

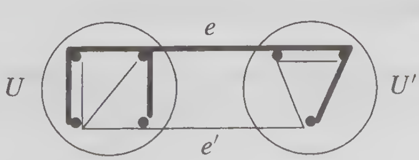
Proof: (a) A tree has no cycles, so Theorem 1.2.14 implies that every edge is a cut-edge. (b) A tree has a unique path linking each pair of vertices (Theorem 2.1.4D), so joining two vertices by an edge creates exactly one cycle. (c) As in the proof of B$\Rightarrow$A,C in Theorem 2.1.4, iteratively deleting edges from cycles in a connected graph yields a connected acyclic subgraph. ■

We apply Corollary 2.1.5 to prove two results about pairs of spanning trees. We use subtraction and addition to indicate deletion and inclusion of edges.

2.1.6. Proposition. If T, T' are spanning trees of a connected graph G and $e \in E(T) - E(T')$, then there is an edge $e' \in E(T') - E(T)$ such that $T - e + e'$ is a spanning tree of G.

Proof: By Corollary 2.1.5a, every edge of T is a cut-edge of T. Let U and U' be the two components of $T - e$. Since T' is connected, T' has an edge e' with endpoints in U and U'. Now $T - e + e'$ is connected, has $n(G) - 1$ edges, and is a spanning tree of G.

(In the figure below, T is bold, T' is solid, and they share two edges.) ■



2.1.7. Proposition. If T, T' are spanning trees of a connected graph G and $e \in E(T) - E(T')$, then there is an edge $e' \in E(T') - E(T)$ such that $T' + e - e'$ is a spanning tree of G.

Proof: By Corollary 2.1.5b, The graph $T' + e$ contains a unique cycle C. Since T is acyclic, there is an edge $e' \in E(C) - E(T)$. Deleting e' breaks the only cycle in $T' + e$. Now $T' + e - e'$ is connected and acyclic and is a spanning tree of G.

(In the figure above, adding e to T creates a cycle C of length five; all four edges of $C - e$ belong to $E(T) - E(T')$ and can serve as e'.) ■

The edge e' can be chosen to satisfy the conclusions of Propositions 2.1.6–2.1.7 simultaneously, as illustrated in the figure between them (Exercise 37).

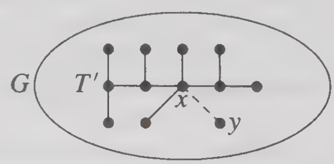
The next result illustrates proof by induction using deletion of a leaf.

2.1.8. Proposition. If T is a tree with k edges and G is a simple graph with $\delta(G) \geq k$, then T is a subgraph of G.

Proof: We use induction on k. Basis step: $k = 0$. Every simple graph contains K_1, which is the only tree with no edges.

Induction step: $k > 0$. We assume that the claim holds for trees with fewer than k edges. Since $k > 0$, Lemma 2.1.3 allows us to choose a leaf v in T; let u be its neighbor. Consider the smaller tree $T' = T - v$. By the induction hypothesis, G contains T' as a subgraph, since $\delta(G) \geq k > k - 1$.

Let x be the vertex in this copy of T' that corresponds to u (see illustration). Because T' has only $k - 1$ vertices other than u and $d_G(x) \geq k$, x has a neighbor y in G that is not in this copy of T'. Adding the edge xy expands this copy of T' into a copy of T in G, with y playing the role of v. ■



The inequality of Proposition 2.1.8 is sharp; the graph K_k has minimum degree $k - 1$, but it contains no tree with k edges. The proposition implies that every n-vertex simple graph G with more than $n(k - 1)$ edges has T as a subgraph (Exercise 34). Erdős and Sós conjectured the stronger statement that $e(G) > n(k - 1)/2$ forces T as a subgraph (Erdős [1964]). This has been proved for graphs without 4-cycles (Saclé–Woźniak [1997]). Ajtai, Komlós, and Szemerédi proved an asymptotic version, as reported in Soffer [2000].

DISTANCE IN TREES AND GRAPHS

When using graphs to model communication networks, we want vertices to be close together to avoid communication delays. We measure distance using lengths of paths.

2.1.9. Definition. If G has a u,v-path, then the **distance** from u to v, written $d_G(u,v)$ or simply $d(u,v)$, is the least length of a u,v-path. If G has no such path, then $d(u,v)=\infty$. The **diameter** (diam G) is $\max_{u,v\in V(G)} d(u,v)$.

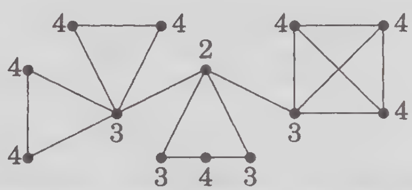
The **eccentricity** of a vertex u, written $\epsilon(u)$, is $\max_{v\in V(G)} d(u,v)$. The **radius** of a graph G, written rad G, is $\min_{u\in V(G)} \epsilon(u)$.

The diameter equals the maximum of the vertex eccentricities. In a disconnected graph, the diameter and radius (and every eccentricity) are infinite, because distance between vertices in different components is infinite. We use the word "diameter" due to its use in geometry, where it is the greatest distance between two elements of a set.

2.1.10. Example. The Petersen graph has diameter 2, since nonadjacent vertices have a common neighbor. The hypercube Q_k has diameter k, since it takes k steps to change all k coordinates. The cycle C_n has diameter $\lfloor n/2 \rfloor$. In each of these, every vertex has the same eccentricity, and diam G = rad G.

For $n \ge 3$, the n-vertex tree of least diameter is the star, with diameter 2 and radius 1. The one of largest diameter is the path, with diameter $n-1$ and radius $\lceil (n-1)/2 \rceil$. Every path in a tree is the shortest (the only!) path between its endpoints, so the diameter of a tree is the length of its longest path.

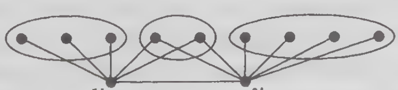
In the graph below, each vertex is labeled with its eccentricity. The radius is 2, the diameter is 4, and the length of the longest path is 7. ∎



To have large diameter, many edges must be missing. Thus we expect the complement of a graph with large diameter to have small diameter. We use the simple observation that a graph has diameter at most 2 if and only if nonadjacent vertices always have common neighbors (see also Exercise 15).

2.1.11. Theorem. If G is a simple graph, then diam $G \ge 3 \Rightarrow$ diam $\overline{G} \le 3$.

Proof: When diam $G > 2$, there exist nonadjacent vertices $u, v \in V(G)$ with no common neighbor. Hence every $x \in V(G) - \{u,v\}$ has at least one of $\{u,v\}$ as a nonneighbor. This makes x adjacent in $\overline{G}$ to at least one of $\{u,v\}$ in $\overline{G}$. Since also $uv \in E(\overline{G})$, for every pair x, y there is an x,y-path of length at most 3 in $\overline{G}$ through $\{u,v\}$. Hence diam $\overline{G} \le 3$. ∎



$\overline{G}$

2.1.12. Definition. The **center** of a graph G is the subgraph induced by the vertices of minimum eccentricity.

The center of a graph is the full graph if and only if the radius and diameter are equal. We next describe the centers of trees. In the induction step, we delete *all* leaves instead of just one.

2.1.13. Theorem. (Jordan [1869]) The center of a tree is a vertex or an edge.

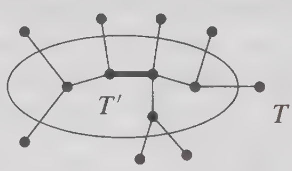
Proof: We use induction on the number of vertices in a tree T.

Basis step: $n(T) \le 2$. With at most two vertices, the center is the entire tree.

Induction step: $n(T) > 2$. Form T' by deleting every leaf of T. By Lemma 2.1.3, T' is a tree. Since the internal vertices on paths between leaves of T remain, T' has at least one vertex.

Every vertex at maximum distance in T from a vertex $u \in V(T)$ is a leaf (otherwise, the path reaching it from u can be extended farther). Since all the leaves have been removed and no path between two other vertices uses a leaf, $\epsilon_{T'}(u) = \epsilon_T(u) - 1$ for every $u \in V(T')$. Also, the eccentricity of a leaf in T is greater than the eccentricity of its neighbor in T. Hence the vertices minimizing $\epsilon_T(u)$ are the same as the vertices minimizing $\epsilon_{T'}(u)$.

We have shown that T and T' have the same center. By the induction hypothesis, the center of T' is a vertex or an edge. ■



In a communication network, large diameter may be acceptable if most pairs can communicate via short paths. This leads us to study the average distance instead of the maximum. Since the average is the sum divided by $\binom{n}{2}$ (the number of vertex pairs), it is equivalent to study $D(G) = \sum_{u,v \in V(G)} d_G(u, v)$.

The sum $D(G)$ has been called the **Wiener index** of G (also written $W(G)$). Wiener used it to study the boiling point of paraffin. Molecules can be modeled by graphs with vertices for atoms and edges for atomic bonds. Many chemical properties of molecules are related to the Wiener index of the corresponding graphs. We study the extreme values of $D(G)$.

2.1.14. Theorem. Among trees with n vertices, the Wiener index $D(T) = \sum_{u,v} d(u, v)$ is minimized by stars and maximized by paths, both uniquely.

Proof: Since a tree has $n - 1$ edges, it has $n - 1$ pairs of vertices at distance 1, and all other pairs have distance at least 2. The star achieves this and hence minimizes $D(T)$. To show that no other tree achieves this, consider a leaf x in T, and let v be its neighbor. If all other vertices have distance 2 from x, then they must be neighbors of v, and T is the star. The value is $D(K_{1,n-1}) = (n - 1) + 2\binom{n-1}{2} = (n - 1)^2$.

For the maximization, consider first $D(P_n)$. This equals the sum of the distances from an endpoint u to the other vertices, plus $D(P_{n-1})$. We have $\sum_{v\in V(P_n)} d(u,v) = \sum_{i=0}^{n-1} i = \binom{n}{2}$. Thus $D(P_n) = D(P_{n-1}) + \binom{n}{2}$. With Pascal's Formula $\binom{n}{k} + \binom{n}{k-1} = \binom{n+1}{k}$ (see Appendix A), induction yields $D(P_n) = \binom{n+1}{3}$.

We prove by induction on n that among n-vertex tree, P_n is the only tree that maximizes $D(T)$.

Basis step: $n = 1$. The only tree with one vertex is P_1.

Induction step: $n > 1$. Let u be a leaf of an n-vertex tree T. Now $D(T) = D(T-u) + \sum_{v\in V(T)} d(u,v)$. By the induction hypothesis, $D(T-u) \le D(P_{n-1})$, with equality if and only if $T-u$ is a path. Thus it suffices to show that $\sum_{v\in V(T)} d(u,v)$ is maximized only when T is a path and u is an endpoint of T.

Consider the list of distances from u. In P_n, this list is $1, 2, \ldots, n-1$, all distinct. A shortest path from u to a vertex farthest from u contains vertices at all distances from u, so in any tree the set of distances from u to other vertices has no gaps. Thus any repetition makes $\sum_{v\in V(T)} d(u,v)$ smaller than when u is a leaf of a path. When T is not a path, such a repetition occurs. ■

Over all connected n-vertex graphs, $D(G)$ is minimized by K_n. The maximization problem reduces to what we have already done with trees.

2.1.15. Lemma. If H is a subgraph of G, then $d_G(u,v) \le d_H(u,v)$.

Proof: Every u,v-path in H appears also in G, so the shortest u,v-path in G is no longer than the shortest u,v-path in H. ■

2.1.16. Corollary. If G is a connected n-vertex graph, then $D(G) \le D(P_n)$.

Proof: Let T be a spanning tree of G. By Lemma 2.1.15, $D(G) \le D(T)$. By Theorem 2.1.14, $D(T) \le D(P_n)$. ■

DISJOINT SPANNING TREES (optional)

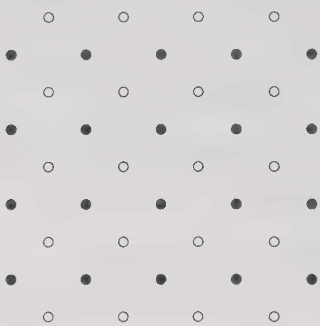
We have seen that every connected graph has a spanning tree. Edge-disjoint spanning trees provide alternate routes when an edge in the primary tree fails. Tutte [1961a] and Nash-Williams [1961] independently characterized graphs having k pairwise edge-disjoint spanning trees (see Exercise 67).

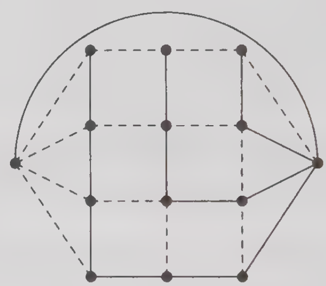
We describe one application of edge-disjoint spanning trees. David Gale devised a game marketed under the name "Bridg-it" (copyright 1960 by Hassenfeld Bros., Inc.—"Hasbro Toys"). Each of two players owns a rectangular grid of posts. The players move alternately, at each move joining two of their own posts by a unit-length bridge. The figure on the left below illustrates the board; Player 1's posts are solid, and Player 2's are hollow. The object of Player 1 is to construct a path of bridges from the left column to the right column; Player 2 wants a path of bridges from the top row to the bottom row.

Bridges cannot cross. Therefore, every bridge that is played eliminates a potential move for the other player. Since every path from left to right crosses every path from top to bottom, the players cannot both win. Note also that the layout of the board is symmetric in the two players.

We argue that Player 2 cannot have a winning strategy. Suppose otherwise. Because the board is symmetric, Player 1 can start with any move and then followthe strategy of Player 2, making an arbitrary move if the strategy of Player 2 ever calls for a bridge that has already been played. Before Player 2 can win, Player 1 wins by using the same strategy.

If the game is played until no further moves are possible, then some player must have won (Exercise 70). Since Player 2 has no winning strategy, this implies that Player 1 has a winning strategy. Here we give an explicit strategythat Player 1 can use to win. (The argument holds more generally in the contextof "matroids"—see Theorem 8.2.46.)





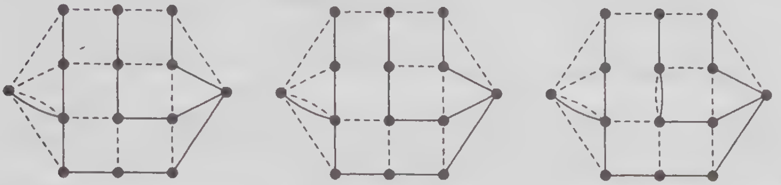
2.1.17. Theorem. Player 1 has a winning strategy in Bridg-it.

Proof: We form a graph of the potential connections for Player 1. Posts on the same end are equivalent, so we collect the (solid) posts from the end columns as single vertices. We add an auxiliary edge between the ends. The figure on the right above illustrates that this graph is the union of two edge-disjoint spanning trees; we omit a technical description of the two trees.

Together, the two trees contain edge-disjoint paths between the goal vertices. Since the auxiliary edge doesn't really exist, we pretend Player 2 moved first and took that edge. A move by Player 2 cuts one edge e in the graph and makes it no longer available. This cuts one of the trees into two components. By Proposition 2.1.6, some edge e' from the other tree reconnects it.

Player 1 chooses such an edge e'. This makes e' uncuttable, in effect putting e' in both spanning trees. After deleting e and making e' a double edge with one copy in each tree, our graph still consists of two edge-disjoint spanning trees. Since Player 2 cannot cut a double edge, Player 2 cannot cut both trees. Thus Player 1 can always defend. The figure below illustrates the strategy.

The process stops when Player 1 has won or when no single edges remain to be cut. In the latter case the remaining edges are double edges and form a spanning tree of bridges built by Player 1. Thus in either case Player 1 has constructed a path connecting the special vertices. ■



Player 1 reconnects   Player 2 cuts   Player 1 reconnects

EXERCISES

2.1.1. (−) For each k, list the isomorphism classes of trees with maximum degree k and at most six vertices. Do the same for diameter k. (Explain why there are no others.)

2.1.2. (−) Let G be a graph.
 a) Prove that G is a tree if and only if G is connected and every edge is a cut-edge.
 b) Prove that G is a tree if and only if adding any edge with endpoints in $V(G)$ creates exactly one cycle.

2.1.3. (−) Prove that a graph is a tree if and only if it is loopless and has exactly one spanning tree.

2.1.4. (−) Prove or disprove: Every graph with fewer edges than vertices has a component that is a tree.

2.1.5. (−) Let G be a graph. Prove that a maximal acyclic subgraph of G consists of a spanning tree from each component of G.

2.1.6. (−) Let T be a tree with average degree a. In terms of a, determine $n(T)$.

2.1.7. (−) Prove that every n-vertex graph with m edges has at least $m - n + 1$ cycles.

2.1.8. (−) Prove that each property below characterizes forests.
 a) Every induced subgraph has a vertex of degree at most 1.
 b) Every connected subgraph is an induced subgraph.
 c) The number of components is the number of vertices minus the number of edges.

2.1.9. (−) For $2 \le k \le n-1$, prove that the n-vertex graph formed by adding one vertex adjacent to every vertex of P_{n-1} has a spanning tree with diameter k.

2.1.10. (−) Let u and v be vertices in a connected n-vertex simple graph. Prove that if $d(u,v) > 2$, then $d(u) + d(v) \le n + 1 - d(u,v)$. Construct examples to show that this can fail whenever $n \ge 3$ and $d(u,v) \le 2$.

2.1.11. (−) Let x and y be adjacent vertices in a graph G. For all $z \in V(G)$, prove that $|d_G(x,z) - d_G(y,z)| \le 1$.

2.1.12. (−) Compute the diameter and radius of the biclique $K_{m,n}$.

2.1.13. (−) Prove that every graph with diameter d has an independent set with at least $\lceil (1+d)/2 \rceil$ vertices.

2.1.14. (−) Suppose that the processors in a computer are named by binary k-tuples, and pairs can communicate directly if and only if their names are adjacent in the k-dimensional cube Q_k. A processor with name u wants to send a message to the processor with name v. How can it find the first step on a shortest path to v?

2.1.15. (−) Let G be a simple graph with diameter at least 4. Prove that $\overline{G}$ has diameter at most 2. (Hint: Use Theorem 2.1.11.)

2.1.16. (−) Given a simple graph G, define G' to be the simple graph on the same vertex set such that $xy \in E(G')$ if and only if x and y are adjacent in G or have a common neighbor in G. Prove that $\text{diam}\,(G') = \lceil \text{diam}\,(G)/2 \rceil$.

• • • • •

2.1.17. (!) Prove C $\Rightarrow$ {A, B} in Theorem 2.1.4 by adding edges to connect components.

2.1.18. (!) Prove that every tree with maximum degree $\Delta > 1$ has at least Δ vertices of degree 1. Show that this is best possible by constructing an n-vertex tree with exactly Δ leaves, for each choice of n, Δ with $n > \Delta \geq 2$.

2.1.19. Prove or disprove: If n_i denotes the number of vertices of degree i in a tree T, then $\sum in_i$ depends only on the number of vertices in T.

2.1.20. A *saturated hydrocarbon* is a molecule formed from k carbon atoms and l hydrogen atoms by adding bonds between atoms such that each carbon atom is in four bonds, each hydrogen atom is in one bond, and no sequence of bonds forms a cycle of atoms. Prove that $l = 2k + 2$. (Bondy–Murty [1976, p27])

2.1.21. Let G be an n-vertex simple graph having a decomposition into k spanning trees. Suppose also that $\Delta(G) = \delta(G) + 1$. For $2k \geq n$, show that this is impossible. For $2k < n$, determine the degree sequence of G in terms of n and k.

2.1.22. Let T be an n-vertex tree having one vertex of each degree i with $2 \leq i \leq k$; the remaining $n - k + 1$ vertices are leaves. Determine n in terms of k.

2.1.23. Let T be a tree in which every vertex has degree 1 or degree k. Determine the possible values of $n(T)$.

2.1.24. Prove that every nontrivial tree has at least two maximal independent sets, with equality only for stars. (Note: maximal $\neq$ maximum.)

2.1.25. Prove that among trees with n vertices, the star has the most independent sets.

2.1.26. (!) For $n \geq 3$, let G be an n-vertex graph such that every graph obtained by deleting one vertex is a tree. Determine $e(G)$, and use this to determine G itself.

2.1.27. (!) Let $d_1, \ldots, d_n$ be positive integers, with $n \geq 2$. Prove that there exists a tree with vertex degrees $d_1, \ldots, d_n$ if and only if $\sum d_i = 2n - 2$.

2.1.28. Let $d_1 \geq \cdots \geq d_n$ be nonnegative integers. Prove that there exists a connected graph (loops and multiple edges allowed) with degree sequence $d_1, \ldots, d_n$ if and only if $\sum d_i$ is even, $d_n \geq 1$, and $\sum d_i \geq 2n - 2$. (Hint: Consider a realization with the fewest components.) Is the statement true for simple graphs?

2.1.29. (!) Every tree is bipartite. Prove that every tree has a leaf in its larger partite set (in both if they have equal size).

2.1.30. Let T be a tree in which all vertices adjacent to leaves have degree at least 3. Prove that T has some pair of leaves with a common neighbor.

2.1.31. Prove that a simple connected graph having exactly two vertices that are not cut-vertices is a path.

2.1.32. Prove that an edge e of a connected graph G is a cut-edge if and only if e belongs to every spanning tree. Prove that e is a loop if and only if e belongs to no spanning tree.

2.1.33. (!) Let G be a connected n-vertex graph. Prove that G has exactly one cycle if and only if G has exactly n edges.

2.1.34. (!) Let T be a tree with k edges, and let G be a simple n-vertex graph with more than $n(k-1) - \binom{k}{2}$ edges. Use Proposition 2.1.8 to prove that $T \subseteq G$ if $n > k$.

2.1.35. (!) Let T be a tree. Prove that the vertices of T all have odd degree if and only if for all $e \in E(T)$, both components of $T - e$ have odd order.

2.1.36. (!) Let T be a tree of even order. Prove that T has exactly one spanning subgraph in which every vertex has odd degree.

2.1.37. (!) Let T, T' be two spanning trees of a connected graph G. For $e \in E(T) - E(T')$, prove that there is an edge $e' \in E(T') - E(T)$ such that $T' + e - e'$ and $T - e + e'$ are both spanning trees of G.

2.1.38. Let T, T' be two trees on the same vertex set such that $d_T(v) = d'_T(v)$ for each vertex v. Prove that T' can be obtained from T' using 2-switches (Definition 1.3.32) so that every graph along the way is also a tree. (Kelmans [1998])

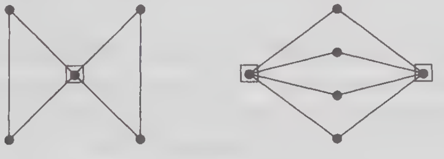
2.1.39. (!) Let G be a tree with $2k$ vertices of odd degree. Prove that G decomposes into k paths. (Hint: Prove the stronger result that the claim holds for all forests.)

2.1.40. (!) Let G be a tree with k leaves. Prove that G is the union of paths $P_1, \ldots, P_{\lceil k/2 \rceil}$ such that $P_i \cap P_j \neq \emptyset$ for all $i \neq j$. (Ando–Kaneko–Gervacio [1996])

2.1.41. For $n \geq 4$, let G be a simple n-vertex graph with $e(G) \geq 2n - 3$. Prove that G has two cycles of equal length. (Chen–Jacobson–Lehel–Shreve [1999] strengthens this.)

2.1.42. Let G be a connected Eulerian graph with at least three vertices. A vertex v in G is *extendible* if every trail beginning at v can be extended to form an Eulerian circuit of G. For example, in the graphs below only the marked vertices are extendible. Prove the following statements about G (adapted from Chartrand–Lesniak [1986, p61]).

a) A vertex $v \in V(G)$ is extendible if and only if $G - v$ is a forest. (Ore [1951])
b) If v is extendible, then $d(v) = \Delta(G)$. (Bäbler [1953])
c) All vertices of G are extendible if and only if G is a cycle.
d) If G is not a cycle, then G has at most two extendible vertices.



2.1.43. Let u be a vertex in a connected graph G. Prove that it is possible to select shortest paths from u to all other vertices of G so that the union of the paths is a tree.

2.1.44. (!) Prove or disprove: If a simple graph with diameter 2 has a cut-vertex, then its complement has an isolated vertex.

2.1.45. Let G be a graph having spanning trees with diameter 2 and diameter l. For $2 < k < l$, prove that G also has a spanning tree with diameter k. (Galvin)

2.1.46. (!) Prove that the trees with diameter 3 are the **double-stars** (two central vertices plus leaves). Count the isomorphism classes of double-stars with n vertices.

2.1.47. (!) *Diameter and radius.*

a) Prove that the distance function $d(u, v)$ on pairs of vertices of a graph satisfies the triangle inequality: $d(u, v) + d(v, w) \geq d(u, w)$.

b) Use part (a) to prove that diam $G \leq 2$rad G for every graph G.

c) For all positive integers r and d that satisfy $r \leq d \leq 2r$, construct a simple graph with radius r and diameter d. (Hint: Build a suitable graph with one cycle.)

2.1.48. (!) For $n \geq 4$, prove that the minimum number of edges in an n-vertex graph with diameter 2 and maximum degree $n - 2$ is $2n - 4$.

2.1.49. Let G be a simple graph. Prove that rad $G \geq 3 \Rightarrow$ rad $\overline{G} \leq 2$.

2.1.50. *Radius and eccentricity.*

a) Prove that the eccentricities of adjacent vertices differ by at most 1.

b) In terms of the radius r, determine the maximum possible distance from a vertex of eccentricity $r + 1$ to the center of G. (Hint: Use a graph with one cycle.)

2.1.51. Let x and y be distinct neighbors of a vertex v in a graph G.

a) Prove that if G is a tree, then $2\epsilon(v) \leq \epsilon(x) + \epsilon(y)$.

b) Determine the smallest graph where this inequality can fail.

2.1.52. Let x be a vertex in a graph G, and suppose that $\epsilon(x) >$ rad G.

a) Prove that if G is a tree, then x has a neighbor with eccentricity $k - 1$.

b) Show that part (a) does not hold for all graphs by constructing, for each even r that is at least 4, a graph with radius r in which x has eccentricity $r + 2$ and has no neighbor with eccentricity $r + 1$. (Hint: Use a graph with one cycle.)

2.1.53. Prove that the center of a graph can be disconnected and can have components arbitrarily far apart by constructing a graph where the center consists of two vertices and the distance between these two vertices is k.

2.1.54. *Centers of trees.* Let T be a tree.

a) Give a noninductive proof that the center of T is a vertex or an edge.

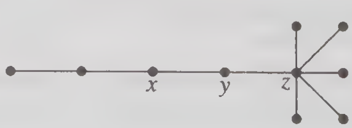
b) Prove that the center of T is one vertex if and only if diam $T = 2$rad T.

c) Use part (a) to prove that if $n(T)$ is odd, then every automorphism of T maps some vertex to itself.

2.1.55. Given $x \in V(G)$, let $s(x) = \sum_{v \in V(G)} d(x, v)$. The **barycenter** of G is the subgraph induced by the set of vertices minimizing $s(x)$ (the set is also called the **median**).

a) Prove that the barycenter of a tree is a single vertex or an edge. (Hint: Study $s(u) - s(v)$ when u and v are adjacent.) (Jordan [1869])

b) Determine the maximum distance between the center and the barycenter in a tree of diameter d. (Example: in the tree below, the center is the edge xy, the barycenter contains only z, and the distance between them is 1.)



2.1.56. Let T be a tree. Prove that T has a vertex v such that for all $e \in E(T)$, the component of $T - e$ containing v has at least $\lceil n(T)/2 \rceil$ vertices. Prove that either v is unique or there are just two adjacent such vertices.

2.1.57. Let $n_1, \ldots, n_k$ be positive integers with sum $n - 1$.

a) By counting edges in complete graphs, prove that $\sum_{i=1}^{k} \binom{n_i}{2} \le \binom{n-1}{2}$.

b) Use part (a) to prove that $\sum_{v \in V(T)} d(u, v) \le \binom{n}{2}$ when u is a vertex of a tree T. (Hint: Use strong induction on the number of vertices.)

2.1.58. (+) Let S and T be trees with leaves $\{x_1, \ldots, x_k\}$ and $\{y_1, \ldots, y_k\}$, respectively. Suppose that $d_S(x_i, x_j) = d_T(y_i, y_j)$ for each pair i, j. Prove that S and T are isomorphic. (Smolenskii [1962])

2.1.59. (!) Let G be a tree with n vertices, k leaves, and maximum degree k.

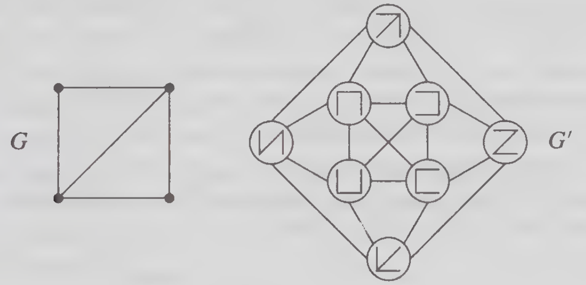
a) Prove that G is the union of k paths with a common endpoint.

b) Determine the maximum and minimum possible values of diam G.

2.1.60. Let G be a graph with diameter d and maximum degree k. Prove that $n(G) \le 1 + [(k-1)^d - 1]k/(k-2)$. (Comment: Equality holds for the Petersen graph.)

2.1.61. (+) Let G be a graph with smallest order among k-regular graphs with girth at least g (Exercise 1.3.16 establishes the existence of such graphs). Prove that G has diameter at most g. (Hint: If $d_G(x, y) > g$, modify G to obtain a smaller k-regular graph with girth at least g.) (Erdős–Sachs [1963])

2.1.62. (!) Let G be a connected graph with n vertices. Define a new graph G' having one vertex for each spanning tree of G, with vertices adjacent in G' if and only if the corresponding trees have exactly $n(G) - 2$ common edges. Prove that G' is connected. Determine the diameter of G'. An example appears below.



2.1.63. (!) Prove that every n-vertex graph with $n + 1$ edges has girth at most $\lfloor (2n+2)/3 \rfloor$. For each n, construct an example achieving this bound.

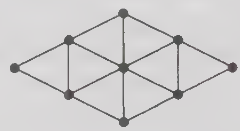
2.1.64. (!) Let G be a connected graph that is not a tree. Prove that some cycle in G has length at most $2(\text{diam } G) + 1$. For each $k \in \mathbb{N}$, show that this is best possible by exhibiting a graph with diameter k and girth $2k + 1$.

2.1.65. (+) Let G be a connected n-vertex graph with minimum degree k, where $k \ge 2$ and $n - 2 \ge 2(k+1)$. Prove that diam $G \le 3(n-2)/(k+1) - 1$. Whenever $k \ge 2$ and $(n-2)/(k+1)$ is an integer greater than 1, construct a graph where the bound holds with equality. (Moon [1965b])

2.1.66. Let $F_1, \ldots, F_m$ be forests whose union is G. Prove that $m \ge \max_{H \subseteq G} \left\lceil \frac{e(H)}{n(H)-1} \right\rceil$. (Comment: Nash-Williams [1964] and Edmonds [1965b] proved that this bound is always achieveable—Corollary 8.2.57).

2.1.67. Prove that the following is a necessary condition for the existence of k pairwise edge-disjoint spanning trees in G: for every partition of the vertices of G into r sets, there are at least $k(r-1)$ edges of G whose endpoints are in different sets of the partition. (Comment: Corollary 8.2.59 shows that this condition is also sufficient - Tutte [1961a], Nash-Williams [1961], Edmonds [1965c].)

2.1.68. Can the graph below be decomposed into edge-disjoint spanning trees? Into isomorphic edge-disjoint spanning trees?



2.1.69. (∗) Consider the graph before Theorem 2.1.17 with 12 vertical edges and 16 edges that are horizontal or slanted. Let $g_{i,j}$ be the ith edge from the top in the jth column of vertical edges. Let $h_{i,j}$ be the jth edge from the left in the ith row of horizontal/diagonal edges. Suppose that Player 1 follows the strategy of Theorem 2.1.17 and first takes $h_{1,1}$. Player 2 deletes $g_{2,2}$, and Player 1 takes $h_{2,3}$. Next Player 2 deletes $v_{3,2}$, and Player 1 takes $h_{4,2}$. Draw the two spanning trees at this point. Given that Player 2 next deletes $g_{2,1}$, list all moves available to Player 1 within the strategy. (Pritikin)

2.1.70. (∗) Prove that Bridg-it cannot end in a tie no matter how the moves are made. That is, prove that when no further moves can be made, one of the players must have built a path connecting his/her goals.

2.1.71. (∗) The players change the rules of Bridg-it so that a player with path between friendly ends is the *loser*. It is forbidden to stall by building a bridge joining end posts or joining posts already connected by a path. Show that Player 2 has a strategy that forces Player 1 to lose. (Hint: Use Proposition 2.1.7 instead of Proposition 2.1.6.) (Pritikin)

2.1.72. (+) Prove that if $G_1, \dots, G_k$ are pairwise-intersecting subtrees of a tree G, then G has a vertex that belongs to all of $G_1, \dots, G_k$. (Hint: Use induction on k. Comment: This result is the **Helly property** for trees.)

2.1.73. (+) Prove that a simple graph G is a forest if and only if for every pairwise intersecting family of paths in G, the paths have a common vertex. (Hint: For sufficiency, use induction on the size of the family of paths.)

2.1.74. Let G be a simple n-vertex graph having $n-2$ edges. Prove that G has an isolated vertex or has two components that are nontrivial trees. Use this to prove inductively that G is a subgraph of $\overline{G}$. (Comment: The claim is not true for all graphs with $n-1$ edges.) (Burns–Schuster [1977])

2.1.75. (+) Prove that every n-vertex tree other than $K_{1,n-1}$ is contained in its complement. (Hint: Use induction on n to prove a stronger statement: if T is an n-vertex tree other than a star, then K_n contains two edge-disjoint copies of T in which the two copies of each non-leaf vertex of T appear at distinct vertices.) (Burns–Schuster [1978])

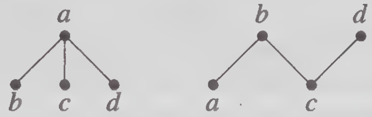
2.1.76. (+) Let S be an n-element set, and let $\{A_1, \dots, A_n\}$ be n distinct subsets of S. Prove that S has an element x such that the sets $A_1 \cup \{x\}, \dots, A_n \cup \{x\}$ are distinct. (Hint: Define a graph with vertices $a_1, \dots, a_n$ such that $a_i \leftrightarrow a_j$ if and only if one of $\{A_i, A_j\}$ is obtained from the other by adding a single element y. Use y as a label on the edge. Prove that there is a forest consisting of one edge with each label used. Use this to obtain the desired x.) (Bondy [1972a])

2.2. Spanning Trees and Enumeration

There are $2^{\binom{n}{2}}$ simple graphs with vertex set $[n] = \{1, \ldots, n\}$, since each pair may or may not form an edge. How many of these are trees? In this section, we solve this counting problem, count spanning trees in arbitrary graphs, and discuss several applications.

ENUMERATION OF TREES

With one or two vertices, only one tree can be formed. With three vertices there is still only one isomorphism class, but the adjacency matrix is determined by which vertex is the center. Thus there are three trees with vertex set [3]. With vertex set [4], there are four stars and 12 paths, yielding 16 trees. With vertex set [5], a careful study yields 125 trees.

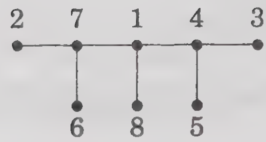


Now we may see a pattern. With vertex set $[n]$, there are n^{n-2} trees; this is **Cayley's Formula**. Prüfer, Kirchhoff, Pólya, Renyi, and others found proofs. J.W. Moon [1970] wrote a book about enumerating classes of trees. We present a bijective proof, establishing a one-to-one correspondence between the set of trees with vertex set $[n]$ and a set of known size.

Given a set S of n numbers, there are exactly n^{n-2} ways to form a list of length $n-2$ with entries in S. The set of lists is denoted S^{n-2} (see Appendix A). We use S^{n-2} to encode the trees with vertex set S. The list that results from a tree is its **Prüfer code**.

2.2.1. Algorithm. (Prüfer code) Production of $f(T) = (a_1, \ldots, a_{n-2})$.
Input: A tree T with vertex set $S \subseteq \mathbb{N}$.
Iteration: At the ith step, delete the least remaining leaf, and let a_i be the *neighbor* of this leaf. ■

2.2.2. Example. After $n-2$ iterations, only one of the original $n-1$ edges remains, and we have produced a list $f(T)$ of length $n-2$ with entries in S. In the tree below, the least leaf is 2; we delete it and record 7. After deleting 3 and 5 and recording 4 each time, the least leaf in the remaining 5-vertex tree is 4. The full code is (744171), and the vertices remaining at the end are 1 and 8. After the first step, the remainder of the Prüfer code is the Prüfer code of the subtree T' with vertex set $[8] - \{2\}$.



If we know the vertex set S, then we can retrieve the tree from the code a. The idea is to retrieve all the edges. We start with the set S of isolated vertices. At each step we create one edge and mark one vertex. When we are ready to consider a_i, there remain $n-i+1$ unmarked vertices and $n-i-1$ entries of a (including a_i). Thus at least two of the unmarked vertices do not appear among the remaining entries of a. Let x be the least of these, add xa_i to the list of edges, and mark x. After repeating this $n-2$ times, two unmarked vertices remain; we join them to form the final edge.

In the example above, the least element of S not in the code is 2, so the first edge added joins 2 and 7, and we mark 2. Now the least unmarked element absent from the rest is 3, and we join it to 4, which is a_2. As we continue, we reconstruct edges in the order they were deleted to obtain a from T.

Throughout the process, each component of the graph we have grown has one unmarked vertex. This is true initially, and thus adding an edge with two unmarked endpoints combines two components. After marking one vertex of the new edge, again each component has one unmarked vertex. After $n-2$ steps, we have two unmarked vertices and therefore two components. Adding the last edge yields a connected graph. We have built a connected graph with n vertices and $n-1$ edges. By Theorem 2.1.4B, it is a tree, but we have not yet proved that its Prüfer code is a. ■

2.2.3. Theorem. (Cayley's Formula [1889]). For a set $S \subseteq \mathbb{N}$ of size n, there are n^{n-2} trees with vertex set S.

Proof: (Prüfer [1918]). This holds for $n = 1$, so we assume $n \geq 2$. We prove that Algorithm 2.2.1 defines a bijection f from the set of trees with vertex set S to the set S^{n-2} of lists of length $n-2$ from S. We must show for each $a = (a_1, \ldots, a_{n-2}) \in S^{n-2}$ that exactly one tree T with vertex set S satisfies $f(T) = a$. We prove this by induction on n.

Basis step: $n = 2$. There is tree with two vertices. The Prüfer code is a list of length 0, and it is the only such list.

Induction step: $n > 2$. Computing $f(T)$ reduces each vertex to degree 1 and then possibly deletes it. Thus every nonleaf vertex in T appears in $f(T)$. No leaf appears, because recording a leaf as a neighbor of a leaf would require reducing the tree to one vertex. Hence the leaves of T are the elements of S not in $f(T)$. If $f(T) = a$, then the first leaf deleted is the least element of S not in a (call it x), and the neighbor of x is a_1.

We are given $a \in S^{n-2}$ and seek all solutions to $f(T) = a$. We have shown that every such tree has x as its least leaf and has the edge xa_1. Deleting x leaves a tree with vertex set $S' = S - \{x\}$. Its Prüfer code is $a' = (a_2, \ldots, a_{n-2})$, an $n-3$-tuple formed from S'.

By the induction hypothesis, there exists exactly one tree T' having vertex set S' and Prüfer code a'. Since every tree with Prüfer code a is formed by adding the edge xa_1 to such a tree, there is at most one solution to $f(T) = a$. Furthermore, adding xa_1 to T' does create a tree with vertex set S and Prüfer code a, so there is at least one solution. ■

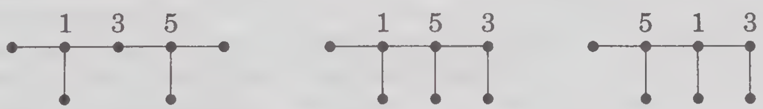
Cayley approached the problem algebraically and counted the trees by their vertex degrees. Prüfer's bijection also provides this information.

2.2.4. Corollary. Given positive integers $d_1, \ldots, d_n$ summing to $2n - 2$, there are exactly $\frac{(n-2)!}{\prod(d_i-1)!}$ trees with vertex set $[n]$ such that vertex i has degree d_i, for each i.

Proof: While constructing the Prüfer code of a tree T, we record x each time we delete a neighbor of x, until we delete x itself or leave x among the last two vertices. Thus each vertex x appears $d_T(x) - 1$ times in the Prüfer code.

Therefore, we count trees with these vertex degrees by counting lists of length $n - 2$ that for each i have $d_i - 1$ copies of i. If we assign subscripts to the copies of each i to distinguish them, then we are permuting $n - 2$ distinct objects and there are $(n - 2)!$ lists. Since the copies of i are not distinguishable, we have counted each desired arrangement $\prod(d_i - 1)!$ times, once for each way to order the subscripts on each type of label. (Appendix A discusses further aspects of this counting problem.) ■

2.2.5. Example. *Trees with fixed degrees.* Consider trees with vertices $\{1, 2, 3, 4, 5, 6, 7\}$ that have degrees $(3, 1, 2, 1, 3, 1, 1)$, respectively. We compute $\frac{(n-2)!}{\prod(d_i-1)!} = 30$; the trees are suggested below. Only the vertices $\{1, 3, 5\}$ are non-leaves. Deleting the leaves yields a subtree on $\{1, 3, 5\}$. There are three such subtrees, determined by which of the three is in the middle.



To complete each tree, we add the appropriate number of leaf neighbors for each non-leaf to give it the desired degree. There are six ways to complete the first tree (pick from the remaining four vertices the two adjacent to vertex 1) and twelve ways to complete each of the others (pick the neighbor of vertex 3 from the remaining four, and then pick the neighbor of the central vertex from the remaining three). ■

SPANNING TREES IN GRAPHS

We can interpret Cayley's Formula in another way. Since the complete graph with vertex set $[n]$ has all edges that can be used in forming trees with vertex set $[n]$, the number of trees with a specified vertex set of size n equals the number of spanning trees in a complete graph on n vertices.

We now consider the more general problem of computing the number of spanning trees in any graph G. In general, G will not have as much symmetry as a complete graph, so it is unreasonable to expect as simple a formula as for K_n, but we can hope for an algorithm that provides a simple way to compute the answer when given a graph G.

2.2.6. Example. Below is the kite. To count the spanning trees, observe that four are paths around the outside cycle in the drawing. The remaining spanning trees use the diagonal edge. Since we must include an edge to each vertex of degree 2, we obtain four more spanning trees. The total is eight. ■

In Example 2.2.6, we counted separately the trees that did or did not contain the diagonal edge. This suggests a recursive procedure to count spanning trees. It is clear that the spanning trees of G not containing e are simply the spanning trees of $G - e$, but how do we count the trees that contain e? The answer uses an elementary operation on graphs.

2.2.7. Definition. In a graph G, **contraction** of edge e with endpoints u, v is the replacement of u and v with a single vertex whose incident edges are the edges other than e that were incident to u or v. The resulting graph $G \cdot e$ has one less edge than G.

In a drawing of G, contraction of e shrinks the edge to a single point. Contracting an edge can produce multiple edges or loops. To count spanning trees correctly, we must keep multiple edges (see Example 2.2.9). In other applications of contraction, the multiple edges may be irrelevant.

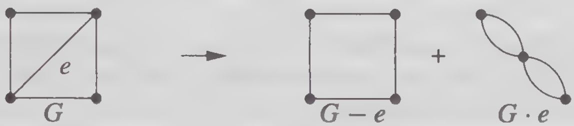
The recurrence applies for all graphs.

2.2.8. Proposition. Let $\tau(G)$ denote the number of spanning trees of a graph G. If $e \in E(G)$ is not a loop, then $\tau(G) = \tau(G - e) + \tau(G \cdot e)$.

Proof: The spanning trees of G that omit e are precisely the spanning trees of $G - e$. To show that G has $\tau(G \cdot e)$ spanning trees containing e, we show that contraction of e defines a bijection from the set of spanning trees of G containing e to the set of spanning trees of $G \cdot e$.

When we contract e in a spanning tree that contains e, we obtain a spanning tree of $G \cdot e$, because the resulting subgraph of $G \cdot e$ is spanning and connected and has the right number of edges. The other edges maintain their identity under contraction, so no two trees are mapped to the same spanning tree of $G \cdot e$ by this operation. Also, each spanning tree of $G \cdot e$ arises in this way, since expanding the new vertex back into e yields a spanning tree of G. Since each spanning tree of $G \cdot e$ arises exactly once, the function is a bijection. ■

2.2.9. Example. *A step in the recurrence.* The graphs on the right each have four spanning trees, so Proposition 2.2.8 implies that the kite has eight spanning trees. Without the multiple edges, the computation would fail. ■



We can save some computation time by recognizing special graphs G where we know $\tau(G)$, such as the graph on the right above.

2.2.10. Remark. If G is a connected loopless graph with no cycle of length at least 3, then $\tau(G)$ is the product of the edge multiplicities. A disconnected graph has no spanning trees. ■

We cannot apply the recurrence of Proposition 2.2.8 when e is a loop. For example, a graph consisting of one vertex and one loop has one spanning tree, but deleting and contracting the loop would count it twice. Since loops do not affect the number of spanning trees, we can delete loops as they arise.

Counting trees recursively requires initial conditions for graphs in which all edges are loops. Such a graph has one spanning tree if it has only one vertex, and it has no spanning trees if it has more than one vertex. If a computer completes the computation by deleting or contracting every edge in a loopless graph G, then it may compute as many as $2^{e(G)}$ terms. Even with savings from Remark 2.2.10, the amount of computation grows exponentially with the size of the graph; this is impractical.

Another technique leads to a much faster computation. The Matrix Tree Theorem, implicit in the work of Kirchhoff [1847], computes $\tau(G)$ using a determinant. This is much faster, because determinants of n-by-n matrices can be computed using fewer than n^3 operations. Also, Cayley's Formula follows from the Matrix Tree Theorem with $G = K_n$ (Exercise 17), but it does not follow easily from Proposition 2.2.8.

Before stating the theorem, we illustrate the computation it specifies.

2.2.11. Example. *A Matrix Tree computation.* Theorem 2.2.12 instructs us to form a matrix by putting the vertex degrees on the diagonal and subtracting the adjacency matrix. We then delete a row and a column and take the determinant. When G is the kite of Example 2.2.9, the vertex degrees are 3, 3, 2, 2. We form the matrix on the left below and take the determinant of the matrix in the middle. The result is the number of spanning trees! ■

$$\begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 3 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{pmatrix} \rightarrow 8$$

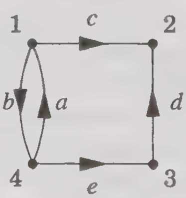
Loops don't affect spanning trees, so we delete them before the computation. The proof of the theorem uses properties of determinants.

2.2.12. Theorem. (Matrix Tree Theorem) Given a loopless graph G with vertex set $v_1, \dots, v_n$, let $a_{i,j}$ be the number of edges with endpoints v_i and v_j. Let Q be the matrix in which entry (i, j) is $-a_{i,j}$ when $i \neq j$ and is $d(v_i)$ when $i = j$. If Q^* is a matrix obtained by deleting row s and column t of Q, then $\tau(G) = (-1)^{s+t} \det Q^*$.

Proof*: We prove this only when $s = t$; the general statement follows from a result in linear algebra (when the columns of a matrix sum to the vector 0, the cofactors are constant in each row—Exercise 8.6.18).

Step 1. If D is an orientation of G, and M is the incidence matrix of D, then $Q = MM^T$. With edges $e_1, \dots, e_m$, the entries of M are $m_{i,j} = 1$ when v_i is the tail of e_j, $m_{i,j} = -1$ when v_i is the head of e_j, and $m_{i,j} = 0$ otherwise. Entry i, j in MM^T is the dot product of rows i and j of M. When $i \neq j$, the product counts -1 for every edge of G joining the two vertices; when $i = j$, it counts 1 for every incident edge and yields the degree.

$$M = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 1 & -1 & 0 & 0 & 1 \end{pmatrix} \end{array}$$



$$Q = \begin{pmatrix} 3 & -1 & 0 & -2 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -2 & 0 & -1 & 3 \end{pmatrix}$$

Step 2. If B is an $(n-1)$-by-$(n-1)$ submatrix of M, then $\det B = \pm 1$ *if the corresponding $n-1$ edges form a spanning tree of G, and otherwise* $\det B = 0$. In the first case, we use induction on n to prove that $\det B = \pm 1$. For $n = 1$, by convention a 0×0 matrix has determinant 1. For $n > 1$, let T be the spanning tree whose edges are the columns of B. Since T has at least two leaves and only one row is deleted, B has a row corresponding to a leaf x of T. This row has only one nonzero entry in B. When computing the determinant by expanding along this row, the only submatrix B' with nonzero weight in the expansion corresponds to the spanning subtree of $G - x$ obtained by deleting x and its incident edge from T. Since B' is an $(n-2)$-by-$(n-2)$ submatrix of the incidence matrix for an orientation of $G - x$, the induction hypothesis yields $\det B' = \pm 1$. Since the nonzero entry in row x is ± 1, we obtain the same result for B.

If the $n-1$ edges corresponding to columns of B do not form a spanning tree, then by Theorem 2.1.4C they contain a cycle C. We form a linear combination of the columns with coefficient 0 if the corresponding edge is not in C, $+1$ if it is followed forward by C, and -1 if it is followed backward by C. The result is total weight 0 at each vertex, so the columns are linearly dependent, which yields $\det B = 0$.

Step 3. Computation of $\det Q^*$. Let M^* be the result of deleting row t of M, so $Q^* = M^*(M^*)^T$. If $m < n-1$, then the determinant is 0 and there are no spanning subtrees, so we assume that $m \geq n-1$. The Binet–Cauchy Formula (Exercise 8.6.19) computes the determinant of a product of non-square matrices using the determinants of square submatrices of the factors. When $m \geq p$, A is p-by-m, and B is m-by-p, it states that $\det AB = \sum_S \det A_S \det B_S$, where the summ runs over all p-sets S in $[m]$, A_S is the submatrix of A consisting of the columns indexed by S, and B_S is the submatrix of B consisting of the rows indexed by S. When we apply the formula to $Q^* = M^*(M^*)^T$, the submatrix A_S is an $(n-1)$-by-$(n-1)$ submatrix of M as discussed in Step 2, and $B_S = A_S^T$. Hence the summation counts $1 = (\pm 1)^2$ for each set of $n-1$ edges corresponding to a spanning tree and 0 for all other sets of $n-1$ edges. ■

DECOMPOSITION AND GRACEFUL LABELINGS

We consider another problem about graph decomposition (Definition 1.1.32). We can always decompose G into single edges; can we decompose G into copies of a larger tree T? This requires that $e(T)$ divides $e(G)$ and $\Delta(G) \geq \Delta(T)$; is that sufficient? Even when G is $e(T)$-regular, this may fail (Exercise 20); for example, the Petersen graph does not decompose into claws.

Häggkvist conjectured that if G is a $2m$-regular graph and T is a tree with m edges, then $E(G)$ decomposes into $n(G)$ copies of T. Even the "simplest" case when G is a clique is still unsettled and notorious.

2.2.13. Conjecture. (Ringel [1964]) If T is a fixed tree with m edges, then K_{2m+1} decomposes into $2m+1$ copies of T. ■

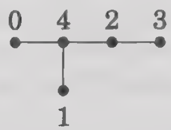
Attempts to prove Ringel's conjecture have focused on the stronger **Graceful Tree Conjecture**. This implies Ringel's conjecture and a similar statement about decomposing complete graphs of even order (Exercise 23).

2.2.14. Definition. A **graceful labeling** of a graph G with m edges is a function $f\colon V(G) \to \{0, \dots, m\}$ such that distinct vertices receive distinct numbers and $\{|f(u) - f(v)| : uv \in E(G)\} = \{1, \dots, m\}$. A graph is **graceful** if it has a graceful labeling.

2.2.15. Conjecture. (Graceful Tree Conjecture—Kotzig, Ringel [1964]) Every tree has a graceful labeling. ■

2.2.16. Theorem. (Rosa [1967]) If a tree T with m edges has a graceful labeling, then K_{2m+1} has a decomposition into $2m+1$ copies of T.

Proof: View the vertices of K_{2m+1} as the congruence classes modulo $2m+1$, arranged circularly. The *difference* between two congruence classes is 1 if they are consecutive, 2 if one class is between them, and so on up to difference m. We group the edges of K_{2m+1} by the difference between the endpoints. For $1 \le j \le m$, there are $2m+1$ edges with difference j.

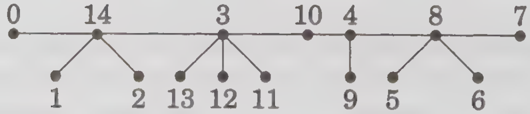


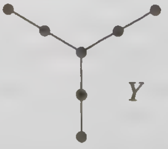
From a graceful labeling of T, we define copies of T in K_{2m+1}; the copies are $T_0, \dots, T_{2m}$. The vertices of T_k are $k, \dots, k+m$ (mod $2m+1$), with $k+i$ adjacent to $k+j$ if and only if i is adjacent to j in the graceful labeling of T. The copy T_0 looks just like the graceful labeling and has one edge with each difference. Moving to the next copy shifts each edge to another having the same difference by adding one to the name of each endpoint. Each difference class of edges has one edge in each T_k, and thus $T_0, \dots, T_{2m}$ decompose K_{2m+1}. ■

Graceful labelings are known to exist for some types of trees and for some other families of graphs (see Gallian [1998]). It is easy to find graceful labelings for stars and paths. We next define a family of trees that generalizes both by permitting the addition of edges incident to a path.

2.2.17. Definition. A **caterpillar** is a tree in which a single path (the **spine**) is incident to (or contains) every edge.

2.2.18. Example. The vertices not on the spine of a caterpillar (the "feet") are leaves. Below we show a graceful labeling of a caterpillar; in fact, every caterpillar is graceful (Exercise 31). The tree Y below is not a caterpillar. ■





2.2.19. Theorem. A tree is a caterpillar if and only if it does not contain the tree Y above.

Proof: Let G' denote the tree obtained from a tree G by deleting each leaf of G. Since all vertices that survive in G' are non-leaves in G, G' has a vertex of degree at least 3 if and only if Y appears in G. Hence G has no copy of Y if and only if $\Delta(G') \le 2$. This is equivalent to G' being a path, which is equivalent to G being a caterpillar. ∎

BRANCHINGS AND EULERIAN DIGRAPHS (optional)

Tutte extended the Matrix Tree Theorem to digraphs. His theorem reduces to the Matrix Tree Theorem when the digraph is symmetric (a digraph is *symmetric* if its adjacency matrix is symmetric, and then it models a graph). There is a surprising connection between this theorem and Eulerian circuits.

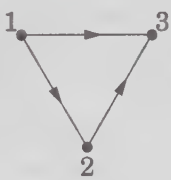
2.2.20. Definition. A **branching** or **out-tree** is an orientation of a tree having a root of indegree 0 and all other vertices of indegree 1. An **in-tree** is an out-tree with edges reversed.

A branching with root v is a union of paths from v (Exercise 33). Each vertex is reached by exactly one path. The analogous result holds for in-trees; an in-tree is a union of paths to the root, one from each vertex.

We state without proof Tutte's theorem to count branchings.

2.2.21. Theorem. (Directed Matrix Tree Theorem—Tutte [1948]) Given a loopless digraph G, let $Q^- = D^- - A'$ and $Q^+ = D^+ - A'$, where D^- and D^+ are the diagonal matrices of indegrees and outdegrees in G, and the i, jth-entry of A' is the number of edges from v_j to v_i. The number of spanning out-trees (in-trees) of G rooted at v_i is the value of each cofactor in the ith row of Q^- (ith column of Q^+). ∎

$$Q^- = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & -1 & 2 \end{pmatrix}$$



$$Q^+ = \begin{pmatrix} 2 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & -1 & 0 \end{pmatrix}$$

2.2.22. Example. The digraph above has two spanning out-trees rooted at 1 and two spanning in-trees rooted at 3. Every cofactor in the first row of Q^- is 2, and every cofactor in the third column of Q^+ is 2. ∎

Isolated vertices don't affect Eulerian circuits. After discarding these, a digraph is Eulerian if and only if indegree equals outdegree at every vertex and the underlying graph is connected (Theorem 1.4.24). Such a digraph also is strongly connected, which allows us to find a spanning in-tree. We will describe Eulerian circuits in terms of a spanning in-tree.

2.2.23. Lemma. In a strong digraph, every vertex is the root of an out-tree (and an in-tree).

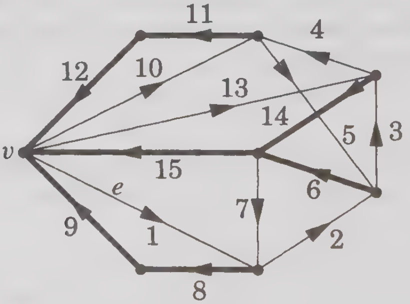
Proof: Consider a vertex v. We iteratively add edges to grow a branching from v. Let S_i be the set of vertices reached when i edges have been added; initialize $S_0 = \{v\}$. Because the digraph is strong, there is an edge leaving S_i (Exercise 1.4.10). We add one such edge to the branching and add its head to S_i to obtain S_{i+1}. This repeats until we have reached all vertices.

To obtain an in-tree of paths to v, reverse all edges and apply the same procedure; the reverse of a strong digraph is also strong. ■

The lemma constructively produces a *search tree* of paths from a root. The next section discusses search trees in more generality.

2.2.24. Algorithm. (Eulerian circuit in directed graph)
Input: An Eulerian digraph G without isolated vertices and a spanning in-tree T consisting of paths to a vertex v.
Step 1: For each $u \in V(G)$, specify an ordering of the edges that leave u, such that for $u \neq v$ the edge leaving u in T comes last.
Step 2: Beginning at v, construct an Eulerian circuit by always exiting the current vertex u along the next unused edge in the ordering specified at u. ■

2.2.25. Example. In the digraph below, the bold edges form an in-tree T of paths to v. The edges labeled in order starting with 1 form an Eulerian circuit. It leaves a vertex along an edge of T only where there is no alternative. If the ordering at v places 1 before 10 before 13, then the algorithm traverses the edges in the order indicated. ■



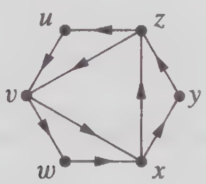
2.2.26. Theorem. Algorithm 2.2.24 always produces an Eulerian circuit.

Proof: Using Lemma 2.2.23, we construct an in-tree T to a vertex v. We then apply Algorithm 2.2.24 to construct a trail. It suffices to show that the trail can end only at v and does so only after traversing all edges.

When we enter a vertex $u \neq v$, the edge leaving u in T has not yet been used, since $d^+(u) = d^-(u)$. Thus whenever we enter u there is still a way out. Therefore the trail can only end at v.

We end when we cannot continue; we are at v and have used all exiting edges. Since $d^-(v) = d^+(v)$, we must also have used all edges entering v. Since we cannot use an edge of T until it is the only remaining edge leaving its tail, we cannot use all edges entering v until we have finished all the other vertices, since T contains a path from each vertex to v. ■

2.2.27. Example. In the digraph below, every in-tree to v contains all of uv, yz, wx, exactly one of $\{zu, zv\}$, and exactly one of $\{xy, xz\}$. There are four in-trees to v. For each in-tree, we consider $\prod(d_i - 1)! = (0!)^3(1!)^3 = 1$ orderings of the edges leaving the vertices. Hence we can obtain one Eulerian circuit from each in-tree, starting along the edge $e = vw$ from v. The four in-trees and the corresponding circuits appear below. ■



In–tree has	Circuit
zu & xy	$(v, w, x, z, v, x, y, z, u)$
zu & xz	$(v, w, x, y, z, v, x, z, u)$
zv & xy	$(v, w, x, z, u, v, x, y, z)$
zv & xz	$(v, w, x, y, z, u, v, x, z)$

Two Eulerian circuits are the same if the successive pairs of edges are the same. From each in-tree to v, Algorithm 2.2.24 generates $\prod_{u \in V(G)}(d^+(u) - 1)!$ different Eulerian circuits. The last out-edge is fixed by the tree for vertices other than v, and since we consider only the cyclic order of the edges we may also choose a particular edge e to start the ordering of edges leaving v. Any change in the exit orderings at vertices specifies at some point different choices for the next edge, so the circuits are distinct. Similarly, circuits obtained from distinct in-trees are distinct. Hence we have generated $c\prod_{u \in V(G)}(d^+(u) - 1)!$ distinct Eulerian circuits, where c is the number of in-trees to v.

In fact, these are all the Eulerian circuits. This yields a combinatorial proof that the number of in-trees to each vertex of an Eulerian digraph is the same. The graph obtained by reversing all the edges has the same number of Eulerian circuits, so the number of out-trees from any vertex also has this value, c. Theorem 2.2.21 provides a computation of c.

2.2.28. Theorem. (van Aardenne-Ehrenfest and de Bruijn [1951]). In an Eulerian digraph with $d_i = d^+(v_i) = d^-(v_i)$ the number of Eulerian circuits is $c\prod_i(d_i - 1)!$, where c counts the in-trees to or out-trees from any vertex.

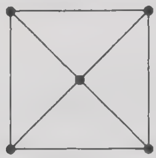
Proof: We have argued that Algorithm 2.2.24 generates this many distinct Eulerian circuits using in-trees to vertex v (starting from v alond e). We need only show that this produces all Eulerian circuits.

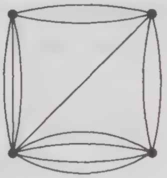
To find the tree and ordering that generates an Eulerian circuit C, follow C from e, and record the order of the edges leaving each vertex. Let T be the subdigraph consisting of the last edge on C leaving each vertex other than v. Since the last edge leaving a vertex occurs in C after all edges entering it, each edge in T extends to a path in T that reaches v. With $n - 1$ edges, T thus forms an in-tree to v. Furthermore, C is the circuit obtained by Algorithm 2.2.24 from T and the orderings of exiting edges that we recorded. ■

EXERCISES

2.2.1. (−) Determine which trees have Prüfer codes that (a) contain only one value, (b) contain exactly two values, or (c) have distinct values in all positions.

2.2.2. (−) Count the spanning trees in the graph on the left below. (Proposition 2.2.8 provides a systematic approach, and then Remark 2.2.10 and Example 2.2.6 can be used to shorten the computation.)



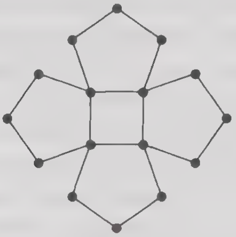


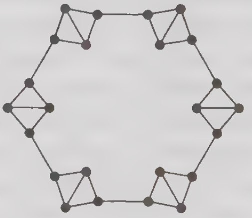
2.2.3. (−) Let G be the graph on the right above. Use the Matrix Tree Theorem to find a matrix whose determinant is $\tau(G)$. Compute $\tau(G)$.

2.2.4. (−) Let G be a simple graph with m edges. Prove that if G has a graceful labeling, then K_{2m+1} decomposes into copies of G. (Hint: Follow the proof of Theorem 2.2.16.)

• • • • •

2.2.5. The graph on the left below was the logo of the 9th Quadrennial International Conference in Graph Theory, held in Kalamazoo in 2000. Count its spanning trees.





2.2.6. (!) Let G be the 3-regular graph with $4m$ vertices formed from m pairwise disjoint kites by adding m edges to link them in a ring, as shown on the right above for $m = 6$. Prove that $\tau(G) = 2m8^m$.

2.2.7. (!) Use Cayley's Formula to prove that the graph obtained from K_n by deleting an edge has $(n-2)n^{n-3}$ spanning trees.

2.2.8. Count the following sets of trees with vertex set $[n]$, giving two proofs for each: one using the Prüfer correspondence and one by direct counting arguments.

a) trees that have 2 leaves.

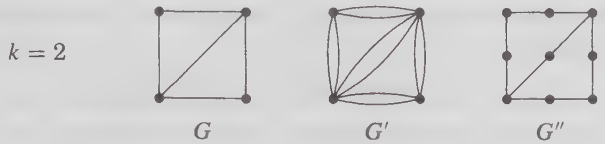
b) trees that have $n-2$ leaves.

2.2.9. Let $S(m,r)$ denote the number of partitions of an m-element set into r nonempty subsets. In terms of these numbers, count the trees with vertex set $\{v_1, \dots, v_n\}$ that have exactly k leaves. (Rényi [1959])

2.2.10. Compute $\tau(K_{2,m})$. Also compute the number of isomorphism classes of spanning trees of $K_{2,m}$.

2.2.11. (+) Compute $\tau(K_{3,m})$.

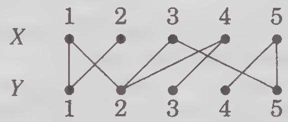
2.2.12. From a graph G we define two new graphs. Let G' be the graph obtained by replacing each edge of G with k copies of that edge. Let G'' be the graph obtained by replacing each edge $uv \in E(G)$ with a u, v-path of length k through $k-1$ new vertices. Determine $\tau(G')$ and $\tau(G'')$ in terms of $\tau(G)$ and k.



2.2.13. Consider $K_{n,n}$ with bipartition X, Y, where $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_n\}$. For each spanning tree T, we form a list $f(T)$ of ordered pairs (written vertically). Having generated part of the list, let u be the least-indexed leaf in X in the remaining subtree, and similarly let v be the least-indexed leaf in Y. Append the pair $\binom{a}{b}$ to the list, where a is the index of the neighbor of u and b is the index of the neighbor of v. Delete $\{u, v\}$. Iterate until $n-1$ pairs have been generated to form $f(T)$ (one edge remains). Part (a) shows that f is well-defined.

a) Prove that every spanning tree of $K_{n,n}$ has a leaf in each partite set.

b) Prove that f is a bijection from the set of spanning trees of $K_{n,n}$ to $([n] \times [n])^{n-1}$. Thus $K_{n,n}$ has n^{2n-2} spanning trees. (Rényi [1966], Kelmans [1992], Pritikin [1995])

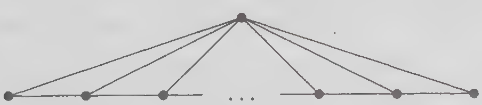


$\rightarrow \quad \binom{1}{4}, \binom{2}{1}, \binom{2}{5}, \binom{5}{3}$

2.2.14. (+) Let $f(r, s)$ be the number of trees with vertex $[n]$ that have partite sets of sizes r and s (with $r + s = n$). Prove that $f(r, s) = \binom{r+s}{s} s^{r-1} r^{s-1}$ if $r \neq s$. What is the formula when $r = s$? (Hint: First show that the Prüfer sequence for such a tree will have $r-1$ of its terms from the partite set of size s and $s-1$ of its terms from the partite set of size r.) (Scoins [1962], Glicksman [1963])

2.2.15. Let G_n be the graph with $2n$ vertices and $3n-2$ edges pictured below, for $n \geq 1$. Prove for $n > 2$ that $\tau(G_n) = 4\tau(G_{n-1}) - \tau(G_{n-2})$. (Kelmans [1967a])

2.2.16. For $n \geq 1$, let a_n be the number of spanning trees in the graph formed from P_n by adding one vertex adjacent to all of $V(P_n)$. For example, $a_1 = 1$, $a_2 = 3$, and $a_3 = 8$. Prove for $n > 1$ that $a_n = a_{n-1} + 1 + \sum_{i=1}^{n-1} a_i$. Use this to prove for $n > 2$ that $a_n = 3a_{n-1} - a_{n-2}$. (Comment: It is also possible to argue directly that $a_n = 3a_{n-1} - a_{n-2}$.)



2.2.17. Use the Matrix Tree Theorem to prove Cayley's Formula.

2.2.18. Use the Matrix Tree Theorem to compute $\tau(K_{r,s})$. (Lovász [1979, p223]—see Kelmans [1965] for a generalization)

2.2.19. (+) Prove combinatorially that the number t_n of trees with vertex set $[n]$ satisfies the recurrence $t_n = \sum_{k=1}^{n-1} k\binom{n-2}{k-1}t_k t_{n-k}$. (Comment: Since $t_n = n^{n-2}$, this proves the identity $n^{n-2} = \sum_{k=1}^{n-1}\binom{n-2}{k-1}k^{k-1}(n-k)^{n-k-2}$.) (Dziobek [1917]; see Lovász [1979, p219])

2.2.20. (!) Prove that a d-regular simple graph G has a decomposition into copies of $K_{1,d}$ if and only if it is bipartite.

2.2.21. (+) Prove that $K_{2m-1,2m}$ decomposes into m spanning paths.

2.2.22. Let G be an n-vertex simple graph that decomposes into k spanning trees. Given also that $\Delta(G) = \delta(G) + 1$, determine the degree sequence of G in terms of n and k.

2.2.23. (!) Prove that if the Graceful Tree Conjecture is true and T is a tree with m edges, then K_{2m} decomposes into $2m-1$ copies of T. (Hint: Apply the cyclically invariant decomposition of K_{2m-1} for trees with $m-1$ edges from the proof of Theorem 2.2.16.)

2.2.24. Of the n^{n-2} trees with vertex set $\{0,\dots,n-1\}$, how many are gracefully labeled by their vertex names?

2.2.25. (!) Prove that if a graph G is graceful and Eulerian, then $e(G)$ is congruent to 0 or 3 mod 4. (Hint: Sum the absolute edge differences (mod 2) in two different ways.)

2.2.26. (+) Prove that C_n is graceful if and only if 4 divides n or $n+1$. (Frucht [1979])

2.2.27. (+) Let G be the graph consisting of k 4-cycles with one common vertex. Prove that G is graceful. (Hint: Put 0 at the vertex of degree $2k$.)

2.2.28. Let $d_1,\dots,d_n$ be positive integers. Prove directly that there exists a caterpillar with vertex degrees $d_1,\dots,d_n$ if and only if $\sum d_i = 2n-2$.

2.2.29. Prove that every tree can be turned into a caterpillar with the same degree sequence using 2-switches (Definition 1.3.32) such that each intermediate graph is a tree.

2.2.30. A bipartite graph is *drawn on a channel* if the vertices of one partite set are placed on one line in the plane (in some order) and the vertices of the other partite set are placed on a line parallel to it and the edges are drawn as straight-line segments between them. Prove that a connected graph G can be drawn on a channel without edge crossings if and only if G is a caterpillar.

2.2.31. (!) An *up/down labeling* is a graceful labeling for which there exists a *critical value* α such that every edge joins vertices with labels above and below α. Prove that every caterpillar has an up/down labeling. Prove that the 7-vertex tree that is not a caterpillar has no up/down-labeling.

2.2.32. (+) Prove that the number of isomorphism classes of n-vertex caterpillars is $2^{n-4} + 2^{\lfloor n/2\rfloor-2}$ if $n \ge 3$. (Harary–Schwenk [1973], Kimble–Schwenk [1981])

2.2.33. (!) Let T be an orientation of a tree such that the heads of the edges are all distinct; the one vertex that is not a head is the *root*. Prove that T is a union of paths from the root. Prove that for each vertex of T, exactly one path reaches it from the root.

2.2.34. (*) Use Theorem 2.2.26 to prove that the algorithm below generates a binary deBruijn cycle of length 2^n (the cycle in Application 1.4.25 arises in this way).

Start with n 0's. Subsequently, append a 1 if doing so does not repeat a previous string of length n, otherwise append a 0.

2.2.35. (*) *Tarry's Algorithm* (as presented by D.G. Hoffman). Consider a castle with finitely many rooms and corridors. Each corridor has two ends; each end has a door into a room. Each room has door(s), each of which leads to a corridor. Each room can be reached from any other by traversing corridors and rooms. Initially, no doors have marks. A robot started in some room will explore the castle using the following rules.

1) After entering a corridor, traverse it and enter the room at the other end.
2) Upon entering a room with all doors unmarked, mark I on the door of entry.
3) In a room with an unmarked door, mark O on such a door and use it.
4) In a room with all doors marked, exit via a door not marked O if one exists.
5) In a room with all doors marked O, stop.

Prove that the robot traverses each corridor exactly twice, once in each direction, and then stops. (Hint: Prove that this holds for the corridors at every reached vertex, and prove that every vertex is reached. Comment: All decisions are completely local; the robot sees nothing other than the current room or corridor. Tarry's Algorithm [1895] and others are described by König [1936, p35–56] and by Fleischner [1983, 1991].)

2.3. Optimization and Trees

"The best spanning tree" may have various meanings. A **weighted graph** is a graph with numerical labels on the edges. When building links to connect locations, the costs of potential links yield a weighted graph. The minimum cost of connecting the system is the minimum total weight of its spanning trees.

Alternatively, the weights may represent distances. In these case we define the length of a path to be the sum of its edge weights. We may seek a spanning tree with small distances. When discussing weighted graphs, **we consider only nonnegative edge weights**.

We also study a problem about finding good trees to encode messages.

MINIMUM SPANNING TREE

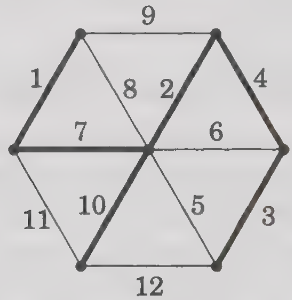
In a connected weighted graph of possible communication links, all spanning trees have $n-1$ edges; we seek one that minimizes or maximizes the sum of the edge weights. For these problems, the most naive heuristic quickly produces an optimal solution.

2.3.1. Algorithm. (Kruskal's Algorithm - for minimum spanning trees.)
Input: A weighted connected graph.
Idea: Maintain an acyclic spanning subgraph H, enlarging it by edges with low weight to form a spanning tree. Consider edges in nondecreasing order of weight, breaking ties arbitrarily.
Initialization: Set $E(H) = \emptyset$.
Iteration: If the next cheapest edge joins two components of H, then include it; otherwise, discard it. Terminate when H is connected. ■

Theorem 2.3.3 verifies that Kruskal's Algorithm produces an optimal tree. Unsophisticated locally optimal heuristics are called **greedy algorithms**. They usually don't guarantee optimal solutions, but this one does.

In a computer, the weights appear in a matrix, with huge weight on "unavailable" edges. Edges of equal weight may be examined in any order; the resulting trees have the same cost. Kruskal's Algorithm begins with a forest of n isolated vertices. Each selected edge combines two components.

2.3.2. Example. Choices in Kruskal's Algorithm depend only on the order of the weights, not on their values. In the graph below we have used positive integers as weights to emphasize the order of examination of edges. The four cheapest edges are selected, but then we cannot take the edges of weight 5 or 6. We can take the edge of weight 7, but then not those of weight 8 or 9. ■



2.3.3. Theorem. (Kruskal [1956]). In a connected weighted graph G, Kruskal's Algorithm constructs a minimum-weight spanning tree.

Proof: We show first that the algorithm produces a tree. It never chooses an edge that completes a cycle. If the final graph has more than one component, then we considered no edge joining two of them, because such an edge would be accepted. Since G is connected, some such edge exists and we considered it. Thus the final graph is connected and acyclic, which makes it a tree.

Let T be the resulting tree, and let T^* be a spanning tree of minimum weight. It $T = T^*$, we are done. If $T \neq T^*$, let e be the first edge chosen for T that is not in T^*. Adding e to T^* creates one cycle C. Since T has no cycle, C has an edge $e' \notin E(T)$. Consider the spanning tree $T^* + e - e'$.

Since T^* contains e' and all the edges of T chosen before e, both e' and e are available when the algorithm chooses e, and hence $w(e) \leq w(e')$. Thus $T^* + e - e'$ is a spanning tree with weight at most T^* that agrees with T for a longer initial list of edges than T^* does.

Repeating this argument eventually yields a minimum-weight spanning tree that agrees completely with T. Phrased extremally, we have proved that the minimum spanning tree agreeing with T the longest is T itself. ■

2.3.4.* Remark. To implement Kruskal's Algorithm, we first sort the m edge weights. We then maintain for each vertex the label of the component containing it, accepting the next cheapest edge if its endpoints have different labels. We merge the two components by changing the label of each vertex in the smaller component to the label of the larger. Since the size of the component at least doubles when a label changes, each label changes at most $\lg n$ times, and the total number of changes is at most $n \lg n$ (we use lg for the base 2 logarithm).

With this labeling method, the running time for large graphs depends on the time to sort m numbers. With this cost included, other algorithms may be faster than Kruskal's Algorithm. In *Prim's Algorithm* (Exercise 10, due also to Jarník), a spanning tree is grown from a single vertex by iteratively adding the cheapest edge that incorporates a new vertex. Prim's and Kruskal's Algorithms have similar running times when edges are pre-sorted by weight.

Both Borůvka [1926] and Jarník [1930] posed and solved the minimum spanning tree problem. Borůvka's algorithm picks the next edge by considering the cheapest edge leaving each component of the current forest. Modern improvements use clever data structures to merge components quickly. Fast versions appear in Tarjan [1984] for when the edges are pre-sorted and in Gabow–Galil–Spencer–Tarjan [1986] for when they are not. Thorough discussion and further references appear in Ahuja–Magnanti–Orlin [1993, Chapter 13]. More recent developments appear in Karger–Klein–Tarjan [1995]. ■

SHORTEST PATHS

How can we find the shortest route from one location to another? How can we find the shortest routes from our home to every place in town? This requires finding shortest paths from one vertex to all other vertices in a weighted graph. Together, these paths form a spanning tree.

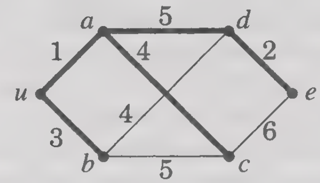
Dijkstra's Algorithm (Dijkstra [1959] and Whiting–Hillier [1960]) solves this problem quickly, using the observation that the u, v-portion of a shortest u, z-path must be a shortest u, v-path. It finds optimal routes from u to other vertices z in increasing order of $d(u, z)$. The **distance** $d(u, z)$ in a weighted graph is the minimum sum of the weights on the edges in a u, z-path (we consider only nonnegative weights).

2.3.5. Algorithm. (Dijkstra's Algorithm—distances from one vertex.)
Input: A graph (or digraph) with nonnegative edge weights and a starting vertex u. The weight of edge xy is $w(xy)$; let $w(xy) = \infty$ if xy is not an edge.
Idea: Maintain the set S of vertices to which a shortest path from u is known, enlarging S to include all vertices. To do this, maintain a tentative distance $t(z)$ from u to each $z \notin S$, being the length of the shortest u, z-path yet found.
Initialization: Set $S = \{u\}$; $t(u) = 0$; $t(z) = w(uz)$ for $z \neq u$.
Iteration: Select a vertex v outside S such that $t(v) = \min_{z \notin S} t(z)$. Add v to S. Explore edges from v to update tentative distances: for each edge vz with $z \notin S$, update $t(z)$ to $\min\{t(z), t(v) + w(vz)\}$.

The iteration continues until $S = V(G)$ or until $t(z) = \infty$ for every $z \notin S$. At the end, set $d(u, v) = t(v)$ for all v. ■

2.3.6. Example. In the weighted graph below, shortest paths from u are found to the other vertices in the order a, b, c, d, e, with distances 1,3,5,6,8, respectively. To reconstruct the paths, we only need the edge on which each shortest path arrives at its destination, because the earlier portion of a shortest u, z-path that reaches z on the edge vz is a shortest u, v-path.

The algorithm can maintain this information by recording the identity of the "selected vertex" whenever the tentative distance to z is updated. When z is selected, the vertex that was recorded when $t(z)$ was last updated is the predecessor of z on the u, z-path of length $d(u, z)$. In this example, the final edges on the paths to a, b, c, d, e generated by the algorithm are ua, ub, ac, ad, de, respectively, and these are the edges of the spanning tree generated from u. ■



With the phrasing given in Algorithm 2.3.5, Dijkstra's Algorithm works also for digraphs, generating an out-tree rooted at u if every vertex is reachable from u. The proof works for graphs and for digraphs. The technique of proving a stronger statement in order to make an inductive proof work is called "loading the induction hypothesis".

2.3.7. Theorem. Given a (di)graph G and a vertex $u \in V(G)$, Dijkstra's Algorithm computes $d(u, z)$ for every $z \in V(G)$.

Proof: We prove the stronger statement that at each iteration,

1) for $z \in S$, $t(z) = d(u, z)$, and

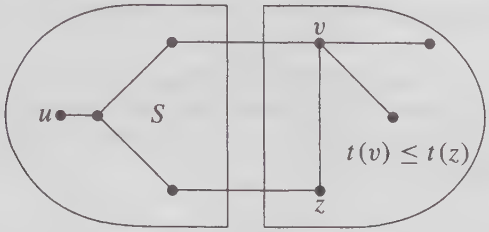
2) for $z \notin S$, $t(z)$ is the least length of a u, z-path reaching z directly from S.

We use induction on $k = |S|$. Basis step: $k = 1$. From the initialization, $S = \{u\}$, $d(u, u) = t(u) = 0$, and the least length of a u, z-path reaching z directly from S is $t(z) = w(u, z)$, which is infinite when uz is not an edge.

Induction step: Suppose that when $|S| = k$, (1) and (2) are true. Let v be a vertex among $z \notin S$ such that $t(z)$ is smallest. The algorithm now chooses v; let $S' = S \cup \{v\}$. We first argue that $d(u, v) = t(v)$. A shortest u, v-path must exit S before reaching v. The induction hypothesis states that the length of the shortest path going directly to v from S is $t(v)$. The induction hypothesis and choice of v also guarantee that a path visiting any vertex outside S and later reaching v has length at least $t(v)$. Hence $d(u, v) = t(v)$, and (1) holds for S'.

To prove (2) for S', let z be a vertex outside S other than v. By the hypothesis, the shortest u, z-path reaching z directly from S has length $t(z)$ (∞ if there is no such path). When we add v to S, we must also consider paths reaching z from v. Since we have now computed $d(u, v) = t(v)$, the shortest such path has length $t(v) + w(vz)$, and we compare this with the previous value of $t(z)$ to find the shortest path reaching z directly from S'.

We have verified that (1) and (2) hold for the new set S' of size $k + 1$; this completes the induction step. ∎



The algorithm maintains the condition that $d(u, x) \le t(z)$ for all $x \in S$ and $z \notin S$; hence it selects vertices in nondecreasing order of distance from u. It computes $d(u, v) = \infty$ when v is unreachable from u. The special case for unweighted graphs is **Breadth-First Search** from u. Here both the algorithm and the proof (Exercise 17) have simpler descriptions.

2.3.8. Algorithm. (Breadth-First Search—BFS)
Input: An unweighted graph (or digraph) and a start vertex u.
Idea: Maintain a set R of vertices that have been reached but not searched and a set S of vertices that have been searched. The set R is maintained as a First-In First-Out list (queue), so the first vertices found are the first vertices explored.
Initialization: $R = \{u\}$, $S = \emptyset$, $d(u, u) = 0$.
Iteration: As long as $R \neq \emptyset$, we search from the first vertex v of R. The neighbors of v not in $S \cup R$ are added to the back of R and assigned distance $d(u, v) + 1$, and then v is removed from the front of R and placed in S. ∎

The largest distance from a vertex u to another vertex is the eccentricity $\epsilon(u)$. Hence we can compute the diameter of a graph by running Breadth-First Search from each vertex.

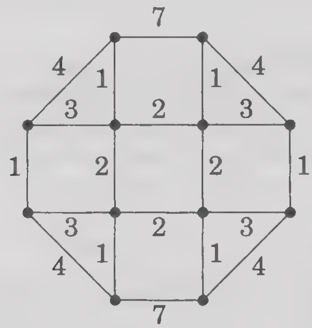
Like Dijkstra's Algorithm, BFS from u yields a tree T in which for each vertex v, the u, v-path is a shortest u, v-path. Thus the graph has no additional edges joining vertices of a u, v-path in T.

Dijkstra's Algorithm figures prominently in the solution of another well-known optimization problem.

2.3.9. Application. A mail carrier must traverse all edges in a road network, starting and ending at the Post Office. The edges have nonnegative weights representing distance or time. We seek a closed walk of minimum total length that uses all the edges. This is the **Chinese Postman Problem**, named in honor of the Chinese mathematician Guan Meigu [1962], who proposed it.

If every vertex is even, then the graph is Eulerian and the answer is the sum of the edge weights. Otherwise, we must repeat edges. Every traversal is an Eulerian circuit of a graph obtained by duplicating edges. Finding the shortest traversal is equivalent to finding the minimum total weight of edges whose duplication will make all vertex degrees even. We say "duplication" because we need not use an edge more than twice. If we use an edge three or more times in making all vertices even, then deleting two of those copies will leave all vertices even. There may be many ways to choose the duplicated edges. ■

2.3.10. Example. In the example below, the eight outer vertices have odd degree. If we match them around the outside to make the degrees even, the extra cost is $4+4+4+4=16$ or $1+7+7+1=16$. We can do better by using all the vertical edges, which total only 10. ■



Adding an edge from an odd vertex to an even vertex makes the even vertex odd. We must continue adding edges until we complete a trail to an odd vertex. The duplicated edges must consist of a collection of trails that pair the odd vertices. We may restrict our attention to paths pairing up the odd vertices (Exercise 24), but the paths may need to intersect.

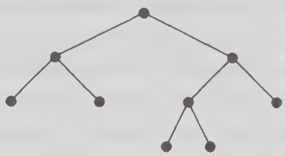
Edmonds and Johnson [1973] described a way to solve the Chinese Postman Problem. If there are only two odd vertices, then we can use Dijkstra's Algorithm to find the shortest path between them and solve the problem. If there are $2k$ odd vertices, then we still can use Dijkstra's Algorithm to find the shortest paths connecting each pair of odd vertices; these are candidates to use in the solution. We use these lengths as weights on the edges of K_{2k}, and then our problem is to find the minimum total weight of k edges that pair up these $2k$ vertices. This is a weighted version of the maximum matching problem discussed in Section 3.3. An exposition appears in Gibbons [1985, p163–165].

TREES IN COMPUTER SCIENCE (optional)

Most applications of trees in computer science use rooted trees.

2.3.11. Definition. A **rooted tree** is a tree with one vertex r chosen as **root**. For each vertex v, let $P(v)$ be the unique v,r-path. The **parent** of v is its neighbor on $P(v)$; its **children** are its other neighbors. Its **ancestors** are the vertices of $P(v)-v$. Its **descendants** are the vertices u such that $P(u)$

contains v. The **leaves** are the vertices with no children. A **rooted plane tree** or **planted tree** is a rooted tree with a left-to-right ordering specified for the children of each vertex.



After a BFS from u, we view the resulting tree T as rooted at u.

2.3.12. Definition. A **binary tree** is a rooted plane tree where each vertex has at most two children, and each child of a vertex is designated as its **left child** or **right child**. The subtrees rooted at the children of the root are the **left subtree** and the **right subtree** of the tree. A **k-ary tree** allows each vertex up to k children.

In many applications of binary trees, all non-leaves have exactly two children (Exercise 26). Binary trees permit storage of data for quick access. We store each item at a leaf and access it by following the path from the root. We encode the path by recording 0 when we move to a left child and 1 when we move to a right child. The search time is the length of this code word for the leaf. Given access probabilities among n items, we want to place them at the leaves of a rooted binary tree to minimize the expected search time.

Similarly, given large computer files and limited storage, we want to encode characters as binary lists to minimize total length. Dividing the frequencies by the total length of the file yields probabilities. This encoding problem then reduces to the problem above.

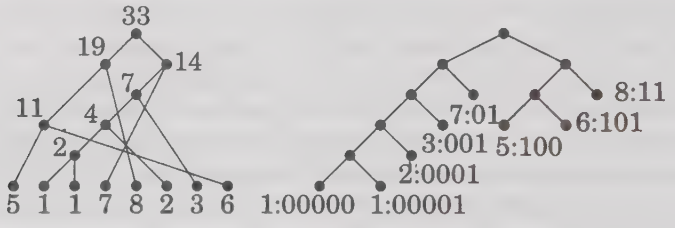
The length of code words may vary; we need a way to recognize the end of the current word. If no code word is an initial portion of another, then the current word ends as soon as the bits since the end of the previous word form a code word. Under this **prefix-free** condition, the binary code words correspond to the leaves of a binary tree using the left/right encoding described above. The expected length of a message is $\sum p_i l_i$, where the ith item has probability p_i and its code has length l_i. Constructing the optimal code is surprisingly easy.

2.3.13. Algorithm. (Huffman's Algorithm [1952]—prefix-free coding)
Input: Weights (frequencies or probabilities) $p_1, \dots, p_n$.
Output: Prefix-free code (equivalently, a binary tree).
Idea: Infrequent items should have longer codes; put infrequent items deeper by combining them into parent nodes.
Initial case: When $n = 2$, the optimal length is one, with 0 and 1 being the codes assigned to the two items (the tree has a root and two leaves; $n = 1$ can also be used as the initial case).

Recursion: When $n > 2$, replace the two least likely items p, p' with a single item q of weight $p + p'$. Treat the smaller set as a problem with $n - 1$ items. After solving it, give children with weights p, p' to the resulting leaf with weight q. Equivalently, replace the code computed for the combined item with its extensions by 1 and 0, assigned to the items that were replaced. ■

2.3.14. Example. *Huffman coding.* Consider eight items with frequencies $5, 1, 1, 7, 8, 2, 3, 6$. Algorithm 2.3.13 combines items according to the tree on the left below, working from the bottom up. First the two items of weight 1 combine to form one of weight 2. Now this and the original item of weight 2 are the least likely and combine to form an item of weight 4. The 3 and 4 now combine, after which the least likely elements are the original items of weights 5 and 6. The remaining combinations in order are $5 + 6 = 11$, $7 + 7 = 14$, $8 + 11 = 19$, and $14 + 19 = 33$.

From the drawing of this tree on the right, we obtain code words. In their original order, the items have code words 100, 00000, 00001, 01, 11, 0001, 001, and 101. The expected length is $\sum p_i l_i = 90/33$. This is less than 3, which would be the expected length of a code using the eight words of length 3. ■



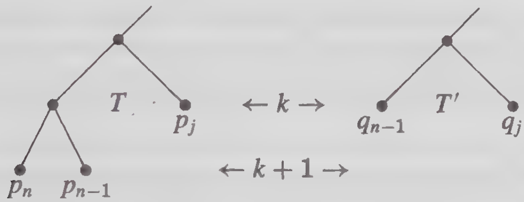
2.3.15. Theorem. Given a probability distribution $\{p_i\}$ on n items, Huffman's Algorithm produces the prefix-free code with minimum expected length.

Proof: We use induction on n. Basis step: $n = 2$. We must send a bit to send a message, and the algorithm encodes each item as a single bit, so the optimum is expected length 1.

Induction step: $n > 2$. Suppose that the algorithm computes the optimal code when given a distribution for $n - 1$ items. Every code assigns items to the leaves of a binary tree. Given a fixed tree with n leaves, we minimize the expected length by greedily assigning the messages with probabilities $p_1 \geq \cdots \geq p_n$ to leaves in increasing order of depth. Thus every optimal code has least likely messages assigned to leaves of greatest depth. Since every leaf at maximum depth has another leaf as its sibling and permuting the items at a given depth does not change the expected length, we may assume that two least likely messages appear as siblings at greatest depth.

Let T be an optimal tree for $p_1, \ldots, p_n$, with the least likely items p_n and p_{n-1} located as sibling leaves at greatest depth. Let T' be the tree obtained from T by deleting these leaves, and let $q_1, \ldots, q_{n-1}$ be the probability distribution obtained by replacing $\{p_{n-1}, p_n\}$ by $q_{n-1} = p_{n-1} + p_n$. The tree T' yields a code for $\{q_i\}$. The expected length for T is the expected length for T' plus q_{n-1}, since if k is the depth of the leaf assigned q_{n-1}, we lose kq_{n-1} and gain $(k+1)(p_{n-1}+p_n)$ in moving from T' to T.

This holds for each choice of T', so it is best to use the tree T' that is optimal for $\{q_i\}$. By the induction hypothesis, the optimal choice for T' is obtained by applying Huffman's Algorithm to $\{q_i\}$. Since the replacement of $\{p_{n-1}, p_n\}$ by q_{n-1} is the first step of Huffman's Algorithm for $\{p_i\}$, we conclude that Huffman's Algorithm generates the optimal tree T for $\{p_i\}$. ■



Huffman's Algorithm computes an optimal prefix-free code, and its expected length is close to the optimum over all types of binary codes. Shannon [1948] proved that for every code with binary digits, the expected length is at least the **entropy** of the discrete probability distribution $\{p_i\}$, defined to be $-\sum p_i \lg p_i$ (Exercise 31). When each p_i is a power of 1/2, the Huffman code meets this bound exactly (Exercise 30).

EXERCISES

2.3.1. (−) Assign integer weights to the edges of K_n. Prove that the total weight on every cycle is even if and only if the total weight on every triangle is even.

2.3.2. (−) Prove or disprove: If T is a minimum-weight spanning tree of a weighted graph G, then the u,v-path in T is a minimum-weight u,v-path in G.

2.3.3. (−) There are five cities in a network. The cost of building a road directly between i and j is the entry $a_{i,j}$ in the matrix below. An infinite entry indicates that there is a mountain in the way and the road cannot be built. Determine the least cost of making all the cities reachable from each other.

$$\begin{pmatrix} 0 & 3 & 5 & 11 & 9 \\ 3 & 0 & 3 & 9 & 8 \\ 5 & 3 & 0 & \infty & 10 \\ 11 & 9 & \infty & 0 & 7 \\ 9 & 8 & 10 & 7 & 0 \end{pmatrix}$$

2.3.4. (−) In the graph below, assign weights $(1,1,2,2,3,3,4,4)$ to the edges in two ways: one way so that the minimum-weight spanning tree is unique, and another way so that the minimum-weight spanning tree is not unique.

2.3.5. (−) There are five cities in a network. The travel time for traveling directly from i to j is the entry $a_{i,j}$ in the matrix below. The matrix is not symmetric (use directed graphs), and $a_{i,j} = \infty$ indicates that there is no direct route. Determine the least travel time and quickest route from i to j for each pair i, j.

$$\begin{pmatrix} 0 & 10 & 20 & \infty & 17 \\ 7 & 0 & 5 & 22 & 33 \\ 14 & 13 & 0 & 15 & 27 \\ 30 & \infty & 17 & 0 & 10 \\ \infty & 15 & 12 & 8 & 0 \end{pmatrix}$$

• • • • •

2.3.6. (!) Assign integer weights to the edges of K_n. Prove that on every cycle the total weight is even if and only if the subgraph consisting of the edges with odd weight is a spanning complete bipartite subgraph. (Hint: Show that every component of the subgraph consisting of the edges with even weight is a complete graph.)

2.3.7. Let G be a weighted connected graph with distinct edge weights. Without using Kruskal's Algorithm, prove that G has only one minimum-weight spanning tree. (Hint: Use Exercise 2.1.34.)

2.3.8. Let G be a weighted connected graph. Prove that no matter how ties are broken in choosing the next edge for Kruskal's Algorithm, the list of weights of a minimum spanning tree (in nondecreasing order) is unique.

2.3.9. Let F be a spanning forest of a connected weighted graph G. Among all edges of G having endpoints in different components of F, let e be one of minimum weight. Prove that among all the spanning trees of G that contain F, there is one of minimum weight that contains e. Use this to give another proof that Kruskal's Algorithm works.

2.3.10. (!) **Prim's Algorithm** grows a spanning tree from a given vertex of a connected weighted graph G, iteratively adding the cheapest edge from a vertex already reached to a vertex not yet reached, finishing when all the vertices of G have been reached. (Ties are broken arbitrarily.) Prove that Prim's Algorithm produces a minimum-weight spanning tree of G. (Jarník [1930], Prim [1957], Dijkstra [1959], independently).

2.3.11. For a spanning tree T in a weighted graph, let $m(T)$ denote the maximum among the weights of the edges in T. Let x denote the minimum of $m(T)$ over all spanning trees of a weighted graph G. Prove that if T is a spanning tree in G with minimum total weight, then $m(T) = x$ (in other words, T also minimizes the maximum weight). Construct an example to show that the converse is false. (Comment: A tree that minimizes the maximum weight is called a **bottleneck** or **minimax** spanning tree.)

2.3.12. In a weighted complete graph, iteratively select the edge of least weight such that the edges selected so far form a disjoint union of paths. After $n-1$ steps, the result is a spanning path. Prove that this algorithm always gives a minimum-weight spanning path, or give an infinite family of counterexamples where it fails.

2.3.13. (!) Let T be a minimum-weight spanning tree in G, and let T' be another spanning tree in G. Prove that T' can be transformed into T by a list of steps that exchange one edge of T' for one edge of T, such that the edge set is always a spanning tree and the total weight never increases.

2.3.14. (!) Let C be a cycle in a connected weighted graph. Let e be an edge of maximum weight on C. Prove that there is a minimum spanning tree not containing e. Use this to prove that iteratively deleting a heaviest non-cut-edge until the remaining graph is acyclic produces a minimum-weight spanning tree.

2.3.15. Let T be a minimum-weight spanning tree in a weighted connected graph G. Prove that T omits some heaviest edge from every cycle in G.

2.3.16. Four people must cross a canyon at night on a fragile bridge. At most two people can be on the bridge at once. Crossing requires carrying a flashlight, and there is only one flashlight (which can cross only by being carried). Alone, the four people cross in 10, 5, 2, 1 minutes, respectively. When two cross together, they move at the speed of the slower person. In 18 minutes, a flash flood coming down the canyon will wash away the bridge. Can the four people get across in time? Prove your answer without using graph theory and describe how the answer can be found using graph theory.

2.3.17. Given a starting vertex u in an unweighted graph or digraph G, prove directly (without Dijkstra's Algorithm) that Algorithm 2.3.8 computes $d(u, z)$ for all $z \in V(G)$.

2.3.18. Explain how to use Breadth-First Search to compute the girth of a graph.

2.3.19. (+) Prove that the following algorithm correctly finds the diameter of a tree. First, run BFS from an arbitrary vertex w to find a vertex u at maximum distance from w. Next, run BFS from u to reach a vertex v at maximum distance from u. Report diam $T = d(u, v)$. (Cormen–Leiserson–Rivest [1990, p476])

2.3.20. *Minimum diameter spanning tree.* An MDST is a spanning tree where the maximum length of a path is as small as possible. Intuition suggests that running Dijkstra's Algorithm from a vertex of minimum eccentricity (a center) will produce an MDST, but this may fail.

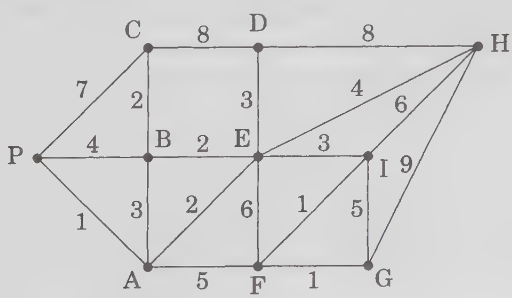
a) Construct a 5-vertex example of an unweighted graph (edge weights all equal 1) such that Dijkstra's Algorithm can be run from some vertex of minimum eccentricity and produce a spanning tree that does not have minimum diameter.

b) Construct a 4-vertex example of a weighted graph such that Dijkstra's algorithm cannot produce an MDST when run from any vertex.

2.3.21. Develop a fast algorithm to test whether a graph is bipartite. The graph is given by its adjacency matrix or by lists of vertices and their neighbors. The algorithm should not need to consider an edge more than twice.

2.3.22. (−) Solve the Chinese Postman Problem in the k-dimensional cube Q_k under the condition that every edge has weight 1.

2.3.23. Every morning the Lazy Postman takes the bus to the Post Office. From there, he chooses a route to reach home as quickly as possible (**NOT** ending at the Post Office). Below is a map of the streets along which he must deliver mail, giving the number of minutes required to walk each block whether delivering or not. P denotes the post office and H denotes home. What must the edges traveled more than once satisfy? How many times will each edge be traversed in the optimal route?



2.3.24. (−) Explain why the optimal trails pairing up odd vertices in an optimal solution to the Chinese Postman Problem may be assumed to be paths. Construct a weighted graph with four odd vertices where the optimal solution to the Chinese Postman Problem requires duplicating the edges on two paths that have a common vertex.

2.3.25. Let G be a rooted tree where every vertex has 0 or k children. Given k, for what values of $n(G)$ is this possible?

2.3.26. Find a recurrence relation to count the binary trees with $n+1$ leaves (here each non-leaf vertex has exactly two children, and the left-to-right order of children matters). When $n = 2$, the possibilities are the two trees below.

2.3.27. Find a recurrence relation for the number of rooted plane trees with n vertices. (As in a rooted binary tree, the subtrees obtained by deleting the root of a rooted plane tree are distinguished by their order from left to right.)

2.3.28. (−) Compute a code with minimum expected length for a set of ten messages whose relative frequencies are $1, 2, 3, 4, 5, 5, 6, 7, 8, 9$. What is the expected length of a message in this optimal code?

2.3.29. (−) The game of *Scrabble* has 100 tiles as listed below. This does not agree with English; "S" is less frequent here, for example, to improve the game. Pretend that these are the relative frequencies in English, and compute a prefix-free code of minimum expected length for transmitting messages. Give the answer by listing the relative frequency for each length of code word. Compute the expected length of the code (per text character). (Comment: ASCII coding uses five bits per letter; this code will beat that. Of course, ASCII suffers the handicap of including codes for punctuation.)

A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*	*R*	*S*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*	*Ø*
9	2	2	4	12	2	3	2	9	1	1	4	2	6	8	2	1	6	4	6	4	2	2	1	2	1	2

2.3.30. Consider n messages occurring with probabilities $p_1, \ldots, p_n$, such that each p_i is a power of $1/2$ (each $p_i \geq 0$ and $\sum p_i = 1$).

a) Prove that the two least likely messages have equal probability.

b) Prove that the expected message length of the Huffman code for this distribution is $-\sum p_i \lg p_i$.

2.3.31. (+) Suppose that n messages occur with probabilities $p_1, \ldots, p_n$ and that the words are assigned distinct binary code words. Prove that for every code, the expected length of a code word with respect to this distribution is at least $-\sum p_i \lg p_i$. (Hint: Use induction on n.) (Shannon [1948])

Chapter 3

Matchings and Factors

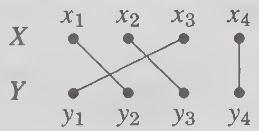
3.1. Matchings and Covers

Within a set of people, some pairs are compatible as roommates; under what conditions can we pair them all up? Many applications of graphs involve such pairings. In Example 1.1.9 we considered the problem of filling jobs with qualified applicants. Bipartite graphs have a natural vertex partition into two sets, and we want to know whether the two sets can be paired using edges. In the roommate question, the graph need not be bipartite.

3.1.1. Definition. A **matching** in a graph G is a set of non-loop edges with no shared endpoints. The vertices incident to the edges of a matching M are **saturated** by M; the others are **unsaturated** (we say M*-saturated* and M*-unsaturated*). A **perfect matching** in a graph is a matching that saturates every vertex.

3.1.2. Example. *Perfect matchings in* $K_{n,n}$. Consider $K_{n,n}$ with partite sets $X = \{x_1, \dots, x_n\}$ and $Y = \{y_1, \dots, y_n\}$. A perfect matching defines a bijection from X to Y. Successively finding mates for $x_1, x_2, \dots$ yields $n!$ perfect matchings.

Each matching is represented by a permutation of $[n]$, mapping i to j when x_i is matched to y_j. We can express the matchings as matrices. With X and Y indexing the rows and columns, we let position i, j be 1 for each edge $x_i y_j$ in a matching M to obtain the corresponding matrix. There is one 1 in each row and each column. ■



$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

3.1.3. Example. *Perfect matchings in complete graphs.* Since it has odd order, K_{2n+1} has no perfect matching. The number f_n of perfect matchings in K_{2n} is the number of ways to pair up $2n$ distinct people. There are $2n-1$ choices for the partner of v_{2n}, and for each such choice there are f_{n-1} ways to complete the matching. Hence $f_n = (2n-1)f_{n-1}$ for $n \geq 1$. With $f_0 = 1$, it follows by induction that $f_n = (2n-1) \cdot (2n-3) \cdots (1)$.

There is also a counting argument for f_n. From an ordering of $2n$ people, we form a matching by pairing the first two, the next two, and so on. Each ordering thus yields one matching. Each matching is generated by $2^n n!$ orderings, since change the order of the pairs or the order within a pair does not change the resulting matching. Thus there are $f_n = (2n)!/(2^n n!)$ perfect matchings. ■

The usual drawing of the Petersen graph shows a perfect matching and two 5-cycles; counting the perfect matchings takes some effort (Exercise 14). The inductive construction of the hypercube Q_k readily yields many perfect matchings (Exercise 16), but counting them exactly is difficult. The graphs below have even order but no perfect matchings.

MAXIMUM MATCHINGS

A matching is a set of edges, so its **size** is the number of edges. We can seek a large matching by iteratively selecting edges whose endpoints are not used by the edges already selected, until no more are available. This yields a maximal matching but maybe not a maximum matching.

3.1.4. Definition. A **maximal matching** in a graph is a matching that cannot be enlarged by adding an edge. A **maximum matching** is a matching of maximum size among all matchings in the graph.

A matching M is maximal if every edge not in M is incident to an edge already in M. Every maximum matching is a maximal matching, but the converse need not hold.

3.1.5. Example. *Maximal $\neq$ maximum.* The smallest graph having a maximal matching that is not a maximum matching is P_4. If we take the middle edge, then we can add no other, but the two end edges form a larger matching. Below we show this phenomenon in P_4 and in P_6. ■

In Example 3.1.5, replacing the bold edges by the solid edges yields a larger matching. This gives us a way to look for larger matchings.

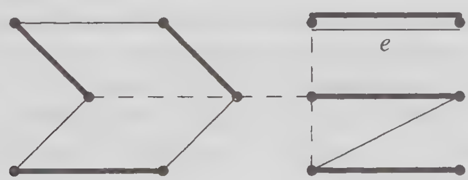
3.1.6. Definition. Given a matching M, an M-**alternating path** is a path that alternates between edges in M and edges not in M. An M-alternating path whose enpoints are unsaturated by M is an M-**augmenting path**.

Given an M-augmenting path P, we can replace the edges of M in P with the other edges of P to obtain a new matching M' with one more edge. Thus when M is a maximum matching, there is no M-augmenting path.

In fact, we prove next that maximum matchings are characterized by the absence of augmenting paths. We prove this by considering two matchings and examining the set of edges belonging to exactly one of them. We define this operation for any two graphs with the same vertex set. (The operation is defined in general for any two sets; see Appendix A.)

3.1.7. Definition. If G and H are graphs with vertex set V, then the **symmetric difference** $G \triangle H$ is the graph with vertex set V whose edges are all those edges appearing in exactly one of G and H. We also use this notation for sets of edges; in particular, if M and M' are matchings, then $M \triangle M' = (M - M') \cup (M' - M)$.

3.1.8. Example. In the graph below, M is the matching with five solid edges, M' is the one with six bold edges, and the dashed edges belong to neither M nor M'. The two matchings have one common edge e; it is not in their symmetric difference. The edges of $M \triangle M'$ form a cycle of length 6 and a path of length 3. ■



3.1.9. Lemma. Every component of the symmetric difference of two matchings is a path or an even cycle.

Proof: Let M and M' be matchings, and let $F = M \triangle M'$. Since M and M' are matchings, every vertex has at most one incident edge from each of them. Thus F has at most two edges at each vertex. Since $\Delta(F) \leq 2$, every component of F is a path or a cycle. Furthermore, every path or cycle in F alternates between edges of $M - M'$ and edges of $M' - M$. Thus each cycle has even length, with an equal number of edges from M and from M'. ■

3.1.10. Theorem. (Berge [1957]) A matching M in a graph G is a maximum matching in G if and only if G has no M-augmenting path.

Proof: We prove the contrapositive of each direction; G has a matching larger than M if and only if G has an M-augmenting path. We have observed that an M-augmenting path can be used to produce a matching larger than M.

For the converse, let M' be a matching in G larger than M; we construct an M-augmenting path. Let $F = M \triangle M'$. By Lemma 3.1.9, F consists of paths and even cycles; the cycles have the same number of edges from M and M'. Since $|M'| > |M|$, F must have a component with more edges of M' than of M. Such a component can only be a path that starts and ends with an edge of M'; thus it is an M-augmenting path in G. ■

HALL'S MATCHING CONDITION

When we are filling jobs with applicants, there may be many more applicants than jobs; successfully filling the jobs will not use all applicants. To model this problem, we consider an X, Y-bigraph (bipartite graph with bipartition X, Y—Definition 1.2.17), and we seek a matching that saturates X.

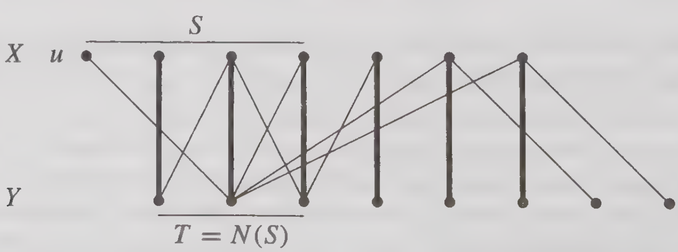
If a matching M saturates X, then for every $S \subseteq X$ there must be at least $|S|$ vertices that have neighbors in S, because the vertices matched to S must be chosen from that set. We use $N_G(S)$ or simply $N(S)$ to denote the set of vertices having a neighbor in S. Thus $|N(S)| \geq |S|$ is a necessary condition.

The condition "For all $S \subseteq X$, $|N(S)| \geq |S|$" is **Hall's Condition**. Hall proved that this obvious necessary condition is also sufficient (TONCAS).

3.1.11. Theorem. (Hall's Theorem—P. Hall [1935]) An X, Y-bigraph G has a matching that saturates X if and only if $|N(S)| \geq |S|$ for all $S \subseteq X$.

Proof: *Necessity*. The $|S|$ vertices matched to S must lie in $N(S)$.

Sufficiency. To prove that Hall's Condition is sufficient, we prove the contrapositive. If M is a maximum matching in G and M does not saturate X, then we obtain a set $S \subseteq X$ such that $|N(S)| < |S|$. Let $u \in X$ be a vertex unsaturated by M. Among all the vertices reachable from u by M-alternating paths in G, let S consist of those in X, and let T consist of those in Y (see figure below with M in bold). Note that $u \in S$.



We claim that M matches T with $S - \{u\}$. The M-alternating paths from u reach Y along edges not in M and return to X along edges in M. Hence every vertex of $S - \{u\}$ is reached by an edge in M from a vertex in T. Since there is no M-augmenting path, every vertex of T is saturated; thus an M-alternating path reaching $y \in T$ extends via M to a vertex of S. Hence these edges of M yield a bijection from T to $S - \{u\}$, and we have $|T| = |S - \{u\}|$.

The matching between T and $S - \{u\}$ yields $T \subseteq N(S)$. In fact, $T = N(S)$. Suppose that $y \in Y - T$ has a neighbor $v \in S$. The edge vy cannot be in M, since u is unsaturated and the rest of S is matched to T by M. Thus adding vy to an M-alternating path reaching v yields an M-alternating path to y. This contradicts $y \notin T$, and hence vy cannot exist.

With $T = N(S)$, we have proved that $|N(S)| = |T| = |S| - 1 < |S|$ for this choice of S. This completes the proof of the contrapositive. ■

One can also prove sufficiency by assuming Hall's Condition, supposing that no matching saturates X, and obtaining a contradiction. As we have seen, lack of a matching saturating X yields a violation of Hall's Condition. Contradicting the hypothesis usually means that the contrapositive of the desired implication has been proved. Thus we have stated the proof in that language.

3.1.12. Remark. Theorem 3.1.11 implies that whenever an X, Y-bigraph has no matching saturating X, we can verify this by exhibiting a subset of X with too few neighbors.

Note also that the statement and proof permit multiple edges. ■

Many proofs of Hall's Theorem have been published; see Mirsky [1971, p38] and Jacobs [1969] for summaries. A proof by M. Hall [1948] leads to a lower bound on the number of matchings that saturate X, as a function of the vertex degrees. We consider algorithmic aspects in Section 3.2.

When the sets of the bipartition have the same size, Hall's Theorem is the **Marriage Theorem**, proved originally by Frobenius [1917]. The name arises from the setting of the compatibility relation between a set of n men and a set of n women. If every man is compatible with k women and every woman is compatible with k men, then a perfect matching must exist. Again multiple edges are allowed, which enlarges the scope of applications (see Theorem 3.3.9 and Theorem 7.1.7, for example).

3.1.13. Corollary. For $k > 0$, every k-regular bipartite graph has a perfect matching.

Proof: Let G be a k-regular X, Y-bigraph. Counting the edges by endpoints in X and by endpoints in Y shows that $k\,|X| = k\,|Y|$, so $|X| = |Y|$. Hence it suffices to verify Hall's Condition; a matching that saturates X will also saturate Y and be a perfect matching.

Consider $S \subseteq X$. Let m be the number of edges from S to $N(S)$. Since G is k-regular, $m = k\,|S|$. These m edges are incident to $N(S)$, so $m \le k\,|N(S)|$. Hence $k\,|S| \le k\,|N(S)|$, which yields $|N(S)| \ge |S|$ when $k > 0$. Having chosen $S \subseteq X$ arbitrarily, we have established Hall's condition. ■

One can also use contradiction here. Assuming that G has no perfect matching yields a set $S \subseteq X$ such that $|N(S)| < |S|$. The argument obtaining a contradiction amounts to a rewording of the direct proof given above.

MIN-MAX THEOREMS

When a graph G does not have a perfect matching, Theorem 3.1.10 allows us to prove that M is a maximum matching by proving that G has no M-augmenting path. Exploring all M-alternating paths to eliminate the possibility of augmentation could take a long time.

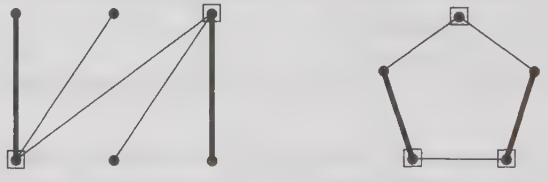
We faced a similar situation when proving that a graph is not bipartite. Instead of checking all possible bipartitions, we can exhibit an odd cycle. Here again, instead of exploring all M-alternating paths, we would prefer to exhibit an explicit structure in G that forbids a matching larger than M.

3.1.14. Definition. A **vertex cover** of a graph G is a set $Q \subseteq V(G)$ that contains at least one endpoint of every edge. The vertices in Q *cover* $E(G)$.

In a graph that represents a road network (with straight roads and no isolated vertices), we can interpret the problem of finding a minimum vertex cover as the problem of placing the minimum number of policemen to guard the entire road network. Thus "cover" means "watch" in this context.

Since no vertex can cover two edges of a matching, the size of every vertex cover is at least the size of every matching. Therefore, obtaining a matching and a vertex cover of the same size PROVES that each is optimal. Such proofs exist for bipartite graphs, but not for all graphs.

3.1.15. Example. *Matchings and vertex covers.* In the graph on the left below we mark a vertex cover of size 2 and show a matching of size 2 in bold. The vertex cover of size 2 prohibits matchings with more than 2 edges, and the matching of size 2 prohibits vertex covers with fewer than 2 vertices. As illustrated on the right, the optimal values differ by 1 for an odd cycle. The difference can be arbitrarily large (Exercise 3.3.10). ■



3.1.16. Theorem. (König [1931], Egerváry [1931]) If G is a bipartite graph, then the maximum size of a matching in G equals the minimum size of a vertex cover of G.

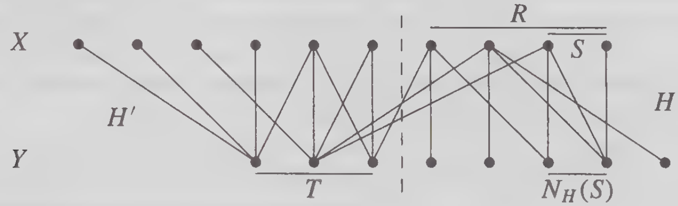
Proof: Let G be an X, Y-bigraph. Since distinct vertices must be used to cover the edges of a matching, $|Q| \geq |M|$ whenever Q is a vertex cover and M is a matching in G. Given a smallest vertex cover Q of G, we construct a matching of size $|Q|$ to prove that equality can always be achieved.

Partition Q by letting $R = Q \cap X$ and $T = Q \cap Y$. Let H and H' be the subgraphs of G induced by $R \cup (Y - T)$ and $T \cup (X - R)$, respectively. We use

Hall's Theorem to show that H has a matching that saturates R into $Y - T$ and H' has a matching that saturates T. Since H and H' are disjoint, the two matchings together form a matching of size $|Q|$ in G.

Since $R \cup T$ is a vertex cover, G has no edge from $Y - T$ to $X - R$. For each $S \subseteq R$, we consider $N_H(S)$, which is contained in $Y - T$. If $|N_H(S)| < |S|$, then we can substitute $N_H(S)$ for S in Q to obtain a smaller vertex cover, since $N_H(S)$ covers all edges incident to S that are not covered by T.

The minimality of Q thus yields Hall's Condition in H, and hence H has a matching that saturates R. Applying the same argument to H' yields the matching that saturates T. ∎



As graph theory continues to develop, new proofs of fundamental results like the König–Egerváry Theorem appear; see Rizzo [2000].

3.1.17. Remark. A **min-max relation** is a theorem stating equality between the answers to a minimization problem and a maximization problem over a class of instances. The König–Egerváry Theorem is such a relation for vertex covering and matching in bipartite graphs.

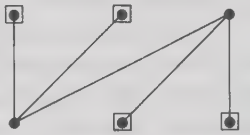
For the discussions in this text, we think of a **dual pair** of optimization problems as a maximization problem **M** and a minimization problem **N**, defined on the same instances (such as graphs), such that for every candidate solution M to **M** and every candidate solution N to **N**, the value of M is less than or equal to the value of N. Often the "value" is cardinality, as above where **M** is maximum matching and **N** is minimum vertex cover.

When **M** and **N** are dual problems, obtaining candidate solutions M and N that have the same value PROVES that M and N are optimal solutions for that instance. We will see many pairs of dual problems in this book. A min-max relation states that, on some class of instances, these short proofs of optimality exist. These theorems are desirable because they save work! Our next objective is another such theorem for independent sets in bipartite graphs. ∎

INDEPENDENT SETS AND COVERS

We now turn from matchings to independent sets. The **independence number** of a graph is the maximum size of an independent set of vertices.

3.1.18. Example. The independence number of a bipartite graph does *not* always equal the size of a partite set. In the graph below, both partite sets have size 3, but we have marked an independent set of size 4. ■



No vertex covers two edges of a matching. Similarly, no edge contains two vertices of an independent set. This yields another dual covering problem.

3.1.19. Definition. An **edge cover** of G is a set L of edges such that every vertex of G is incident to some edge of L.

We say that the vertices of G are *covered* by the edges of L. In Example 3.1.18, the four edges incident to the marked vertices form an edge cover; the remaining two vertices are covered "for free".

Only graphs without isolated vertices have edge covers. A perfect matching forms an edge cover with $n(G)/2$ edges. In general, we can obtain an edge cover by adding edges to a maximum matching.

3.1.20. Definition. For the optimal sizes of the sets in the independence and covering problems we have defined, we use the notation below.

maximum size of independent set	$\alpha(G)$
maximum size of matching	$\alpha'(G)$
minimum size of vertex cover	$\beta(G)$
minimum size of edge cover	$\beta'(G)$

A graph may have many independent sets of maximum size (C_5 has five of them), but the independence number $\alpha(G)$ is a single integer ($\alpha(C_5) = 2$). The notation treats the numbers that answer these optimization problems as graph parameters, like the order, size, maximum degree, diameter, etc. Our use of $\alpha'(G)$ to count the edges in a maximum matching suggests a relationship with the parameter $\alpha(G)$ that counts the vertices in a maximum independent set. We explore this relationship in Section 7.1.

We use $\beta(G)$ for minimum vertex cover due to its interaction with maximum matching. The "prime" goes on $\beta'(G)$ rather than on $\beta(G)$ because $\beta(G)$ counts a set of vertices and $\beta'(G)$ counts a set of edges.

In this notation, the König–Egerváry Theorem states that $\alpha'(G) = \beta(G)$ for every bipartite graph G. We will prove that also $\alpha(G) = \beta'(G)$ for bipartite graphs without isolated vertices. Since no edge can cover two vertices of an independent set, the inequality $\beta'(G) \geq \alpha(G)$ is immediate. (When $S \subseteq V(G)$, we often use $\overline{S}$ to denote $V(G) - S$, the remaining vertices).

3.1.21. Lemma. In a graph G, $S \subseteq V(G)$ is an independent set if and only if $\overline{S}$ is a vertex cover, and hence $\alpha(G) + \beta(G) = n(G)$.

Proof: If S is an independent set, then every edge is incident to at least one vertex of $\overline{S}$. Conversely, if $\overline{S}$ covers all the edges, then there are no edges joining vertices of S. Hence every maximum independent set is the complement of a minimum vertex cover, and $\alpha(G) + \beta(G) = n(G)$. ■

The relationship between matchings and edge coverings is more subtle. Nevertheless, a similar formula holds.

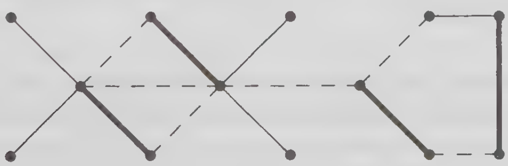
3.1.22. Theorem. (Gallai [1959]) If G is a graph without isolated vertices, then $\alpha'(G) + \beta'(G) = n(G)$.

Proof: From a maximum matching M, we will construct an edge cover of size $n(G) - |M|$. Since a smallest edge cover is no bigger than this cover, this will imply that $\beta'(G) \le n(G) - \alpha'(G)$. Also, from a minimum edge cover L, we will construct a matching of size $n(G) - |L|$. Since a largest matching is no smaller than this matching, this will imply that $\alpha'(G) \ge n(G) - \beta'(G)$. These two inequalities complete the proof.

Let M be a maximum matching in G. We construct an edge cover of G by adding to M one edge incident to each unsaturated vertex. We have used one edge for each vertex, except that each edge of M takes care of two vertices, so the total size of this edge cover is $n(G) - |M|$, as desired.

Now let L be a minimum edge cover. If both endpoints of an edge e belong to edges in L other than e, then $e \notin L$, since $L - \{e\}$ is also an edge cover. Hence each component formed by edges of L has at most one vertex of degree exceeding 1 and is a star (a tree with at most one non-leaf). Let k be the number of these components. Since L has one edge for each non-central vertex in each star, we have $|L| = n(G) - k$. We form a matching M of size $k = n(G) - |L|$ by choosing one edge from each star in L. ■

3.1.23. Example. The graph below has 13 vertices. A matching of size 4 appears in bold, and adding the solid edges yields an edge cover of size 9. The dashed edges are not needed in the cover. The edge cover consists of four stars; from each we extract one edge (bold) to form the matching. ■



3.1.24. Corollary. (König [1916]) If G is a bipartite graph with no isolated vertices, then $\alpha(G) = \beta'(G)$.

Proof: By Lemma 3.1.21 and Theorem 3.1.22, $\alpha(G) + \beta(G) = \alpha'(G) + \beta'(G)$. Subtracting the König–Egerváry relation $\alpha'(G) = \beta(G)$ completes the proof. ■

DOMINATING SETS (optional)

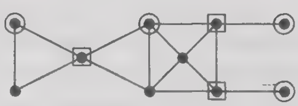
The edges covered by one vertex in a vertex cover are the edges incident to it; they form a star. The vertex cover problem can be described as covering the edge set with the fewest stars. Sometimes we instead want to cover the vertex set with fewest stars. This is equivalent to our next graph parameter.

3.1.25. Example. A company wants to establish transmission towers in a remote region. The towers are located at inhabited buildings, and each inhabited building must be reachable. If a transitter at x can reach y, then also one at y can reach x. Given the pairs that can reach each other, how many transmitters are needed to cover all the buildings?

A similar problem comes from recreational mathematics: How many queens are needed to attack all squares on a chessboard? (Exercise 56). ■

3.1.26. Definition. In a graph G, a set $S \subseteq V(G)$ is a **dominating set** if every vertex not in S has a neighbor in S. The **domination number** $\gamma(G)$ is the minimum size of a dominating set in G.

3.1.27. Example. The graph G below has a minimal dominating set of size 4 (circles) and a minimum dominating set of size 3 (squares): $\gamma(G) = 3$. ■



Berge [1962] introduced the notion of domination. Ore [1962] coined this terminology, and the notation $\gamma(G)$ appeared in an early survey (Cockayne–Hedetniemi [1977]). An entire book (Haynes–Hedetniemi–Slater [1998]) is devoted to domination and its variations.

3.1.28. Example. Covering the vertex set with stars may not require as many stars as covering the edge set. When a graph G has no isolated vertices, every vertex cover *is* a dominating set, so $\gamma(G) \leq \beta(G)$. The difference can be large; $\gamma(K_n) = 1$, but $\beta(K_n) = n - 1$. ■

When studying domination as an extremal problem, we try to obtain bounds in terms of other graph parameters, such as the order and the minimum degree. A vertex of degree k dominates itself and k other vertices; thus every dominating set in a k-regular graph G has size at least $n(G)/(k+1)$. For every graph with minimum degree k, a greedy algorithm produces a dominating set not too much bigger than this.

3.1.29. Definition. The **closed neighborhood** $N[v]$ of a vertex v in a graph is $N(v) \cup \{v\}$; it is the set of vertices *dominated by* v.

3.1.30. Theorem. (Arnautov [1974], Payan [1975]) Every n-vertex graph with minimum degree k has a dominating set of size at most $n\frac{1+\ln(k+1)}{k+1}$.

Proof: (Alon [1990]) Let G be a graph with minimum degree k. Given $S \subseteq V(G)$, let U be the set of vertices not dominated by S. We claim that some vertex y outside S dominates at least $|U|(k+1)/n$ vertices of U. Each vertex in U has at least k neighbors, so $\sum_{v\in U}|N[v]| \ge |U|(k+1)$. Each vertex of G is counted at most n times by these $|U|$ sets, so some vertex y appears at least $|U|(k+1)/n$ times and satisfies the claim.

We iteratively select a vertex that dominates the most of the remaining undominated vertices. We have proved that when r undominated vertices remain, after the next selection at most $r(1-(k+1)/n)$ undominated vertices remain. Hence after $n\frac{\ln(k+1)}{k+1}$ steps the number of undominated vertices is at most

$$n(1-\tfrac{k+1}{n})^{n\ln(k+1)/(k+1)} < ne^{-\ln(k+1)} = \tfrac{n}{k+1}$$

The selected vertices and these remaining undominated vertices together form a dominating set of size at most $n\frac{1+\ln(k+1)}{k+1}$. ■

3.1.31. Remark. This bound is also proved by probabilistic methods in Theorem 8.5.10. Caro–Yuster–West [2000] showed that for large k the total domination number satisfies a bound asymptotic to this. Alon [1990] used probabilistic methods to show that this bound is asymptotically sharp when k is large.

Exact bounds remain of interest for small k. Among connected n-vertex graphs, $\delta(G) \ge 2$ implies $\gamma(G) \le 2n/5$ (McCuaig–Shepherd [1989], with seven small exceptions), and $\delta(G) \ge 3$ implies $\gamma(G) \le 3n/8$ (Reed [1996]). Exercise 53 requests constructions achieving these bounds. ■

Many variations on the concept of domination are studied. In Example 3.1.25, for example, one might want the transmitters to be able to communicate with each other, which requires that they induce a connected subgraph.

3.1.32. Definition. A dominating set S in G is
a **connected dominating set** if $G[S]$ is connected,
an **independent dominating set** if $G[S]$ is independent, and
a **total dominating set** if $G[S]$ has no isolated vertex.

Each variation adds a constraint, so dominating sets of these types are at least as large as $\gamma(G)$. Exercises 54–60 explore these variations. Studying independent dominating sets amounts to studying maximal independent sets. This leads to a nice result about claw-free graphs.

3.1.33. Lemma. A set of vertices in a graph is an independent dominating set if and only if it is a maximal independent set.

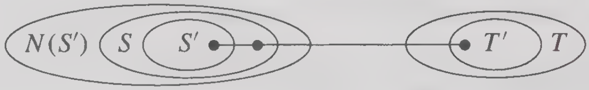
Proof: Among independent sets, S is maximal if and only if every vertex outside S has a neighbor in S, which is the condition for S to be a dominating set. ■

3.1.34. Theorem. (Bollobás–Cockayne [1979]) Every claw-free graph has an independent dominating set of size $\gamma(G)$.

Proof: Let S be a minimum dominating set in a claw-free graph G. Let S' be a maximal independent subset of S. Let $T = V(G) - N(S')$. Let T' be a maximal independent subset of S.

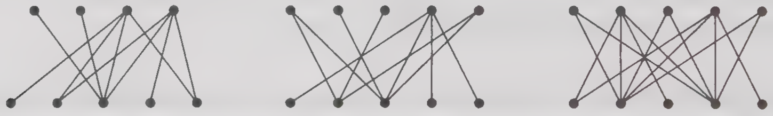
Since T' contains no neighbor of S', $S' \cup T'$ is independent. Since S' is maximal in S, we have $S \subseteq N(S')$. Since T' is maximal in T, T' dominates T. Hence $S' \cup T'$ is a dominating set.

It remains to show that $|S' \cup T'| \le \gamma(G)$. Since S' is maximal in S, T' is independent, and G is claw-free, each vertex of $S - S'$ has at most one neighbor in T'. Since S is dominating, each vertex of T' has at least one neighbor in $S - S'$. Hence $|T'| \le |S - S'|$, which yields $|S' \cup T'| \le |S| = \gamma(G)$. ■



EXERCISES

3.1.1. (−) Find a maximum matching in each graph below. Prove that it is a maximum matching by exhibiting an optimal solution to the dual problem (minimum vertex cover). Explain why this proves that the matching is optimal.



3.1.2. (−) Determine the minimum size of a maximal matching in the cycle C_n.

3.1.3. (−) Let S be the set of vertices saturated by a matching M in a graph G. Prove that some maximum matching also saturates all of S. Must the statement be true for every maximum matching?

3.1.4. (−) For each of $\alpha, \alpha', \beta, \beta'$, characterize the simple graphs for which the value of the parameter is 1.

3.1.5. (−) Prove that $\alpha(G) \ge \frac{n(G)}{\Delta(G)+1}$ for every graph G.

3.1.6. (−) Let T be a tree with n vertices, and let k be the maximum size of an independent set in T. Determine $\alpha'(T)$ in terms of n and k.

3.1.7. (−) Use Corollary 3.1.24 to prove that a graph G is bipartite if and only if $\alpha(H) = \beta'(H)$ for every subgraph H of G with no isolated vertices.

• • • • •

3.1.8. (!) Prove or disprove: Every tree has at most one perfect matching.

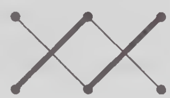
3.1.9. (!) Prove that every maximal matching in a graph G has at least $\alpha'(G)/2$ edges.

3.1.10. Let M and N be matchings in a graph G, with $|M| > |N|$. Prove that there exist matchings M' and N' in G such that $|M'| = |M| - 1$, $|N'| = |N| + 1$, and M', N' have the same union and intersection (as edge sets) as M, N.

3.1.11. Let C and C' be cycles in a graph G. Prove that $C \triangle C'$ decomposes into cycles.

3.1.12. Let C and C' be cycles of length k in a graph with girth k. Prove that $C \triangle C'$ is a single cycle if and only if $C \cap C'$ is a single path. (Jiang [2001])

3.1.13. Let M and M' be matchings in an X, Y-bigraph G. Suppose that M saturates $S \subseteq X$ and that M' saturates $T \subseteq Y$. Prove that G has a matching that saturates $S \cup T$. For example, below we show M as bold edges and M' as thin edges; we can saturate $S \cup T$ by using one edge from each.



3.1.14. Let G be the Petersen graph. In Example 7.1.9, analysis by cases is used to show that if M is a perfect matching in G, then $G - M = C_5 + C_5$. Assume this.

a) Prove that every edge of G lies in four 5-cycles, and count the 5-cycles in G.

b) Determine the number of perfect matchings in G.

3.1.15. a) Prove that for every perfect matching M in Q_k and every coordinate $i \in [k]$, there are an even number of edges in M whose endpoints differ in coordinate i.

b) Use part (a) to count the perfect matchings in Q_3.

3.1.16. For $k \geq 2$, prove that Q_k has at least $2^{(2^{k-2})}$ perfect matchings.

3.1.17. The *weight* of a vertex in Q_k is the number of 1s in its label. Prove that for every perfect matching in Q_k, the number of edges matching words of weight i to words of weight $i+1$ is $\binom{k-1}{i}$, for $0 \leq i \leq k-1$.

3.1.18. (!) Two people play a game on a graph G, alternately choosing distinct vertices. Player 1 start by choosing any vertex. Each subsequent choice must be adjacent to the preceding choice (of the other player). Thus together they follow a path. The last player able to move wins.

Prove that the second player has a winning strategy if G has a perfect matching, and otherwise the first player has a winning strategy. (Hint: For the second part, the first player should start with a vertex omitted by some maximum matching.)

3.1.19. (!) Let $\mathbf{A} = (A_1, \ldots, A_m)$ be a collection of subsets of a set Y. A **system of distinct representatives** (SDR) for $\mathbf{A}$ is a set of distinct elements $a_1, \ldots, a_m$ in Y such that $a_i \in A_i$. Prove that $\mathbf{A}$ has an SDR if and only if $|\bigcup_{i \in S} A_i| \geq |S|$ for every $S \subseteq \{1, \ldots, m\}$. (Hint: Transform this to a graph problem.)

3.1.20. The people in a club are planning their summer vacations. Trips $t_1, \ldots, t_n$ are available, but trip t_i has capacity n_i. Each person likes some of the trips and will travel on at most one. In terms of which people like which trips, derive a necessary and sufficient condition for being able to fill all trips (to capacity) with people who like them.

3.1.21. (!) Let G be an X, Y-bigraph such that $|N(S)| > |S|$ whenever $\emptyset \neq S \subset X$. Prove that every edge of G belongs to some matching that saturates X.

3.1.22. Prove that a bipartite graph G has a perfect matching if and only if $|N(S)| \geq |S|$ for all $S \subseteq V(G)$, and present an infinite class of examples to prove that this characterization does not hold for all graphs.

3.1.23. (+) *Alternative proof of Hall's Theorem.* Consider a bipartite graph G with bipartition X, Y, satisfying $|N(S)| \geq |S|$ for every $S \subseteq X$. Use induction on $|X|$ to prove that G has a matching that saturates X. (Hint: First consider the case where $|N(S)| > |S|$ for every proper subset S of X. When this does not hold, consider a minimal nonempty $T \subseteq X$ such that $|N(T)| = |T|$.) (M. Hall [1948], Halmos–Vaughan [1950])

3.1.24. (!) A **permutation matrix** P is a 0,1-matrix having exactly one 1 in each row and column. Prove that a square matrix of nonnegative integers can be expressed as the sum of k permutation matrices if and only if all row sums and column sums equal k.

3.1.25. (!) A **doubly stochastic matrix** Q is a nonnegative real matrix in which every row and every column sums to 1. Prove that a doubly stochastic matrix Q can be expressed $Q = c_1 P_1 + \cdots + c_m P_m$, where $c_1, \ldots, c_m$ are nonnegative real numbers summing to 1 and $P_1, \ldots, P_m$ are permutation matrices. For example,

$$\begin{pmatrix} 1/2 & 1/3 & 1/6 \\ 0 & 1/6 & 5/6 \\ 1/2 & 1/2 & 0 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} + \frac{1}{3}\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} + \frac{1}{6}\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

(Hint: Use induction on the number of nonzero entries in Q.) (Birkhoff [1946], von Neumann [1953])

3.1.26. (!) A deck of mn cards with m values and n suits consists of one card of each value in each suit. The cards are dealt into an n-by-m array.

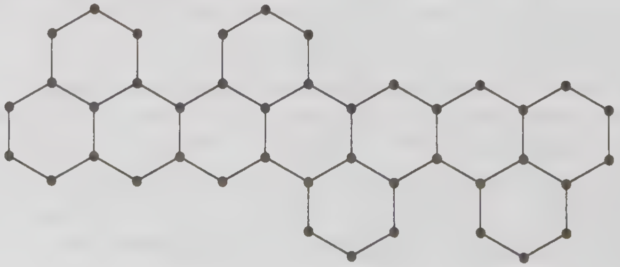
a) Prove that there is a set of m cards, one in each column, having distinct values.

b) Use part (a) to prove that by a sequence of exchanges of cards of the same value, the cards can be rearranged so that each column consists of n cards of distinct suits. (Enchev [1997])

3.1.27. (!) *Generalizing Tic-Tac-Toe.* A **positional game** consists of a set $X = x_1, \ldots, x_n$ of positions and a family $W_1, \ldots, W_m$ of winning sets of positions (Tic-Tac-Toe has nine positions and eight winning sets). Two players alternately choose positions; a player wins by collecting a winning set.

Suppose that each winning set has size at least a and each position appears in at most b winning sets (in Tic-Tac-Toe, $a = 3$ and $b = 4$). Prove that Player 2 can force a draw if $a \geq 2b$. (Hint: Form an X, Y-bigraph G, where $Y = \{w_1, \ldots, w_m\} \cup \{w'_1, \ldots, w'_m\}$, with edges $x_i w_j$ and $x_i w'_j$ whenever $x_i \in W_j$. How can Player 2 use a matching in G? Comment: This result implies that Player 2 can force a draw in d-dimensional Tic-Tac-Toe when the sides are long enough.)

3.1.28. (!) Exhibit a perfect matching in the graph below or give a short proof that it has none. (Lovász–Plummer [1986, p7])



3.1.29. (!) Use the König–Egerváry Theorem to prove that every bipartite graph G has a matching of size at least $e(G)/\Delta(G)$. Use this to conclude that every subgraph of $K_{n,n}$ with more than $(k-1)n$ edges has a matching of size at least k.

3.1.30. (!) Determine the maximum number of edges in a simple bipartite graph that contains no matching with k edges and no star with l edges. (Isaak)

3.1.31. Use the König–Egerváry Theorem to prove Hall's Theorem.

3.1.32. (!) In an X,Y-bigraph G, the **deficiency** of a set S is $\mathrm{def}(S) = |S| - |N(S)|$; note that $\mathrm{def}(\emptyset) = 0$. Prove that $\alpha'(G) = |X| - \max_{S \subseteq X} \mathrm{def}(S)$. (Hint: Form a bipartite graph G' such that G' has a matching that saturates X if and only if G has a matching of the desired size, and prove that G' satisfies Hall's Condition.) (Ore [1955])

3.1.33. (!) Use Exercise 3.1.32 to prove the König–Egerváry Theorem. (Hint: Obtain a matching and a vertex cover of the same size from a set with maximum deficiency.)

3.1.34. (!) Let G be an X,Y-bigraph with no isolated vertices, and define *deficiency* as in Exercise 3.1.32. Prove that Hall's Condition holds for a matching saturating X if and only if each subset of Y has deficiency at most $|Y| - |X|$.

3.1.35. Let G be an X,Y-bigraph. Prove that G is $(k+1)K_2$-free if and only if each $S \subseteq X$ has a subset of size at most k with neighborhood $N(S)$. (Liu–Zhou [1997])

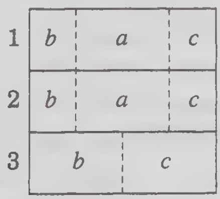
3.1.36. Let G be an X,Y-bigraph having a matching that saturates X. Letting $m = |X|$, prove that G has at most $\binom{m}{2}$ edges belonging to no matching of size m. Construct examples to show that this is best possible for every m.

3.1.37. (+) Let G be an X,Y-bigraph having a matching that saturates X.

a) Let S and T be subsets of X such that $|N(S)| = |S|$ and $|N(T)| = |T|$. Prove that $|N(S \cap T)| = |S \cap T|$.

b) Prove that X has some vertex x such that every edge incident to x belongs to some maximum matching. (Hint: Consider a minimal nonempty set $S \subseteq X$ such that $|N(S)| = |S|$, if any exists.)

3.1.38. (+) An island of area n has n married hunter/farmer couples. The Ministry of Hunting divides the island into n equal-sized hunting regions. The Ministry of Agriculture divides it into n equal-sized farming regions. The Ministry of Marriage requires that each couple receive two overlapping regions. By Exercise 3.1.25, this is always possible. Prove a stronger result: guarantee a pairing where each couple's two regions share area at least $4/(n+1)^2$ when n is odd and $4/[n(n+2)]$ when n is even. Prove also that no larger common area can be guaranteed; the example below achieves equality for $n=3$. (Marcus–Ree [1959], Floyd [1990])



$$\begin{pmatrix} .5 & .25 & .25 \\ .5 & .25 & .25 \\ 0 & .5 & .5 \end{pmatrix}$$

3.1.39. Let G be a nontrivial simple graph. Prove that $\alpha(G) \le n(G) - e(G)/\Delta(G)$. Conclude that $\alpha(G) \le n(G)/2$ when G also is regular. (P. Kwok)

3.1.40. Let G be a bipartite graph. Prove that $\alpha(G) = n(G)/2$ if and only if G has a perfect matching.

3.1.41. A connected n-vertex graph has exactly one cycle if and only if it has exactly n edges (Exercise 2.1.30). Let C be the cycle in such a graph G. Assuming the result of Exercise 3.1.40 for trees, prove that $\alpha(G) \geq \lfloor n(G)/2 \rfloor$, with equality if and only if $G - V(C)$ has a perfect matching.

3.1.42. (!) An algorithm to greedily build a large independent set iteratively selects a vertex of minimum degree in the remaining graph and deletes it and its neighbors. Prove that this algorithm produces an independent set of size at least $\sum_{v \in V(G)} \frac{1}{d(v)+1}$ in a graph G. (Caro [1979], Wei [1981])

3.1.43. Let M be a maximal matching and L a minimal edge cover in a graph with no isolated vertices. Prove the statements below. (Norman–Rabin [1959], Gallai [1959])

a) M is a maximum matching if and only if M is contained in a minimum edge cover.

b) L is a minimum edge cover if and only if L contains a maximum matching.

3.1.44. (−) Let G be a simple graph in which the sum of the degrees of any k vertices is less than $n - k$. Prove that every maximal independent set in G has more than k vertices. (Meyer [1972])

3.1.45. An edge e of a graph G is **α-critical** if $\alpha(G - e) > \alpha(G)$. Suppose that xy and xz are α-critical edges in G. Prove that G has an induced subgraph that is an odd cycle containing xy and xz. (Hint: Let Y, Z be maximum independent sets in $G - xy$ and $G - xz$, respectively. Let $H = G[Y \triangle Z]$. Prove that every component of H has the same number of vertices from Y and from Z. Use this to prove that y and z belong to the same component of H.) (Berge [1970], with a difficult generalization in Markossian–Karapetian [1984])

3.1.46. (∗−) Characterize the graphs with domination number 1.

3.1.47. (∗−) Find the smallest tree where the domination number and the vertex cover number are not equal.

3.1.48. (∗−) Determine $\gamma(C_n)$ and $\gamma(P_n)$.

3.1.49. (∗) Let G be a graph without isolated vertices, and let S be a minimal dominating set in G. Prove that $\overline{S}$ is a dominating set. Conclude that $\gamma(G) \leq n(G)/2$. (Ore [1962])

3.1.50. (∗) Prove that $\gamma(G) \leq n - \beta'(G) \leq n/2$ when G is an n-vertex graph without isolated vertices. For $1 \leq k \leq n/2$, construct a connected n-vertex graph G with $\gamma(G) = k$.

3.1.51. (∗) Let G be an n-vertex graph.

a) Prove that $\lceil n/(1 + \Delta(G) \rceil \geq \gamma(G) \leq n - \Delta(G)$.

b) Prove that $(1 + \text{diam}\, G)/3 \leq \gamma(G) \leq n - \lfloor \text{diam}\, G/3 \rfloor$.

3.1.52. (∗) Prove that if the diameter of G is at least 3, then $\gamma(\overline{G}) \leq 2$.

3.1.53. (∗) For all $k \in \mathbb{N}$, construct a connected graph with $5k$ vertices and domination number $2k$. Construct a single 3-regular graph G such that $\gamma(G) = 3n(G)/8$.

3.1.54. (∗) Determine the domination number of the Petersen graph, and determine the minimum size of a total dominating set in the Petersen graph.

3.1.55. (∗) In the hypercube Q_4, determine the minimum sizes of a dominating set, an independent dominating set, a connected dominating set, and a total dominating set.

3.1.56. (∗) Find a way to place five queens on an eight-by-eight chessboard that attack all other squares. Show that the five queens cannot be placed so that also they do not attack each other. (Comment: Thus the independent domination number of the "queen's graph" exceeds its domination number; it is 7.)

3.1.57. (∗) For all $n \in \mathbb{N}$, construct an n-vertex tree with domination number 2 in which the minimum size of an independent dominating set is $\lfloor n/2 \rfloor$.

3.1.58. (∗) Prove that a $K_{1,r}$-free graph G has an independent dominating set of size at most $(r-2)\gamma(G) - (r-3)$. (Hint: Generalize the argument of Theorem 3.1.34.) (Bollobás–Cockayne [1979])

3.1.59. (∗) In a graph G of order n, prove that the minimum size of a connected dominating set is n minus the maximum number of leaves in a spanning tree.

3.1.60. (∗) For $k \le 5$, every graph G with $\delta(G) \le k$ has a connected dominating set of size at most $3n(G)/(k+1)$ (Kleitman–West [1991], Griggs–Wu [1992]). Prove that this is sharp using a graph formed from a cyclic arrangement of $3m$ pairwise-disjoint cliques by making each vertex adjacent to every vertex in the clique before it and the clique after it. Let the clique sizes be $\lceil k/2 \rceil$, $\lfloor k/2 \rfloor$, 1, $\lceil k/2 \rceil$, $\lfloor k/2 \rfloor$, 1, $\cdots$.

3.2. Algorithms and Applications

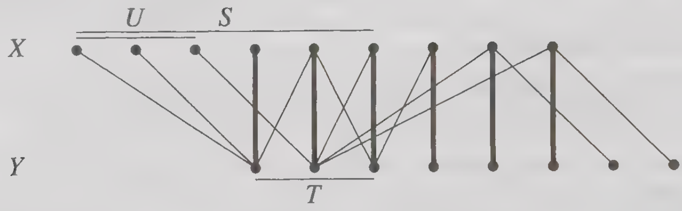
MAXIMUM BIPARTITE MATCHING

To find a maximum matching, we iteratively seek augmenting paths to enlarge the current matching. In a bipartite graph, if we don't find an augmenting path, we will find a vertex cover with the same size as the current matching, thereby proving that the current matching has maximum size. This yields both an algorithm to solve the maximum matching problem and an algorithmic proof of the König–Egerváry Theorem.

Given a matching M in an X,Y-bigraph G, we search for M-augmenting paths from each M-unsaturated vertex in X. We need only search from vertices in X, because every augmenting path has odd length and thus has ends in both X and Y. We will search from the unsaturated vertices in X simultaneously. Starting with a matching of size 0, $\alpha'(G)$ applications of the Augmenting Path Algorithm produce a maximum matching.

3.2.1. Algorithm. (Augmenting Path Algorithm).
Input: An X,Y-bigraph G, a matching M in G, and the set U of M-unsaturated vertices in X.
Idea: Explore M-alternating paths from U, letting $S \subseteq X$ and $T \subseteq Y$ be the sets of vertices reached. *Mark* vertices of S that have been explored for path extensions. As a vertex is reached, record the vertex from which it is reached.
Initialization: $S = U$ and $T = \emptyset$.
Iteration: If S has no unmarked vertex, stop and report $T \cup (X - S)$ as a minimum cover and M as a maximum matching. Otherwise, select an unmarked $x \in S$. To explore x, consider each $y \in N(x)$ such that $xy \notin M$. If y is unsaturated, terminate and report an M-augmenting path from U to y. Otherwise, y is matched to some $w \in X$ by M. In this case, include y in T (reached from x)

and include w in S (reached from y). After exploring all such edges incident to x, mark x and iterate. ■



When exploring x in the iterative step, we may reach a vertex $y \in T$ that we have reached previously. Recording x as the previous vertex on the path may change which M-augmenting path we report, but it won't change whether such a path exists.

3.2.2. Theorem. Repeatedly applying the Augmenting Path Algorithm to a bipartite graph produces a matching and a vertex cover of equal size.

Proof: We need only verify that the Augmenting Path Algorithm produces an M-augmenting path or a vertex cover of size $|M|$. If the algorithm produces an M-augmenting path, we are finished. Otherwise, it terminates by marking all vertices of S and claiming that $R = T \cup (X - S)$ is a vertex cover of size $|M|$. We must prove that R is a vertex cover and has size $|M|$.

To show that R is a vertex cover, it suffices to show that there is no edge joining S to $Y - T$. An M-alternating path from U enters X only on an edge of M. Hence every vertex x of $S - U$ is matched via M to a vertex of T, and there is no edge of M from S to $Y - T$. Also there is no such edge outside M. When the path reaches $x \in S$, it can continue along any edge not in M, and exploring x puts all other neighbors of x into T. Since the algorithm marks all of S before terminating, all edges from S go to T.

Now we study the size of R. The algorithm puts only saturated vertices in T; each $y \in T$ is matched via M to a vertex of S. Since $U \subseteq S$, also each vertex of $X - S$ is saturated, and the edges of M incident to $X - S$ cannot involve T. Hence they are different from the edges saturating T, and we find that M has at least $|T| + |X - S|$ edges. Since there is no matching larger than this vertex cover, we have $|M| = |T| + |X - S| = |R|$. ■

In addition to studying the correctness of algorithms, we are concerned about the time (number of computational steps) they use. We measure this as a function of the size of the input. For graph problems, we usually use the order $n(G)$ and/or size $e(G)$ to measure the input size.

3.2.3. Definition. The **running time** of an algorithm is the maximum number of computational steps used, expressed as a function of the size of the input. A **good algorithm** is one that has polynomial running time.

Running time is often expressed as "$O(f)$", where f is a function of the size of the input. Here $O(f)$ denotes the set of functions g such that $|g(x)|$ is bounded by a constant multiple of $|f(x)|$ when x is sufficiently large (that is, there exist c, a such that $|g(x)| \le c\,|f(x)|$ when $|x| \ge a$).

Many problems we study in Chapters 1-4 have good algorithms; other notions of complexity (Appendix B) need not trouble us yet. Since we don't know how long a particular operation may take on a particular computer, constant factors in running time have little meaning. Hence the "Big Oh" notation $O(f)$ is convenient. When f is a quadratic polynomial, we typically abuse notation by writing $O(n^2)$ instead of $O(f)$ to describe functions that grow at most quadratically in terms of n.

3.2.4. Remark. Let G be an X,Y-bigraph with n vertices and m edges. Since $\alpha'(G) \le n/2$, we find a maximum matching in G by applying Algorithm 3.2.1 at most $n/2$ times. Each application explores a vertex of X at most once, just before marking it; thus it considers each edge at most once. If the time for one edge exploration is bounded by a constant, then this algorithm to find a maximum matching runs in time $O(nm)$. Theorem 3.2.22 presents a faster algorithm, with running time $O(\sqrt{n}m)$. Section 3.3 discusses a good algorithm for maximum matching in general graphs. ■

WEIGHTED BIPARTITE MATCHING

Our results on maximum matching generalize to weighted X,Y-bigraphs, where we seek a matching of maximum total weight. If our graph is not all of $K_{n,n}$, then we insert the missing edges and assign them weight 0. This does not affect the numbers we can obtain as the weight of a matching. Thus we assume that our graph is $K_{n,n}$.

Since we consider only nonnegative edge weights, some maximum weighted matching is a perfect matching; thus we seek a perfect matching. We solve both the maximum weighted matching problem and its dual.

3.2.5. Example. *Weighted bipartite matching and its dual.* A farming company owns n farms and n processing plants. Each farm can produce corn to the capacity of one plant. The profit that results from sending the output of farm i to plant j is $w_{i,j}$. Placing weight $w_{i,j}$ on edge x_iy_j gives us a weighted bipartite graph with partite sets $X = \{x_1, \dots, x_n\}$ and $Y = \{y_1, \dots, y_n\}$. The company wants to select edges forming a matching to maximize total profit.

The government claims that too much corn is being produced, so it will pay the company not to process corn. The government will pay u_i if the company agrees not to use farm i and v_j if it agrees not to use plant j. If $u_i + v_j < w_{i,j}$, then the company makes more by using the edge x_iy_j than by taking the government payments for those vertices. In order to stop all production, the government must offer amounts such that $u_i + v_j \ge w_{i,j}$ for all i, j. The government wants to find such values to minimize $\sum u_i + \sum v_j$. ■

3.2.6. Definition. A **transversal** of an n-by-n matrix consists of n positions, one in each row and each column. Finding a transversal with maximum sum is the **Assignment Problem**. This is the matrix formulation of the **maximum weighted matching** problem, where nonnegative weight $w_{i,j}$ is assigned to edge $x_i y_j$ of $K_{n,n}$ and we seek a perfect matching M to maximize the total weight $w(M)$.

With these weights, a **(weighted) cover** is a choice of labels $u_i, \ldots, u_n$ and $v_j, \ldots, v_n$ such that $u_i + v_j \geq w_{i,j}$ for all i, j. The **cost** $c(u, v)$ of a cover (u, v) is $\sum u_i + \sum v_j$. The **minimum weighted cover** problem is that of finding a cover of minimum cost.

Note that the problem of minimum weight perfect matching can be solved using maximum weight matching; simply replace each weight $w_{i,j}$ with $M - w_{i,j}$ for some large number M.

The next lemma shows that the weighted matching and weighted cover problems are dual problems.

3.2.7. Lemma. For a perfect matching M and cover (u, v) in a weighted bipartite graph G, $c(u, v) \geq w(M)$. Also, $c(u, v) = w(M)$ if and only if M consists of edges $x_i y_j$ such that $u_i + v_j = w_{i,j}$. In this case, M and (u, v) are optimal.

Proof: Since M saturates each vertex, summing the constraints $u_i + v_j \geq w_{i,j}$ that arise from its edges yields $c(u, v) \geq w(M)$ for every cover (u, v). Furthermore, if $c(u, v) = w(M)$, then equality must hold in each of the n inequalities summed. Finally, since $c(u, v) \geq w(M)$ for every matching and every cover, $c(u, v) = w(M)$ implies that there is no matching with weight greater than $c(u, v)$ and no cover with cost less than $w(M)$. ■

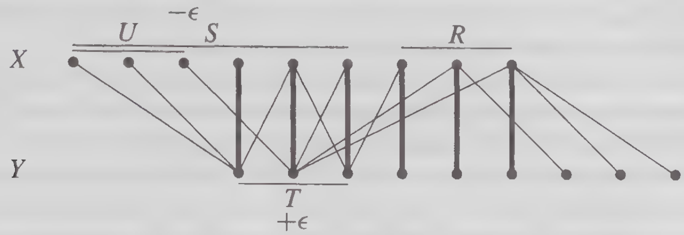
A matching and a cover have the same value only when the edges of the matching are covered with equality. This leads us to an algorithm.

3.2.8. Definition. The **equality subgraph** $G_{u,v}$ for a cover (u, v) is the spanning subgraph of $K_{n,n}$ having the edges $x_i y_j$ such that $u_i + v_j = w_{i,j}$.

If $G_{u,v}$ has a perfect matching, then its weight is $\sum u_i + \sum v_j$, and by Lemma 3.2.7 we have the optimal solution. Otherwise, we find a matching M and a vertex cover Q of the same size in $G_{u,v}$ (by using the Augmenting Path Algorithm, for example). Let $R = Q \cap X$ and $T = Q \cap Y$. Our matching of size $|Q|$ consists of $|R|$ edges from R to $Y - T$ and $|T|$ edges from T to $X - R$, as shown below. To seek a larger matching in the equality subgraph, we change (u, v) to introduce an edge from $X - R$ to $Y - T$ while maintaining equality on all edges of M.

A cover requires $u_i + v_j \geq w_{i,j}$ for all i, j; the difference $u_i + v_j - w_{i,j}$ is the **excess** for i, j. Edges joining $X - R$ and $Y - T$ are not in $G_{u,v}$ and have positive excess. Let ϵ be the minimum excess on the edges from $X - R$ to $Y - T$. Reducing u_i by ϵ for all $x_i \in X - R$ maintains the cover condition for these edges while bringing at least one into the equality subgraph. To maintain the cover condition for the edges from $X - R$ to T, we also increase v_j by ϵ for $y_j \in T$.

We repeat the procedure with the new equality subgraph; eventually we obtain a cover whose equality subgraph has a perfect matching. The resulting algorithm was named the **Hungarian Algorithm** by Kuhn in honor of the work of König and Egerváry on which it is based.



3.2.9. Algorithm. (Hungarian Algorithm—Kuhn [1955], Munkres [1957]).
Input: A matrix of weights on the edges of $K_{n,n}$ with bipartition X, Y.
Idea: Iteratively adjusting the cover (u, v) until the equality subgraph $G_{u,v}$ has a perfect matching.
Initialization: Let (u, v) be a cover, such as $u_i = \max_j w_{i,j}$ and $v_j = 0$.
Iteration: Find a maximum matching M in $G_{u,v}$. If M is a perfect matching, stop and report M as a maximum weight matching. Otherwise, let Q be a vertex cover of size $|M|$ in $G_{u,v}$. Let $R = X \cap Q$ and $T = Y \cap Q$. Let

$$\epsilon = \min\{u_i + v_j - w_{i,j}\colon x_i \in X - R,\ y_j \in Y - T\}.$$

Decrease u_i by ϵ for $x_i \in X - R$, and increase v_j by ϵ for $y_j \in T$. Form the new equality subgraph and repeat. ■

We have presented the algorithm using bipartite graphs, but repeatedly drawing a changing equality subgraph is awkward. Therefore, we compute with matrices. The initial weights form a matrix A with $w_{i,j}$ in position i, j. We associate the vertices and the labels (u, v) with the rows and columns, which serve as X and Y, respectively. We subtract $w_{i,j}$ from $u_i + v_j$ to obtain the **excess matrix**: $c_{i,j} = u_i + v_j - w_{i,j}$. The edges of the equality subgraph correspond to 0s in the excess matrix.

3.2.10. Example. *Solving the Assignment Problem.* The first matrix below is the matrix of weights. The others display a cover (u, v) and the corresponding excess matrix. We underscore entries in the excess matrix to mark a maximum matching M of $G_{u,v}$, which appears as bold edges in the equality subgraphs drawn for the first two excess matrices. (Drawing the equality subgraphs is not necessary.) A matching in $G_{u,v}$ corresponds to a set of 0s in the excess matrix with no two in any row or column; call this a **partial transversal**.

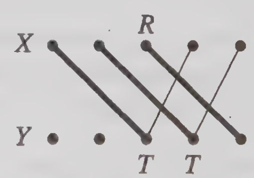
A set of rows and columns covering the 0s in the excess matrix is a **covering set**; this corresponds to a vertex cover in $G_{u,v}$. A covering set of size less than n yields progress toward a solution, since the next weighted cover costs less. We study the 0s in the excess matrix and find a partial transversal and a covering set of the same size. In a small matrix, we can do this by inspection.

We underscore the 0s of a partial transversal, and we use Rs and Ts to label the rows and columns of the covering set. At each iteration, we compute the minimum excess on the positions *not* in a covered row or column (in rows $X - R$ and columns $Y - T$). These uncovered positions have positive excess (the corresponding edges are not in the equality subgraph). The value ϵ defined in Algorithm 3.2.9 is the minimum of these excesses. We reduce the label u_i by ϵ on rows not in R and increase the label v_j by ϵ on columns in T.

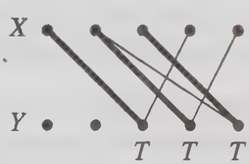
In the example below, the covering set used in the first iteration reduces the cost of the cover but does not augment the maximum matching in the equality subgraph. The second iteration produces a perfect matching. Using the last three columns as a covering set in the first iteration would augment the matching immediately.

The transversal of 0s after the final iteration identifies a perfect matching whose total weight equals the cost of the final cover. The corresponding edges have weights $5, 4, 6, 8, 8$ in the original data, which sum to 31. The labels $4, 5, 7, 4, 6$ and $0, 0, 2, 2, 1$ in the final cover satisfy each edge exactly and also sum to 31. The value of the optimal solution is unique, but the solution itself is not; this example has many maximum weight matchings and many minimum cost covers, but all have total weight 31. ■

$$
\begin{pmatrix} 4 & 1 & 6 & 2 & 3 \\ 5 & 0 & 3 & 7 & 6 \\ 2 & 3 & 4 & 5 & 8 \\ 3 & 4 & 6 & 3 & 4 \\ 4 & 6 & 5 & 8 & 6 \end{pmatrix}
\rightarrow
\begin{array}{c} \\ 6 \\ 7 \\ 8 \\ 6 \\ 8 \\ \\ \end{array}
\begin{array}{c} \begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \end{array} \\ \begin{pmatrix} 2 & 5 & \underline{0} & 4 & 3 \\ 2 & 7 & 4 & \underline{0} & 1 \\ 6 & 5 & 4 & 3 & \underline{0} \\ 3 & 2 & 0 & 3 & 2 \\ 4 & 2 & 3 & 0 & 2 \end{pmatrix} \\ \begin{array}{ccccc} & & T & T & \end{array} \end{array}
\begin{array}{c} \\ \\ \\ R \\ \\ \\ \\ \end{array}
$$



$$
\begin{array}{c} \\ 5 \\ 6 \\ 8 \\ 5 \\ 7 \\ \\ \end{array}
\begin{array}{c} \begin{array}{ccccc} 0 & 0 & 1 & 1 & 0 \end{array} \\ \begin{pmatrix} 1 & 4 & \underline{0} & 4 & 2 \\ 1 & 6 & 4 & \underline{0} & 0 \\ 6 & 5 & 5 & 4 & \underline{0} \\ 2 & 1 & 0 & 3 & 1 \\ 3 & 1 & 3 & 0 & 1 \end{pmatrix} \\ \begin{array}{ccccc} & & T & T & T \end{array} \end{array}
$$



$$
\rightarrow
\begin{array}{c} \\ 4 \\ 5 \\ 7 \\ 4 \\ 6 \end{array}
\begin{array}{c} \begin{array}{ccccc} 0 & 0 & 2 & 2 & 1 \end{array} \\ \begin{pmatrix} 0 & 3 & \underline{0} & 4 & 2 \\ \underline{0} & 5 & 4 & 0 & 0 \\ 5 & 4 & 5 & 4 & \underline{0} \\ 1 & \underline{0} & 0 & 3 & 1 \\ 2 & 0 & 3 & \underline{0} & 1 \end{pmatrix} \end{array}
$$

3.2.11. Theorem. The Hungarian Algorithm finds a maximum weight matching and a minimum cost cover.

Proof: The algorithm begins with a cover. It can terminate only when the equality subgraph has a perfect matching, which guarantees equal value for the current matching and cover. Suppose that (u, v) is the current cover and that the equality subgraph has no perfect matching. Let (u', v') denote the new lists of numbers assigned to the vertices. Because ϵ is the minimum of a nonempty finite set of positive numbers, $\epsilon > 0$.

We verify first that (u', v') is a cover. The change of labels on vertices of $X - R$ and T yields $u'_i + v'_j = u_i + v_j$ for edges $x_i y_j$ from $X - R$ to T or from R to $Y - T$. If $x_i \in R$ and $y_j \in T$, then $u'_i + v'_j = u_i + v_j + \epsilon$, and the weight remains covered. If $x_i \in X - R$ and $y_j \in Y - T$, then $u'_i + v'_j$ equals $u_i + v_j - \epsilon$, which by the choice of ϵ is at least $w_{i,j}$.

The algorithm terminates only when the equality subgraph has a perfect matching, so it suffices to show that it does terminate. Suppose that the weights $w_{i,j}$ are rational. Multiplying the weights by their least common denominator yields an equivalent problem with integer weights. We can now assume that the labels in the current cover also are integers. Thus each excess is also an integer, and at each iteration we reduce the cost of the cover by an integer amount. Since the cost starts at some value and is bounded below by the weight of a perfect matching, after finitely many iterations we have equality.

For real-valued weights in general, see Remark 3.2.12). ■

3.2.12.* Remark. When the weights are real numbers, the algorithm still works if we obtain vertex covers in the equality subgraph more carefully. We show that the algorithm terminates within n^2 iterations. Because the edges of M remain in the new equality subgraph, the size of the current matching never decreases. Since the size of the matching can increase at most n times, it suffices to show that it must increase within n iterations.

If we find the maximum matching M by iterating the Augmenting Path Algorithm, then the last iteration presents us with a vertex cover. We find it by exploring M-alternating paths from the set U of M-unsaturated vertices in X. With S and T denoting the sets of vertices reachable in X and T, we obtain the vertex cover $R \cup T$, where $R = X - S$.

Applying a step of the Hungarian Algorithm using the vertex cover $R \cup T$ maintains equality on M and all the edges in M-alternating paths from U. Edges from T to R disappear from the equality subgraph, but we don't care because they don't appear in M-alternating paths from U. Introducing an edge from S to $Y - T$ either creates an M-augmenting path or increases T while leaving U unchanged. Since we can increase T at most n times, we obtain a larger matching in the equality subgraph within n iterations. ■

3.2.13.* Remark. The maximum matching and vertex cover problems in bipartite graphs are special cases of the weighted problems. Given a bipartite graph G, form a weighted graph with weight 1 on the edges of G and weight 0 on the edges of $K_{n,n}$. The maximum weight of a matching is $\alpha'(G)$.

Given integer weights, the Hungarian algorithm always maintains integer labels in the weighted cover. Hence in this weighted cover problem we may restrict the values (labels) used to be integers. Further thought shows that these integers will always be 0 or 1.

The vertices receiving label 1 must cover the weight on the edges of G, so they form a vertex cover for G. Minimizing the sum of labels under the integer restriction is equivalent to finding the minimum number of vertices in a vertex cover for G. Hence the answer to the weighted cover problem is $\beta(G)$. ■

3.2.14.* Application. *Street Sweeping and the Transportation Problem.* A cleaning machine sweeping a curb must move in the same direction as traffic. This yields a digraph; a two-way street generates two oppositely directed edges, while a one-way street generates two edges in the same direction. We consider a simple version of the **Street Sweeping Problem**, discussed in more detail in Roberts [1978] as based on Tucker–Bodin [1976].

In New York City, parking is prohibited from some curbs each day to allow for street sweeping. For each day, this defines a **sweep subgraph** G of the full digraph H of curbs, consisting of those available for sweeping. Each $e \in E(H)$ has a **deadheading time** $t(e)$ needed to travel it without sweeping.

The question is how to sweep G while minimizing the total deadheading time spent without sweeping. This is a generalization of a directed version of the Chinese Postman Problem. If indegree equals outdegree at each vertex of G, then no deadheading is needed. Otherwise, we duplicate edges of G or add edges from H to obtain an Eulerian digraph G' containing G.

Let X be the set of vertices with excess indegree; let $\sigma(x) = d_G^-(x) - d_G^+(x)$ for $x \in X$. Let Y be the set with excess outdegree; let $\partial(y) = d_G^-(y) - d_G^+(y)$ for $y \in Y$. Note that $\sum_{x\in X} \sigma(x) = \sum_{y\in Y} \partial(y)$. To obtain G' from G, we must add $\sigma(x)$ edges with tails at $x \in X$ and $\partial(y)$ edges with heads at $y \in Y$. Since G' needs net outdegree 0 at each vertex, the additions form paths from X to Y. The cost $c(xy)$ of an x,y-path is the distance from x to y in the weighted digraph H, which can be found by Dijkstra's Algorithm.

This yields the **Transportation Problem**. Given supply $\sigma(x)$ for $x \in X$, demand $\partial(y)$ for $y \in Y$, cost $c(xy)$ per unit sent from x to y, and $\sum \sigma(x) = \sum \partial(y)$, we want to satisfy the demands at least total cost. A version of the problem was introduced by Kantorovich [1939]; the form above arose (with a constructive solution) in Hitchcock [1941] (see also Koopmans [1947]). The problem is discussed at length in Ford–Fulkerson [1962, p93–130].

When the supplies and demands are rational, the Assignment Problem can be applied. First scale up to obtain integer supplies and demands. Next define a matrix with $\sum \sigma(x)$ rows and columns. For each $x \in X$, create $\sigma(x)$ rows. For each $y \in Y$, create $\delta(y)$ columns. When row i and column j represent x and y, let $w_{i,j} = M - c(xy)$, where $M = \max_{x,y} c(xy)$. A maximum weight matching now yields a minimum cost solution to the Transportation Problem. A generalization of the Transportation Problem appears in Section 4.3. ■

STABLE MATCHINGS (optional)

Instead of optimizing total weight for a matching, we may try to optimize using preferences. Given n men and n women; we want to establish n "stable" marriages. If man x and woman a are paired with other partners, but x prefers a to his current partner and a prefers x to her current partner, then they might leave their current partners and switch to each other. In this situation we say that the unmatched pair (x, a) is an **unstable pair**.

3.2.15. Definition. A perfect matching is a **stable matching** if it yields no unstable unmatched pair.

3.2.16. Example. Given men x, y, z, w, women a, b, c, d, and preferences listed below, the matching $\{xa, yb, zd, wc\}$ is a stable matching. ■

Men $\{x, y, z, w\}$	Women $\{a, b, c, d\}$
$x : a > b > c > d$	$a : z > x > y > w$
$y : a > c > b > d$	$b : y > w > x > z$
$z : c > d > a > b$	$c : w > x > y > z$
$w : c > b > a > d$	$d : x > y > z > w$

In their paper "College admissions and the stability of marriage", Gale and Shapley proved that a stable matching always exists and can be found using a relatively simple algorithm. In the algorithm, men and women do not play symmetric roles; we will discuss this importance of this difference later. The algorithm below generates the matching of Example 3.2.16.

3.2.17. Algorithm. (Gale–Shapley Proposal Algorithm)
Input: Preference rankings by each of n men and n women.
Idea: Produce a stable matching using proposals by maintaining information about who has proposed to whom and who has rejected whom.
Iteration: Each man proposes to the highest woman on his preference list who has not previously rejected him. If each woman receives exactly one proposal, stop and use the resulting matching. Otherwise, every woman receiving more than one proposal rejects all of them except the one that is highest on her preference list. Every woman receiving a proposal says "maybe" to the most attractive proposal received. ■

3.2.18. Theorem. (Gale–Shapley [1962]) The Proposal Algorithm produces a stable matching.

Proof: The algorithm terminates (with some matching), because on each nonterminal iteration, the total length of the lists of potential mates for the men decreases. This can happen only n^2 times.

Key Observation: the sequence of proposals made by each man is nonincreasing in his preference list, and the sequence of men to whom a woman says "maybe" is nondecreasing in her preference list, culminating in the man assigned. This holds because men propose repeatedly to the same woman until rejected, and women say "maybe" to the same man until a better offer arrives.

If the result is not stable, then there is an unstable unmatched pair (x, a), with x matched to b and y matched to a. By the key observation, x never proposed to a during the algorithm, since a received a mate less desirable than x. The key observation also implies that x would not have proposed to b without earlier proposing to a. This contradiction confirms the stability of the result. ■

The asymmetry of the proposal algorithm suggests asking which sex is happier. When the first choices of the men are distinct, they all get their first choice, and the women are stuck with whomever proposed. When the algorithm runs with women proposing, every woman is at least as happy as when men do the proposing, and every man is at least as unhappy. In Example 3.2.16, running the algorithm with women proposing immediately yields the matching $\{xd, yb, ca, wc\}$, in which all women are matched to their first choices. In fact, among all stable matchings, every man is happiest in the one produced by the male-proposal algorithm, and every woman is happiest under the female-proposal algorithm (Exercise 11). Societal conventions thus favor men.

The algorithm is used in another setting. Each year, the graduates of medical schools submit preference lists of hospitals where they wish to be residents. The hospitals have their own preferences; we model a hospital with multiple openings as several hospitals with the same preference list. Chaos in the market for residents (then called interns) forced hospitals to devise and implement the algorithm ten years before the Gale–Shapley paper defined and solved the problem! The result was the National Resident Matching Program, a non-profit corporation established in 1952 to provide a uniform appointment date and matching procedure.

Who is happier with the outcome? Since the medical organizations ran the algorithm, it is not surprising that initially they did the proposing and were happier with the outcome. The distinction is even clearer in another setting; students applying for jobs have preferences, but the employers make the proposals, called "job offers". Unhappiness with the NRMP caused the system to be changed in 1998 to a student-proposing algorithm. In 1998 the system processed 35,823 applicants for 22,451 positions. Additional details about the system can be found at nrmp.aamc.org/nrmp/mainguid/ on the World Wide Web.

There may be stable matchings other than those found by the two versions of the proposal algorithm. To seek a "fair" stable matching, we could give each person a number of points with which to rate preferences. The weight for the pair xa is then the sum of the points that x gives to a and a gives to x. The Hungarian Algorithm would yield a matching of maximum total weight, but this might not be a stable matching (Exercise 10). Other approaches appear in the books Knuth [1976] and Gusfield–Irving [1989], which discuss stable marriages and related topics.

FASTER BIPARTITE MATCHING (optional)

We began this section with an algorithm for finding maximum matchings in bipartite graphs. The running time can be improved by seeking augmenting paths in a clever order; when short augmenting paths are available, we needn't explore many edges to find one. Using a Breadth-First Search simultaneously from all the unsaturated vertices of X, we can find many paths of the same length with one examination of the edge set. Hopcroft and Karp [1973] proved that subsequent augmentations must use longer paths, so the searches can be grouped in phases finding paths of the same lengths. They combined these ideas to show that few phases are needed, enabling maximum matchings in n-vertex bipartite graphs to be found in $O(n^{2.5})$ time.

3.2.19. Remark. If M is a matching of size r and M^* is a matching of size $s > r$, then there exist at least $s - r$ vertex-disjoint M-augmenting paths. At least this many such paths can be found in $M \triangle M^*$. ■

The next lemma implies that the sequence of path lengths in successive shortest augmentations is nondecreasing. Here we treat paths as sets of edges, and cardinality indicates number of edges.

3.2.20. Lemma. If P is a shortest M-augmenting path and P' is $M \triangle P$-augmenting, then $|P'| \ge |P| + 2\,|P \cap P'|$ (treating P as an edge set).

Proof: Note that $M \triangle P$ is the matching obtained by using P to augment M. Let N be the matching $(M \triangle P) \triangle P'$ obtained by using P' to augment $M \triangle P$. Since $|N| = |M| + 2$, Remark 3.2.19 guarantees that $M \triangle N$ contains two disjoint M-augmenting paths P_1 and P_2. Each of these is at least as long as P, since P is a shortest M-augmentating path.

Since N is obtained from M by switching the edges in P and then switching the edges in P', an edge belongs to exactly one of M and N if and only if it belongs to exactly one of P and P'. Therefore, $M \triangle N = P \triangle P'$. This yields $|P \triangle P'| \ge |P_1| + |P_2| \ge 2\,|P|$. Thus

$$2\,|P| \le |P \triangle P'| = |P| + |P'| - 2\,|P \cap P'|.$$

We conclude that $|P'| \ge |P| + 2\,|P \cap P'|$. ■

3.2.21. Lemma. If $P_1, P_2, \cdots$ is a list of successive shortest augmentations, then the augmentations of the same length are vertex-disjoint paths.

Proof: We use the method of contradiction. Let P_k, P_l with $l > k$ be a closest pair in the list that have the same size but are not vertex-disjoint. By Lemma 3.2.20, the lengths of successive shortest augmenting paths are nondecreasing, so $P_k, \ldots, P_l$ all have the same length. Since P_k, P_l is a closest intersecting pair with the same length, the paths $P_{k+1}, \ldots, P_l$ are pairwise disjoint.

Let M' be the matching given by the augmentations $P_1, \ldots, P_k$. Since $P_{k+1}, \ldots, P_l$ are pairwise disjoint, P_l is an M'-augmenting path. By Lemma 3.2.20, $|P_l| \ge |P_k| + |P_l \cap P_k|$. Since $|P_l| = |P_k|$, there is no common edge.

On the other hand, there must be a common edge. Each vertex of P_k is saturated in M' using an edge of P_k, and every vertex of an M'-augmenting path P_l that is saturated in M' (such as a vertex common to P_l and P_k) must contribute its saturated edge to P_l.

The contradiction implies that there is no such pair P_k, P_l. ■

3.2.22. Theorem. (Hopcroft–Karp [1973]) The breadth-first phased maximum matching algorithm runs in $O(\sqrt{n}m)$ time on bipartite graphs with n vertices and m edges.

Proof: By Lemmas 3.2.20–3.2.21, searching simultaneously from all unsaturated vertices of X for shortest augmentations yields vertex-disjoint paths, after which all other augmenting paths are longer. Hence the augmentations of each length are found in one examination of the edge set, running in time $O(m)$. It suffices to prove that there are at most $2\left\lfloor\sqrt{n/2}\right\rfloor+2$ phases.

List the augmenting paths as $P_1,\ldots,P_s$ in order by length, with $s=\alpha'(G)\le n/2$. Since paths of the same length are vertex-disjoint, each P_{i+1} is an augmenting path for the matching M_i formed by using $P_1,\ldots,P_i$. It suffices to prove the more general statement that whenever $P_1,\ldots,P_s$ are successive shortest augmenting paths that build a maximum matching, the number of distinct lengths among these paths is at most $2\lfloor\sqrt{s}\rfloor+2$.

Let $r=\lfloor s-\sqrt{s}\rfloor$. Because $|M_r|=r$ and the maximum matching has size s, Remark 3.2.19 yields at least $s-r$ vertex-disjoint M_r-augmenting paths. The shortest of these paths uses at most $\lfloor r/(s-r)\rfloor$ edges from M_r. Hence $|P_{r+1}|\le 2\lfloor r/(s-r)\rfloor+1$. Since $\lfloor r/(s-r)\rfloor<\lfloor s/\lceil\sqrt{s}\rceil\rfloor\le\lfloor\sqrt{s}\rfloor$, the paths up to P_r provide all but the last $\lceil\sqrt{s}\rceil$ augmentations using length at most $2\lfloor\sqrt{s}\rfloor+1$. There are at most $\lfloor\sqrt{s}\rfloor+1$ distinct odd integers up to this value, and even if the last $\lceil\sqrt{s}\rceil$ paths have distinct lengths, they provide at most $\lfloor\sqrt{s}\rfloor+1$ additional lengths, so altogether we use at most $2\lfloor\sqrt{s}\rfloor+2$ distinct lengths. ■

Even and Tarjan [1975] extended this to solve in time $O(\sqrt{n}m)$ a more general problem that includes maximum bipartite matching.

EXERCISES

3.2.1. (−) Using nonnegative edge weights, construct a 4-vertex weighted graph in which the matching of maximum weight is not a matching of maximum size.

3.2.2. (−) Show how to use the Hungarian Algorithm to test for the existence of a perfect matching in a bipartite graph.

3.2.3. (∗−) Give an example of the stable matching problem with two men and two women in which there is more than one stable matching.

3.2.4. (∗−) Determine the stable matchings resulting from the Proposal Algorithm run with men proposing and with women proposing, given the preference lists below.

Men $\{u,v,w,x,y,z\}$	Women $\{a,b,c,d,e,f\}$
$u: a>b>d>c>f>e$	$a: z>x>y>u>v>w$
$v: a>b>c>f>e>d$	$b: y>z>w>x>v>u$
$w: c>b>d>a>f>e$	$c: v>x>w>y>u>z$
$x: c>a>d>b>e>f$	$d: w>y>u>x>z>v$
$y: c>d>a>b>f>e$	$e: u>v>x>w>y>z$
$z: d>e>f>c>b>a$	$f: u>w>x>v>z>y$

• • • • •

3.2.5. Find a transversal of maximum total sum (weight) in each matrix below. Prove that there is no larger weight transversal by exhibiting a solution to the dual problem. Explain why this proves that there is no larger transversal.

(a)	(b)	(c)
4 4 4 3 6	7 8 9 8 7	1 2 3 4 5
1 1 4 3 4	8 7 6 7 6	6 7 8 7 2
1 4 5 3 5	9 6 5 4 6	1 3 4 4 5
5 6 4 7 9	8 5 7 6 4	3 6 2 8 7
5 3 6 8 3	7 6 5 5 5	4 1 3 5 4

3.2.6. Find a minimum-weight transversal in the matrix below, and use duality to prove that the solution is optimal. (Hint: Use a transformation of the problem.)

$$\begin{pmatrix} 4 & 5 & 8 & 10 & 11 \\ 7 & 6 & 5 & 7 & 4 \\ 8 & 5 & 12 & 9 & 6 \\ 6 & 6 & 13 & 10 & 7 \\ 4 & 5 & 7 & 9 & 8 \end{pmatrix}$$

3.2.7. *The Bus Driver Problem.* Let there be n bus drivers, n morning routes with durations $x_1, \ldots, x_n$, and n afternoon routes with durations $y_1, \ldots, y_n$. A driver is paid overtime when the morning route and afternoon route exceed total time t. The objective is to assign one morning run and one afternoon run to each driver to minimize the total amount of overtime. Express this as a weighted matching problem. Prove that giving the ith longest morning route and ith shortest afternoon route to the same driver, for each i, yields an optimal solution. (Hint: Do not use the Hungarian Algorithm; consider the special structure of the matrix.) (R.B. Potts)

3.2.8. Let the entries in matrix A have the form $w_{i,j} = a_i b_j$, where $a_1, \ldots, a_n$ are numbers associated with the rows and $b_1, \ldots, b_n$ are numbers associated with the columns. Determine the maximum weight of a transversal of A. What happens when $w_{i,j} = a_i + b_j$? (Hint: In each case, guess the general pattern by examining the solution when $n = 2$.)

3.2.9. $(*)$ A mathematics department offers k seminars in different topics to its n students. Each student will take one seminar; the ith seminar will have k_i students, where $\sum k_i = n$. Each student submits a preference list ranking the k seminars. An assignment of the students to seminars is *stable* if no two students can both obtain more preferable seminars by switching their assignments. Show how to find a stable assignment using weighted bipartite matching. (Isaak)

3.2.10. $(*)$ Consider n men and n women, each assigning $n - i$ points to the ith person in his or her preference list. Let the weight of a pair be the sum of the points assigned by those two people. Construct an example where no maximum weight matching is a stable matching.

3.2.11. $(*!)$ Prove that if man x is paired with woman a in some stable matching, then a does not reject x in the Gale–Shapley Proposal Algorithm with men proposing. Conclude that among all stable matchings, *every* man is happiest in the matching produced by this algorithm. (Hint: Consider the first occurrence of such a rejection.)

3.2.12. $(*)$ In the Stable Roommates Problem, each of $2n$ people has a preference ordering on the other $2n - 1$ people. A stable matching is a perfect matching such that no unmatched pair prefers each other to their current roommates. Prove that there is no stable matching when the preferences are those below. (Gale–Shapley [1962])

$$\begin{aligned} a &: b > c > d \\ b &: c > a > d \\ c &: a > b > d \\ d &: a > b > c \end{aligned}$$

3.2.13. $(*)$ In the stable roommates problem, suppose that each individual declares a top portion of the preference list as "acceptable". Define the *acceptability graph* to be the graph whose vertices are the people and whose edges are the pairs of people who rank each other as acceptable. Prove that all sets of rankings with acceptability graph G lead to a stable matching if and only if G is bipartite. (Abeledo–Isaak [1991]).

3.3. Matchings in General Graphs

When discussing perfect matchings in graphs, it is natural to consider more general spanning subgraphs.

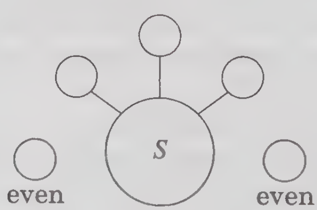
3.3.1. Definition. A **factor** of a graph G is a spanning subgraph of G. A k-**factor** is a spanning k-regular subgraph. An **odd component** of a graph is a component of odd order; the number of odd components of H is $o(H)$.

3.3.2. Remark. A 1-factor and a perfect matching are almost the same thing. The precise distinction is that "1-factor" is a spanning 1-regular subgraph of G, while "perfect matching" is the set of edges in such a subgraph.

A 3-regular graph that has a perfect matching decomposes into a 1-factor and a 2-factor. ■

TUTTE'S 1-FACTOR THEOREM

Tutte found a necessary and sufficient condition for which graphs have 1-factors. If G has a 1-factor and we consider a set $S \subseteq V(G)$, then every odd component of $G - S$ has a vertex matched to something outside it, which can only belong to S. Since these vertices of S must be distinct, $o(G - S) \le |S|$.



The condition "For all $S \subseteq V(G)$, $o(G - S) \le |S|$" is **Tutte's Condition**. Tutte proved that this obvious necessary condition is also sufficient (TONCAS).

Many proofs are known, such as Exercise 13 and Exercise 27. We present the proof by Lovász using the ideas of symmetric difference and extremality.

3.3.3. Theorem. (Tutte [1947]) A graph G has a 1-factor if and only if $o(G - S) \le |S|$ for every $S \subseteq V(G)$.

Proof: (Lovász [1975]). *Necessity.* The odd components of $G - S$ must have vertices matched to distinct vertices of S.

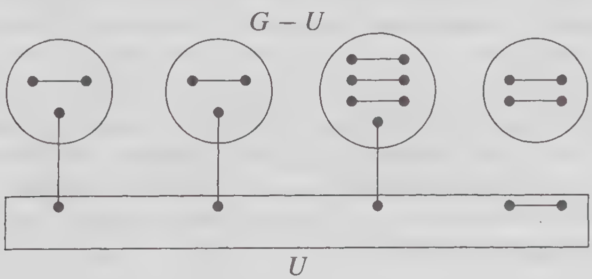
Sufficiency. When we add an edge joining two components of $G - S$, the number of odd components does not increase (odd and even together become one odd component, two components of the same parity become one even component). Hence Tutte's Condition is preserved by addition of edges: if $G' = G + e$ and $S \subseteq V(G)$, then $o(G' - S) \le o(G - S) \le |S|$. Also, if $G' = G + e$ has no 1-factor, then G has no 1-factor.

Therefore, the theorem holds unless there exists a simple graph G such that G satisfies Tutte's Condition, G has no 1-factor, and adding any missing edge to G yields a graph with a 1-factor. Let G be such a graph. We obtain a contradiction by showing that G actually does contain a 1-factor.

Let U be the set of vertices in G that have degree $n(G) - 1$.

Case 1: $G - U$ consists of disjoint complete graphs. In this case, the vertices in each component of $G - U$ can be paired in any way, with one extra in the odd components. Since $o(G - U) \le |U|$ and each vertex of U is adjacent to all of $G - U$, we can match the leftover vertices to vertices of U.

The remaining vertices are in U, which is a clique. To complete the 1-factor, we need only show that an even number of vertices remain in U. We have matched an even number, so it suffices to show that $n(G)$ is even. This follows by invoking Tutte's Condition for $S = \emptyset$, since a graph of odd order would have a component of odd order.

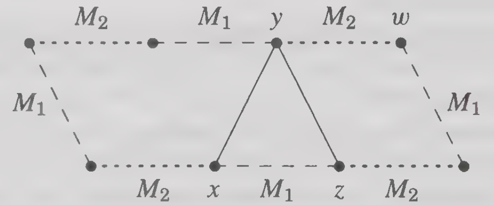


Case 2: $G - U$ is not a disjoint union of cliques. In this case, $G - U$ has two vertices at distance 2; these are nonadjacent vertices x, z with a common neighbor $y \notin U$ (Exercise 1.2.23a). Furthermore, $G - U$ has another vertex w not adjacent to y, since $y \notin U$. By the choice of G, adding an edge to G creates a 1-factor; let M_1 and M_2 be 1-factors in $G + xz$ and $G + yw$, respectively. It suffices to show that $M_1 \triangle M_2$ contains a 1-factor avoiding xz and yw, because this will be a 1-factor in G.

Let $F = M_1 \triangle M_2$. Since $xz \in M_1 - M_2$ and $yw \in M_2 - M_1$, both xz and yw are in F. Since every vertex of G has degree 1 in each of M_1 and M_2, every vertex of G has degree 0 or 2 in F. Hence the components of F are even cycles and isolated vertices (see Lemma 3.1.9). Let C be the cycle of F containing xz.

If C does not also contain yw, then the desired 1-factor consists of the edges of M_2 from C and all of M_1 not in C.

If C contains both yw and xz, as shown below, then to avoid them we use yx or yz. In the portion of C starting from y along yw, we use edges of M_1 to avoid using yw. When we reach $\{x, z\}$, we use zy if we arrive at z (as shown); otherwise, we use xy. In the remainder of C we use the edges of M_2. We have produced a 1-factor of C that does not use xz or yw. Combined with M_1 or M_2 outside C, we have a 1-factor of G. ■



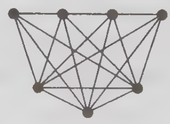
3.3.4. Remark. Like other characterization theorems (such as Theorem 1.2.18 and Theorem 3.1.11)), Theorem 3.3.3 yields short verifications both when the property holds *and* when it doesn't. We prove that G has a 1-factor exists by exhibiting one. When it doesn't exist, Theorem 3.3.3 guarantees that we can exhibit a set whose deletion leaves too many odd components. ■

3.3.5. Remark. For a graph G and any $S \subseteq V(G)$, counting the vertices modulo 2 shows that $|S| + o(G - S)$ has the same parity as $n(G)$. Thus also the difference $o(G - S) - |S|$ has the same parity as $n(G)$. We conclude that if $n(G)$ is even and G has no 1-factor, then $o(G - S)$ exceeds $|S|$ by at least 2 for some S. ■

For non-bipartite graphs (such as odd cycles), there may be a gap between $\alpha'(G)$ and $\beta(G)$ (see also Exercise 10). Nevertheless, another minimization problem yields a min-max relation for $\alpha'(T)$ in general graphs. This min-max relation generalizes Remark 3.3.5. The proof uses a graph transformation that involves a general graph operation.

3.3.6. Definition. The **join** of simple graphs G and H, written $G \vee H$, is the graph obtained from the disjoint union $G + H$ by adding the edges $\{xy : x \in V(G), y \in V(H)\}$.

$P_4 \vee K_3$

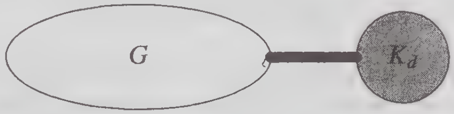


3.3.7. Corollary. (Berge–Tutte Formula—Berge [1958]) The largest number of vertices saturated by a matching in G is $\min_{S\subseteq V(G)}\{n(G) - d(S)\}$, where $d(S) = o(G - S) - |S|$.

Proof: Given $S \subseteq V(G)$, at most $|S|$ edges can match vertices of S to vertices in odd components of $G - S$, so every matching has at least $o(G - S) - |S|$ unsaturated vertices. We want to achieve this bound.

Let $d = \max\{o(G - S) - |S| : S \subseteq V(G)\}$. The case $S = \emptyset$ yields $d \geq 0$. Let $G' = G \vee K_d$. Since $d(S)$ has the same parity as $n(G)$ for each S, we know that $n(G')$ is even. If G' satisfies Tutte's Condition, then we obtain a matching of the desired size in G from a perfect matching in G', because deleting the d added vertices eliminates edges that saturate at most d vertices of G.

The condition $o(G' - S') \leq |S'|$ holds for $S' = \emptyset$ because $n(G')$ is even. If S' is nonempty but does not contain all of K_d, then $G' - S'$ has only one component, and $1 \leq |S'|$. Finally, when $K_d \subseteq S'$, we let $S = S' - V(K_d)$. We have $G' - S' = G - S$, so $o(G' - S') = o(G - S) \leq |S| + d = |S'|$. We have verified that G' satisfies Tutte's Condition. ■



Corollary 3.3.7 guarantees that there is a short PROOF that a maximum matching indeed has maximum size by exhibiting a vertex set S whose deletion leaves the appropriate number of odd components.

Most applications of Tutte's Theorem involve showing that some other condition implies Tutte's Condition and hence guarantees a 1-factor. Some were proved by other means long before Tutte's Theorem was available.

3.3.8. Corollary. (Petersen [1891]) Every 3-regular graph with no cut-edge has a 1-factor.

Proof: Let G be a 3-regular graph with no cut-edge. We prove that G satisfies Tutte's Condition. Given $S \subseteq V(G)$, we count the edges between S and the odd components of $G - S$. Since G is 3-regular, each vertex of S is incident to at most three such edges. If each odd component H of $G - S$ is incident to at least three such edges, then $3o(G - S) \leq 3|S|$ and hence $o(G - S) \leq |S|$, as desired.

Let m be the number of edges from S to H. The sum of the vertex degrees in H is $3n(H) - m$. Since H is a graph, the sum of its vertex degrees must be even. Since $n(H)$ is odd, we conclude that m must also be odd. Since G has no cut-edge, m cannot equal 1. We conclude that there are at least three edges from S to H, as desired. ■

Proof by contradiction would also be natural here. Assuming $o(G - S) > |S|$ also leads to $o(G - S) \leq |S|$, so we rewrite the proof directly. Corollary 3.3.8 is best possible; the Petersen graph satisfies the hypothesis but does not have *two* edge-disjoint 1-factors (Petersen [1898]).

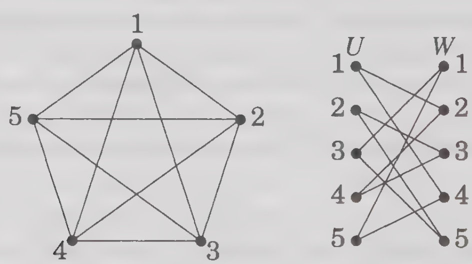
Petersen also proved a sufficient condition for 2-factors. A connected graph with even vertex degrees is Eulerian (Theorem 1.2.26) and decomposes into edge-disjoint cycles (Proposition 1.2.27). For regular graphs of even degree, the cycles in some decomposition can be grouped to form 2-factors.

3.3.9. Theorem. (Petersen [1891]) Every regular graph of even degree has a 2-factor.

Proof: Let G be a $2k$-regular graph with vertices $v_1, \dots, v_n$. Every component of G is Eulerian, with some Eulerian circuit C. For each component, define a bipartite graph H with vertices $u_1, \dots, u_n$ and $w_1, \dots, w_n$ by putting $u_i \leftrightarrow w_j$ if v_j immediately follows v_i somewhere on C. Because C enters and exits each vertex k times, H is k-regular. (Actually, H is the split of the digraph obtained by orienting G in accordance with C—see Definition 1.4.20.)

Being a regular bipartite graph, H has a 1-factor M (Corollary 3.1.13). The edge incident to w_i in H corresponds to an edge entering v_i in C. The edge incident to u_i in H corresponds to an edge exiting v_i. Thus the 1-factor in H transforms into a 2-regular spanning subgraph of this component of G. Doing this for each component of G yields a 2-factor of G. ■

3.3.10. Example. *Construction of a 2-factor.* Consider the Eulerian circuit in $G = K_5$ that successively visits 1231425435. The corresponding bipartite graph H is on the right. For the 1-factor whose u, w-pairs are 12, 43, 25, 31, 54, the resulting 2-factor is the cycle $(1, 2, 5, 4, 3)$. The remaining edges form another 1-factor, which corresponds to the 2-factor $(1, 4, 2, 3, 5)$ that remains in G. ■



f-FACTORS OF GRAPHS (optional)

A factor is a spanning subgraph of G; we ask about existence of factors of special types. A k-factor is a k-regular factor; we have studied 1-factors and 2-factors. We can try to specify the degree at each vertex.

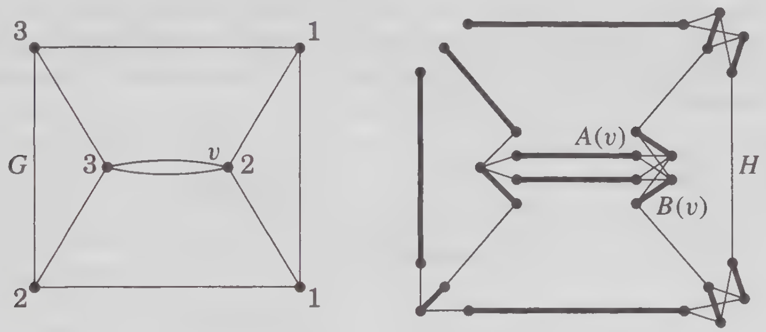
3.3.11. Definition. Given a function $f\colon V(G) \to \mathbb{N} \cup \{0\}$, an f-**factor** of a graph G is a subgraph H such that $d_H(v) = f(v)$ for all $v \in V(G)$.

Tutte [1952] proved a necessary and sufficient condition for a graph G to have an f-factor (see Exercise 29). He later reduced the problem to checking for a 1-factor in a related simple graph. We describe this reduction; it is a beautiful example of transforming a graph problem into a previously solved problem.

3.3.12. Example. *A graph transformation* (Tutte [1954a]). We assume that $f(w) \le d(w)$ for all w; otherwise G has too few edges at w to have an f-factor. We then construct a graph H that has a 1-factor if and only if G has an f-factor. Let $e(w) = d(w) - f(w)$; this is the *excess degree* at w and is nonnegative.

To construct H, replace each vertex v with a biclique $K_{d(v),e(v)}$ having partite sets $A(v)$ of size $d(v)$ and $B(v)$ of size $e(v)$. For each $vw \in E(G)$, add an edge joining one vertex of $A(v)$ to one vertex of $A(w)$. Each vertex of $A(v)$ participates in one such edge.

The figure below shows a graph G, vertex labels given by f, and the resulting simple graph H. The bold edges in H form a 1-factor that corresponds to an f-factor of G. In this example, the f-factor is not unique. ■



3.3.13. Theorem. A graph G has an f-factor if and only if the graph H constructed from G and f as in Example 3.3.12 has a 1-factor.

Proof: *Necessity.* If G has an f-factor, then the corresponding edges in H leave $e(v)$ vertices of $A(v)$ unmatched; match them arbitrarily to the vertices of $B(v)$ to obtain a 1-factor of H.

Sufficiency. From a 1-factor of H, deleting $B(v)$ and the vertices of $A(v)$ matched into $B(v)$ leaves $f(v)$ edges at v. Doing this for each v and merging the remaining $f(v)$ vertices of each $A(v)$ yields a subgraph of G with degree $f(v)$ at v. It is an f-factor of G. ■

Tutte's Condition for a 1-factor in the derived graph H of Example 3.3.12 transforms into a necessary and sufficient condition for an f-factor in G. Among the applications is a proof of the Erdős–Gallai [1960] characterization of degree sequences of simple graphs (Exercise 29).

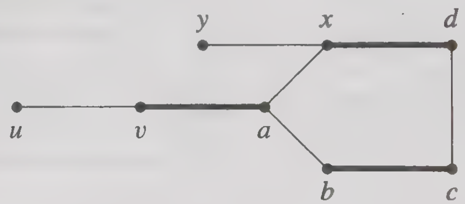
Given an algorithm to find a 1-factor, the correspondence in Theorem 3.3.13 provides an algorithmic test for an f-factor. Instead of just seeking a 1-factor (that is, a perfect matching), we next consider the more general problem of finding a maximum matching in a graph.

EDMONDS' BLOSSOM ALGORITHM (optional)

Berge's Theorem (Theorem 3.1.10) states that a matching M in G has maximum size if and only if G has no M-augmenting path. We can thus find a maximum matching using successive augmenting paths. Since we augment at most $n/2$ times, we obtain a good algorithm if the search for an augmenting path does not take too long. Edmonds [1965a] presented the first such algorithm in his famous paper "Paths, trees, and flowers".

In bipartite graphs, we can search quickly for augmenting paths (Algorithm 3.2.1) because we explore from each vertex at most once. An M-alternating path from u can reach a vertex x in the same partite set as u only along a saturated edge. Hence only once can we reach and explore x. This property fails in graphs with odd cycles, because M-alternating paths from an unsaturated vertex may reach x both along saturated and along unsaturated edges.

3.3.14. Example. In the graph below, with M indicated in bold, a search for shortest M-augmenting paths from u reaches x via the unsaturated edge ax. If we do not also consider a longer path reaching x via a saturated edge, then we miss the augmenting path u, v, a, b, c, d, x, y. ■



We describe Edmonds' solution to this difficulty. If an exploration of M-alternating paths from u reaches a vertex x by an unsaturated edge in one path and by a saturated edge in another path, then x belongs to an odd cycle. Alternating paths from u can diverge only when the next edge is unsaturated (leaving vertex a in Example 3.3.14); when the next edge is saturated there is only one choice for it. From the vertex where the paths diverge, the path reaching x on an unsaturated edge has odd length, and the path reaching it on a saturated edge has even length. Together, they form an odd cycle.

3.3.15. Definition. Let M be a matching in a graph G, and let u be an M-unsaturated vertex. A **flower** is the union of two M-alternating paths from u that reach a vertex x on steps of opposite parity (having not done so earlier). The **stem** of the flower is the maximal common initial path (of nonnegative even length). The **blossom** of the flower is the odd cycle obtained by deleting the stem.

In Example 3.3.14, the flower is the full graph except y, the stem is the path u, v, a, and the blossom is the 5-cycle. The horticultural terminology echoes the use of *tree* for the structures given by most search procedures.

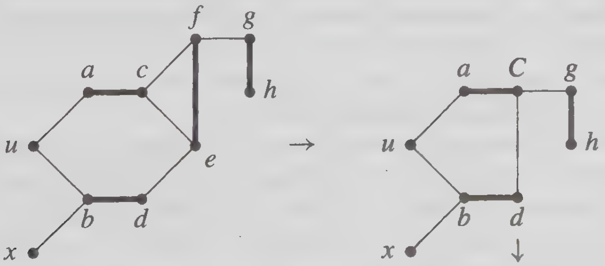
Blossoms do not impede our search. For each vertex z in a blossom, some M-alternating u, z-path reaches z on a saturated edge, found by traveling the proper direction around the blossom to reach z from the stem. We therefore can continue our search along any unsaturated edge from the blossom to a vertex not yet reached. Example 3.3.14 shows such an extension that immediately reaches an unsaturated vertex and completes an M-augmenting path.

Since each vertex of a blossom is saturated by an edge on these paths, no saturated edge emerges from a blossom (except the stem). The effect of these two observations is that we can view the entire blossom as a single "supervertex" reached along the saturated edge at the end of the stem. We search from all vertices of the supervertex blossom simultaneously along unsaturated edges.

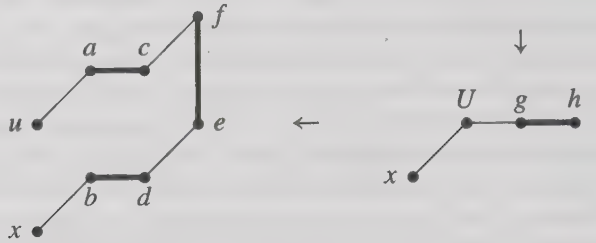
We implement this consolidation by contracting the edges of a blossom B when we find it. The result is a new saturated vertex b incident to the last (saturated) edge of the stem. Its other incident edges are the unsaturated edges joining vertices of B to vertices outside B. We explore from b in the usual way to extend our search. We may later find another blossom containing b; we then contract again. If we find an M-alternating path in the final graph from u to an unsaturated vertex x, then we can undo the contractions to obtain an M-augmenting path to x in the original graph.

Except for the treatment of blossoms, the approach is that of Algorithm 3.2.1 for exploring M-alternating paths. In the corresponding phrasing, T is the set of vertices of the current graph reached along unsaturated edges, and S is the set of vertices reached along saturated edges. The vertices that arise by contracting blossoms belong to S.

3.3.16. Example. Let M be the bold matching in the graph on the left below. We search from the unsaturated vertex u for an M-augmenting path. We first explore the unsaturated edges incident to u, reaching a and b. Since a and b are saturated, we immediately extend the paths along the edges ac and bd. Now $S = \{u, c, d\}$. If we next explore from c, then we find its neighbors e and f along unsaturated edges. Since $ef \in M$, we discover the blossom with vertex set $\{c, e, f\}$. We contract the blossom to obtain the new vertex C, changing S to $\{u, C, d\}$. This yields the graph on the right.



Suppose we now explore from the vertex $C \in S$. Unsaturated edges take us to g and to d. Since g is saturated by the edge gh, we place h in S. Since d is already in S, we have found another blossom. The paths reaching d are u, b, d and u, a, C, d. We contract the blossom, obtaining the new vertex U and the graph on the right below, with $S = \{U, h\}$. We next explore from h, finding nothing new (if we exhaust S without reaching an unsaturated vertex, then there is no M-augmenting path from u). Finally, we explore from U, reaching the unsaturated vertex x.



Having recorded the edge on which we reached each vertex, we can extract an M-augmenting u, x-path. We reached x from U, so we expand the blossom back into $\{u, a, C, d, b\}$ and find that x is reached from U along bx. The path in the blossom U that reaches b on a saturated edge ends with C, d, b. Since C is a blossom in the original graph, we expand C back into $\{c, f, e\}$. Note that d is reached from C by the unsaturated edge ed. The path from the "base" of C that reaches e along a saturated edge is c, f, e. Finally, c was reached from a and a from u, so we obtain the full M-augmenting path u, a, c, f, e, d, b, x. ■

We summarize the steps of the algorithm, glossing over the details of implementation, especially the treatment of contractions.

3.3.17. Algorithm. (Edmonds' Blossom Algorithm [1965a]—sketch).
Input: A graph G, a matching M in G, an M-unsaturated vertex u.
Idea: Explore M-alternating paths from u, recording for each vertex the vertex from which it was reached, and contracting blossoms when found. Maintain sets S and T analogous to those in Algorithm 3.2.1, with S consisting of u and the vertices reached along saturated edges. Reaching an unsaturated vertex yields an augmentation.
Initialization: $S = \{u\}$ and $T = \emptyset$.
Iteration: If S has no unmarked vertex, stop; there is no M-augmenting path from u. Otherwise, select an unmarked $v \in S$. To explore from v, successively consider each $y \in N(v)$ such that $y \notin T$.

If y is unsaturated by M, then trace back from y (expanding blossoms as needed) to report an M-augmenting u, y-path.

If $y \in S$, then a blossom has been found. Suspend the exploration of v and contract the blossom, replacing its vertices in S and T by a single new vertex in S. Continue the search from this vertex in the smaller graph.

Otherwise, y is matched to some w by M. Include y in T (reached from v), and include w in S (reached from y).

After exploring all such neighbors of v, mark v and iterate. ■

We cannot explore all unsaturated vertices simultaneously as in Algorithm 3.2.1, because the membership of vertices in blossoms depends on the choice of the initial unsaturated vertex. Nevertheless, if we find no M-augmenting path from u, then we can delete u from the graph and ignore it in the subsequent search for a maximum matching (Exercise 26).

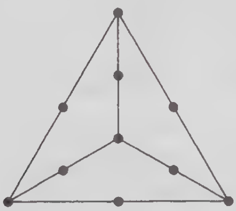
3.3.18. Remark. Edmonds' original algorithm runs in time $O(n^4)$. The implementation in Ahuja–Magnanti–Orlin [1993, p483–494] runs in time $O(n^3)$. This requires (1) appropriate data structures to represent the blossoms and to process contractions, and (2) careful analysis of the number of contractions that can be performed, the time spent exploring edges, and the time spent contracting and expanding blossoms.

The first algorithm solving the maximum matching problem in less than cubic time was the $O(n^{5/2})$ algorithm in Even–Kariv [1975]. The best algorithm now known runs in time $O(n^{1/2}m)$ for a graph with n vertices and m edges (this is faster than $O(n^{5/2})$ for sparse graphs). The algorithm is rather complicated and appears in Micali–Vazirani [1980], with a complete proof in Vazirani [1994].

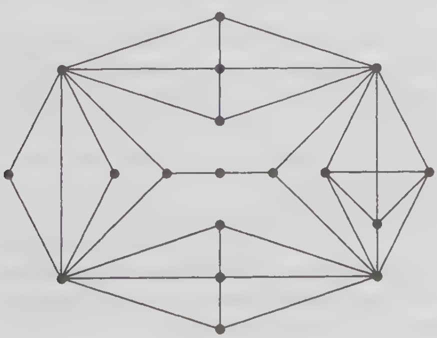
We have not discussed the weighted matching problem for general graphs. Edmonds [1965d] found an algorithm for this, which was implemented in time $O(n^3)$ by Gabow [1975] and by Lawler [1976]. Faster algorithms appear in Gabow [1990] and in Gabow–Tarjan [1989]. ■

EXERCISES

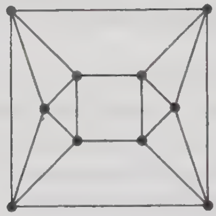
3.3.1. (−) Determine whether the graph below has a 1-factor.



3.3.2. (−) Exhibit a maximum matching in the graph below, and use a result in this section to give a short proof that it has no larger matching.



3.3.3. (−) In the graph drawn below, exhibit a k-factor for each k in $\{0, 1, 2, 3, 4\}$.



3.3.4. (−) Let G be a k-regular bipartite graph. Prove that G can be decomposed into r-factors if and only if r divides k.

3.3.5. (−) Given graphs G and H, determine the number of components and maximum degree of $G \vee H$ in terms of parameters of G and H.

• • • • •

3.3.6. (!) Prove that a tree T has a perfect matching if and only if $o(T - v) = 1$ for every $v \in V(T)$. (Chungphaisan)

3.3.7. (!) For each $k > 1$, construct a k-regular simple graph having no 1-factor.

3.3.8. Prove that if a graph G decomposes into 1-factors, then G has no cut-vertex. Draw a connected 3-regular simple graph that has a 1-factor and has a cut-vertex.

3.3.9. Prove that every graph G has a matching of size at least $n(G)/(1 + \Delta(G))$. (Hint: Apply induction on $e(G)$.) (Weinstein [1963])

3.3.10. (!) For every graph G, prove that $\beta(G) \le 2\alpha'(G)$. For each $k \in \mathbb{N}$, construct a simple graph G with $\alpha'(G) = k$ and $\beta(G) = 2k$.

3.3.11. Let T be a set of vertices in a graph G. Prove that G has a matching that saturates T if and only if for all $S \subseteq V(G)$, the number of odd components of $G - S$ contained in $G[T]$ is at most $|S|$.

3.3.12. (!) *Extension of König–Egerváry Theorem to general graphs.* Given a graph G, let $S_1, \ldots, S_k$ and T be subsets of $V(G)$ such that each S_i has odd size. These sets form a **generalized cover** of G if every edge of G has one endpoint in T or both endpoints in some S_i. The **weight** of a generalized cover is $|T| + \sum \lfloor |S_i|/2 \rfloor$. Let $\beta^*(G)$ be the minimum weight of a generalized cover. Prove that $\alpha'(G) = \beta^*(G)$. (Hint: Apply Corollary 3.3.7. Comment: every vertex cover is a generalized cover, and thus $\beta^*(G) \le \beta(G)$.)

3.3.13. (+) *Tutte's Theorem from Hall's Theorem.* Let G be a graph such that $o(G - S) \le |S|$ for all $S \subseteq V(G)$. Let T be a maximal vertex subset such that $o(G - T) = |T|$.

a) Prove that every component of $G - T$ is odd, and conclude that $T \ne \emptyset$.

b) Let C be a component of $G - T$. Prove that Tutte's Condition holds for every subgraph of C obtained by deleting one vertex. (Hint: Since $C - x$ has even order, a violation requires $o(C - x - S) \ge |S| + 2$.)

c) Let H be a bipartite graph with partite sets T and $\mathbf{C}$, where $\mathbf{C}$ is the set of components of $G - T$. For $t \in T$ and $C \in \mathbf{C}$, put $tC \in E(H)$ if and only if $N_G(t)$ contains a vertex of C. Prove that H satisfies Hall's Condition for a matching that saturates $\mathbf{C}$.

d) Use parts (a), (b), (c), and Hall's Theorem to prove Tutte's 1-factor Theorem by induction on $n(G)$. (Anderson [1971], Mader [1973])

3.3.14. For $k \in \mathbb{N}$, let G be a simple graph such that $\delta(G) \geq k$ and $n(G) \geq 2k$. Prove that $\alpha'(G) \geq k$. (Hint: Apply Corollary 3.3.7.) (Brandt [1994])

3.3.15. Let G be a 3-regular graph with at most two cut-edges. Prove that G has a 1-factor. (Petersen [1891])

3.3.16. (!) Let G be a k-regular graph of even order that remains connected when any $k-2$ edges are deleted. Prove that G has a 1-factor.

3.3.17. With G as in Exercise 3.3.16, use Remark 3.3.5 to prove that every edge of G belongs to some 1-factor. (Comment: This strengthens Exercise 3.3.16.) (Schönberger [1934] for $k = 3$, Berge [1973, p162])

3.3.18. (+) For each odd k greater than 1, construct a graph G with no 1-factor that is k-regular and remains connected when any $k-3$ edges are deleted. (Comment: Thus Exercise 3.3.16 is best possible.)

3.3.19. (!) Let G be a 3-regular simple graph with no cut-edge. Prove that G decomposes into copies of P_4. (Hint: Use Theorem 3.3.9.)

3.3.20. (!) Prove that a 3-regular simple graph has a 1-factor if and only if it decomposes into copies of P_4.

3.3.21. (+) Let G be a $2m$-regular graph, and let T be a tree with m edges Prove that if the diameter of T is less than the girth of G, then G decomposes into copies of T. (Hint: Use Theorem 3.3.9 to give an inductive proof of the stronger result that G has such a decomposition in which each vertex is used once as an image of each vertex of T.) (Häggkvist)

3.3.22. (!) Let G be an X, Y-bigraph. Let H be the graph obtained from G by adding one vertex to Y if $n(G)$ is odd and then adding edges to make Y a clique.

a) Prove that G has a matching of size $|X|$ if and only if H has a 1-factor.

b) Prove that if G satisfies Hall's Condition ($|N(S)| \geq |S|$ for all $S \subseteq X$), then H satisfies Tutte's Condition ($o(H-T) \leq |T|$ for all $T \subseteq V(H)$).

c) Use parts (a) and (b) to obtain Hall's Theorem from Tutte's Theorem.

3.3.23. Let G be a claw-free connected graph of even order.

a) Let T be a spanning tree of G generated by Breadth-First Search (Algorithm 2.3.8). Let x and y be vertices that have a common parent in T other than the root. Prove that x and y must be adjacent.

b) Use part (a) to prove that G has a 1-factor. (Comment: Without part (a), one can simply prove the stronger result that the last edge in a longest path belongs to a 1-factor.) (Sumner [1974a], Las Vergnas [1975])

3.3.24. (!) Let G be a simple graph of even order n having a set S of size k such that $o(G-S) > k$. Prove that G has at most $\binom{k}{2} + k(n-k) + \binom{n-2k-1}{2}$ edges, and that this bound is best possible. Use this to determine the maximum size of a simple n-vertex graph with no 1-factor. (Erdős–Gallai [1961])

3.3.25. A graph G is **factor-critical** if each subgraph $G-v$ obtained by deleting one vertex has a 1-factor. Prove that G is factor-critical if and only if $n(G)$ is odd and $o(G-S) \leq |S|$ for all nonempty $S \subseteq V(G)$. (Gallai [1963a])

3.3.26. (!) Let M be a matching in a graph G, and let u be an M-unsaturated vertex. Prove that if G has no M-augmenting path that starts at u, then u is unsaturated in some maximum matching in G.

3.3.27. ($*$) Assuming that Algorithm 3.3.17 is correct, we develop an algorithmic proof of Tutte's Theorem (Theorem 3.3.3).

a) Let G be a graph with no perfect matching, and let M be a maximum matching in G. Let S and T be the sets generated when running Algorithm 3.3.17 from u. Prove that $|T| < |S| \le o(G - T)$.

b) Use part (a) to prove Theorem 3.3.3.

3.3.28. ($*$) Let $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. Given $f\colon V(G) \to \mathbb{N}_0$, the graph G is f**-soluble** if there exists $w\colon E(G) \to \mathbb{N}_0$ such that $\sum_{uv \in E(G)} w(uv) = f(v)$ for every $v \in V(G)$.

a) Prove that G has an f-factor if and only if the graph H obtained from G by subdividing each edge twice and defining f to be 1 on the new vertices is f-soluble. (This reduces testing for an f-factor to testing f-solubility.)

b) Given G and an $f\colon V(G) \to \mathbb{N}_0$, construct a graph H (with proof) such that G is f-soluble if and only if H has a 1-factor. (Tutte [1954a])

3.3.29. ($*+$) *Tutte's f-factor condition and graphic sequences.* Given $f\colon V(G) \to \mathbb{N}_0$, define $f(S) = \sum_{v \in S} f(v)$ for $S \subseteq V(G)$. For are disjoint subsets S, T of $V(G)$, let $q(S, T)$ denote the number of components Q of $G - S - T$ such that $e(Q, T) + f(V(Q))$ is odd, where $e(Q, T)$ is the number of edges from Q to T. Tutte [1952, 1954a] proved that G has an f-factor if and only if

$$q(S, T) + f(T) - \sum_{v \in T} d_{G-S}(v) \le f(S)$$

for all choices of disjoint subsets $S, T \subset V$.

a) **The Parity Lemma**. Let $\delta(S, T) = f(S) - f(T) + \sum_{v \in T} d_{G-S}(v) - q(S, T)$. Prove that $\delta(S, T)$ has the same parity as $f(V)$ for disjoint sets $S, T \subseteq V(G)$. (Hint: Use induction on $|T|$.)

b) Suppose that $G = K_n$ and $f(v_i) = d_i$, where $\sum d_i$ is even and $d_1 \ge \cdots \ge d_n$. Use the f-factor condition and part (a) to prove that G has an f-factor if and only if $\sum_{i=1}^{k} d_i \le (n - 1 - s)k + \sum_{i=n+1-s}^{n} d_i$ for all k, s with $k + s \le n$.

c) Conclude that $d_1, \dots, d_n \ge 0$ are the vertex degrees of a simple graph if and only if $\sum d_i$ is even and $\sum_{i=1}^{k} d_i \le k(k-1) + \sum_{i=k+1}^{n} \min\{k, d_i\}$ for $1 \le k \le n$. (Erdős–Gallai [1960])

Chapter 4

Connectivity and Paths

4.1. Cuts and Connectivity

A good communication network is hard to disrupt. We want the graph (or digraph) of possible transmissions to remain connected even when some vertices or edges fail. When communication links are expensive, we want to achieve these goals with few edges. Loops are irrelevant for connection, so in this chapter we assume that our graphs and digraphs **have no loops**, especially when considering degree conditions.

CONNECTIVITY

How many vertices must be deleted to disconnect a graph?

4.1.1. Definition. A **separating set** or **vertex cut** of a graph G is a set $S \subseteq V(G)$ such that $G - S$ has more than one component. The **connectivity** of G, written $\kappa(G)$, is the minimum size of a vertex set S such that $G - S$ is disconnected or has only one vertex. A graph G is k-**connected** if its connectivity is at least k.

A graph other than a complete graph is k-connected if and only if every separating set has size at least k. We can view "k-connected" as a structural condition, while "connectivity k" is the solution of an optimization problem.

4.1.2. Example. *Connectivity of K_n and $K_{m,n}$.* Because a clique has no separating set, we need to adopt a convention for its connectivity. This explains the phrase "or has only one vertex" in Definition 4.1.1. We obtain $\kappa(K_n) = n - 1$, while $\kappa(G) \le n(G) - 2$ when G is not a complete graph. With this convention, most general results about connectivity remain valid on complete graphs.

Consider a bipartition X, Y of $K_{m,n}$. Every induced subgraph that has at least one vertex from X and from Y is connected. Hence every separating set of $K_{m,n}$ contains X or Y. Since X and Y themselves are separating sets (or leave only one vertex), we have $\kappa(K_{m,n}) = \min\{m, n\}$. The connectivity of $K_{3,3}$ is 3; the graph is 1-connected, 2-connected, and 3-connected, but not 4-connected. ■

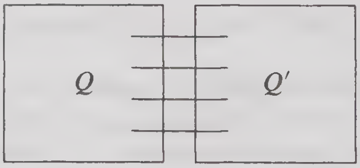
A graph with more than two vertices has connectivity 1 if and only if it is connected and has a cutvertex. A graph with more than one vertex has connectivity 0 if and only if it is disconnected. The 1-vertex graph K_1 is annoyingly inconsistent; it is connected, but for consistency in discussing connectivity we set $\kappa(K_1) = 0$.

4.1.3. Example. *The hypercube Q_k.* For $k \geq 2$, the neighbors of one vertex in Q_k form a separating set, so $\kappa(Q_k) \leq k$. To prove that $\kappa(Q_k) = k$, we show that every vertex cut has size at least k. We use induction on k.

Basis step: $k \in \{0, 1\}$. For $k \leq 1$, Q_k is a clique with $k+1$ vertices and has connectivity k.

Induction step: $k \geq 2$). By the induction hypothesis, $\kappa(Q_{k-1}) = k - 1$. Consider the description of Q_k as two copies Q and Q' of Q_{k-1} plus a matching that joins corresponding vertices in Q and Q' (Example 1.3.8). Let S be a vertex cut in Q_k. If $Q - S$ is connected and $Q' - S$ is connected, then $Q_k - S$ is also connected unless S contains at least one endpoint of every matched pair. This requires $|S| \geq 2^{k-1}$, but $2^{k-1} \geq k$ for $k \geq 2$.

Hence we may assume that $Q - S$ is disconnected, which means that S has at least $k - 1$ vertices in Q, by the induction hypothesis. If S contains no vertices of Q', then $Q' - S$ is connected and all vertices of $Q - S$ have neighbors in $Q' - S$, so $Q_k - S$ is connected. Hence S must also contain a vertex of Q'. This yields $|S| \geq k$, as desired. ■

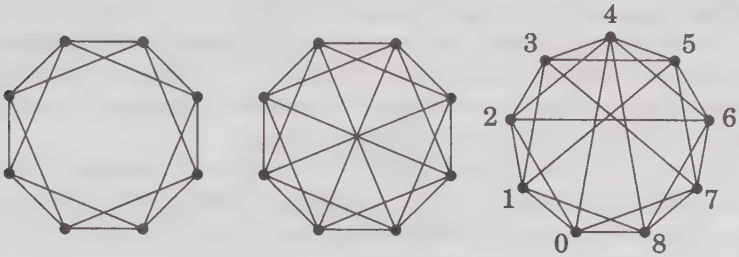


Deleting the neighbors of a vertex disconnects a graph (or leaves only one vertex), so $\kappa(G) \leq \delta(G)$. Equality need not hold; $2K_m$ has minimum degree $m - 1$ but connectivity 0.

Since connectivity k requires $\delta(G) \geq k$, it also requires at least $\lceil kn/2 \rceil$ edges. The k-dimensional cube achieves this bound, but only for the case $n = 2^k$. The bound is best possible whenever $k < n$, as shown by the next example.

4.1.4. Example. *Harary graphs.* Given $k < n$, place n vertices around a circle, equally spaced. If k is even, form $H_{k,n}$ by making each vertex adjacent to the nearest $k/2$ vertices in each direction around the circle. If k is odd and n is even, form $H_{k,n}$ by making each vertex adjacent to the nearest $(k-1)/2$ vertices in each direction and to the diametrically opposite vertex. In each case, $H_{k,n}$ is k-regular.

When k and n are both odd, index the vertices by the integers modulo n. Construct $H_{k,n}$ from $H_{k-1,n}$ by adding the edges $i \leftrightarrow i + (n-1)/2$ for $0 \le i \le (n-1)/2$. The graphs $H_{4,8}$, $H_{5,8}$, and $H_{5,9}$ appear below. ■



4.1.5. Theorem. (Harary [1962a]) $\kappa(H_{k,n}) = k$, and hence the minimum number of edges in a k-connected graph on n vertices is $\lceil kn/2 \rceil$.

Proof: We prove only the even case $k = 2r$, leaving the odd case as Exercise 12. Let $G = H_{k,n}$. Since $\delta(G) = k$, it suffices to prove $\kappa(G) \ge k$. For $S \subseteq V(G)$ with $|S| < k$, we prove that $G - S$ is connected. Consider $u, v \in V(G) - S$. The original circular arrangement has a clockwise u,v-path and a counterclockwise u,v-path along the circle; let A and B be the sets of internal vertices on these two paths.

Since $|S| < k$, the pigeonhole principle implies that in one of $\{A, B\}$, S has fewer than $k/2$ vertices. Since in G each vertex has edges to the next $k/2$ vertices in a particular direction, deleting fewer than $k/2$ consecutive vertices cannot block travel in that direction. Thus we can find a u,v-path in $G - S$ via the set A or B in which S has fewer than $k/2$ vertices. ■

Harary's construction determines the degree conditions that *allow* a graph to be k-connected. Exercise 22 determines the degree conditions that *force* a simple graph to be k-connected. Since it depends on vertex deletions, connectivity is not affected by deleting extra copies of multiple edges. Hence we state degree conditions for k-connectedness only in the context of simple graphs.

4.1.6. Remark. A direct proof of $\kappa(G) \ge k$ considers a vertex cut S and proves that $|S| \ge k$, or it considers a set S with fewer than k vertices and proves that $G - S$ is connected. The indirect approach assumes a cut of size less than k and obtains a contradiction. The indirect proof may be easier to find, but the direct proof may be clearer to state.

Note also that if $k < n(G)$ and G has a vertex cut of size less than k, then G has a vertex cut of size $k-1$ (first delete the cut, then continue deleting vertices until $k-1$ are gone, retaining a vertex in each of two components).

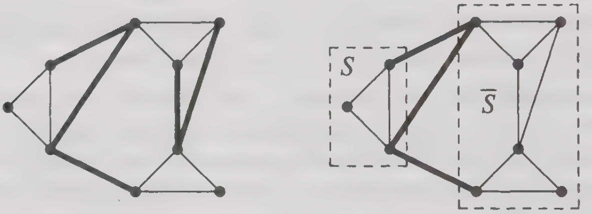
Finally, proving $\kappa(G) = k$ also requires presenting a vertex cut of size k; this is usually the easy part. ■

EDGE-CONNECTIVITY

Perhaps our transmitters are secure and never fail, but our communication links are subject to noise or other disruptions. In this situation, we want to make it hard to disconnect our graph by deleting edges.

4.1.7. Definition. A **disconnecting set** of edges is a set $F \subseteq E(G)$ such that $G - F$ has more than one component. A graph is **k-edge-connected** if every disconnecting set has at least k edges. The **edge-connectivity** of G, written $\kappa'(G)$, is the minimum size of a disconnecting set (equivalently, the maximum k such that G is k-edge-connected).

Given $S, T \subseteq V(G)$, we write $[S, T]$ for the set of edges having one endpoint in S and the other in T. An **edge cut** is an edge set of the form $[S, \overline{S}]$, where S is a nonempty proper subset of $V(G)$ and $\overline{S}$ denotes $V(G) - S$.



disconnecting set edge cut

4.1.8. Remark. *Disconnecting set vs. edge cut.* Every edge cut is a disconnecting set, since $G - [S, \overline{S}]$ has no path from S to $\overline{S}$. The converse is false, since a disconnecting set can have extra edges. Above we show a disconnecting set and an edge cut in bold; see also Exercise 13.

Nevertheless, *every minimal disconnecting set of edges is an edge cut* (when $n(G) > 1$). If $G - F$ has more than one component for some $F \subseteq E(G)$, then for some component H of $G - F$ we have deleted all edges with exactly one endpoint in H. Hence F contains the edge cut $[V(H), \overline{V(H)}]$, and F is not a minimal disconnecting set unless $F = [V(H), \overline{V(H)}]$. ■

The notation for edge-connectivity continues our convention of using a "prime" for an edge parameter analogous to a vertex parameter. Using the same base letter emphasizes the analogy and avoids the confusion of using many different letters – and running out of them.

Deleting one endpoint of each edge in an edge cut F deletes every edge of F. This suggests that $\kappa(G) \le \kappa'(G)$. However, we must be careful not to delete the only vertex of a component of $G - F$ and thereby leave a connected subgraph.

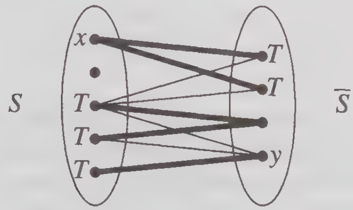
4.1.9. Theorem. (Whitney [1932a]) If G is a simple graph, then

$$\kappa(G) \le \kappa'(G) \le \delta(G).$$

Proof: The edges incident to a vertex v of minimum degree form an edge cut; hence $\kappa'(G) \le \delta(G)$. It remains to show that $\kappa(G) \le \kappa'(G)$.

We have observed that $\kappa(G) \le n(G) - 1$ (see Example 4.1.2). Consider a smallest edge cut $[S, \overline{S}]$. If every vertex of S is adjacent to every vertex of $\overline{S}$, then $\left|[S, \overline{S}]\right| = |S|\left|\overline{S}\right| \ge n(G) - 1 \ge \kappa(G)$, and the desired inequality holds.

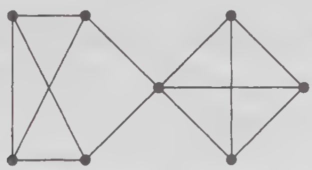
Otherwise, we choose $x \in S$ and $y \in \overline{S}$ with $x \not\leftrightarrow y$. Let T consist of all neighbors of x in $\overline{S}$ and all vertices of $S - \{x\}$ with neighbors in $\overline{S}$. Every x, y-path passes through T, so T is a separating set. Also, picking the edges from x to $T \cap \overline{S}$ and one edge from each vertex of $T \cap S$ to $\overline{S}$ (shown bold below) yields $|T|$ distinct edges of $[S, \overline{S}]$. Thus $\kappa'(G) = \left|[S, \overline{S}]\right| \ge |T| \ge \kappa(G)$. ■



We have seen that $\kappa(G) = \delta(G)$ when G is a complete graph, a biclique, a hypercube, or a Harary graph. By Theorem 4.1.9, also $\kappa'(G) = \delta(G)$ for these graphs. Nevertheless, in many graphs the set of edges incident to a vertex of minimum degree is not a minimum edge cut. The situation $\kappa'(G) < \delta(G)$ is precisely the situation where no minimum edge cut isolates a vertex.

4.1.10. Example. *Possibility of $\kappa < \kappa' < \delta$.* For graph G below, $\kappa(G) = 1$, $\kappa'(G) = 2$, and $\delta(G) = 3$. Note that no minimum edge cut isolates a vertex.

Each inequality can be arbitarily weak. When $G = K_m + K_m$, we have $\kappa(G) = \kappa'(G) = 0$ but $\delta(G) = m - 1$. When G consists of two m-cliques sharing a single vertex, we have $\kappa'(G) = \delta(G) = m - 1$ but $\kappa(G) = 1$. ■

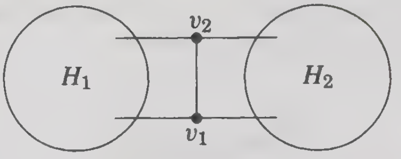


Various conditions force equalities among the parameters; for example, $\kappa'(G) = \delta(G)$ when G has diameter 2 (Exercise 25). For 3-regular graphs, connectivity and edge-connectivity are always equal.

4.1.11. Theorem. If G is a 3-regular graph, then $\kappa(G) = \kappa'(G)$.

Proof: Let S be a minimum vertex cut ($|S| = \kappa(G)$). Since $\kappa(G) \le \kappa'(G)$ always, we need only provide an edge cut of size $|S|$. Let H_1, H_2 be two components of $G - S$. Since S is a minimum vertex cut, each $v \in S$ has a neighbor in H_1 and a neighbor in H_2. Since G is 3-regular, v cannot have two neighbors in H_1 and two in H_2. For each $v \in S$, delete the edge from v to a member of $\{H_1, H_2\}$ where v has only one neighbor.

These $\kappa(G)$ edges break all paths from H_1 to H_2 except in the case below, where a path can enter S via v_1 and leave via v_2. In this case we delete the edge to H_1 for both v_1 and v_2 to break all paths from H_1 to H_2 through $\{v_1, v_2\}$. ■



When $\kappa'(G) < \delta(G)$, a minimum edge cut cannot isolate a vertex. In fact, whenever $\left|[S, \overline{S}]\right| < \delta(G)$, the set S (and also $\overline{S}$) must be much larger than a single vertex. This follows from a simple relationship between the size of the edge cut $[S, \overline{S}]$ and the size of the subgraph induced by S.

4.1.12. Proposition. If S is a set of vertices in a graph G, then

$$\left|[S, \overline{S}]\right| = [\textstyle\sum_{v \in S} d(v)] - 2e(G[S]).$$

Proof: An edge in $G[S]$ contributes twice to $\sum_{v \in S} d(v)$, while an edge in $[S, \overline{S}]$ contributes only once to the sum. Since this counts all contributions, we obtain $\sum_{v \in S} d(v) = \left|[S, \overline{S}]\right| + 2e(G[S])$. ■

4.1.13. Corollary. If G is a simple graph and $\left|[S, \overline{S}]\right| < \delta(G)$ for some nonempty proper subset S of $V(G)$, then $|S| > \delta(G)$.

Proof: By Proposition 4.1.12, we have $\delta(G) > \sum_{v \in S} d(v) - 2e(G[S])$. Using $d(v) \geq \delta(G)$ and $2e(G[S]) \leq |S|\,(|S| - 1)$ yields

$$\delta(G) > |S|\,\delta(G) - |S|\,(|S| - 1).$$

This inequality requires $|S| > 1$, so we can combine the terms involving $\delta(G)$ and cancel $|S| - 1$ to obtain $|S| > \delta(G)$. ■

As a set of edges, an edge cut may contain another edge cut. For example, $K_{1,2}$ has three edge cuts, but one of them contains the other two. The minimal non-empty edge cuts of a graph have useful structural properties.

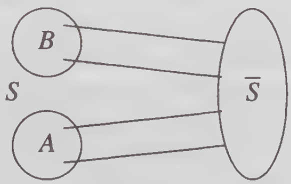
4.1.14. Definition. A **bond** is a minimal nonempty edge cut.

Here "minimal" means that no proper nonempty subset is also an edge cut. We characterize bonds in connected graphs.

4.1.15. Proposition. If G is a connected graph, then an edge cut F is a bond if and only if $G - F$ has exactly two components.

Proof: Let $F = [S, \overline{S}]$ be an edge cut. Suppose first that $G - F$ has exactly two components, and let F' be a proper subset of F. The graph $G - F'$ contains the two components of $G - F$ plus at least one edge between them, making it connected. Hence F is a minimal disconnecting set and is a bond.

For the converse, suppose that $G - F$ has more than two components. Since $G - F$ is the disjoint union of $G[S]$ and $G[\,]$, one of these has at least two components. Assume by symmetry that it is $G[S]$. We can thus write $S = A \cup B$, where no edges join A and B. Now the edge cuts $[A, \overline{A}]$ and $[B, \overline{B}]$ are proper subsets of F, so F is not a bond. ■

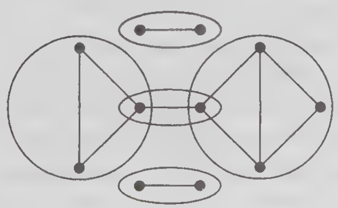


BLOCKS

A connected graph with no cut-vertex need not be 2-connected, since it can be K_1 or K_2. Connected subgraphs without cut-vertices provide a useful decomposition of a graph.

4.1.16. Definition. A **block** of a graph G is a maximal connected subgraph of G that has no cut-vertex. If G itself is connected and has no cut-vertex, then G *is* a block.

4.1.17. Example. *Blocks.* If H is a block of G, then H as a graph has no cut-vertex, but H may contain vertices that are cut-vertices of G. For example, the graph drawn below has five blocks; three copies of K_2, one of K_3, and one subgraph that is neither a cycle nor a clique. ■



4.1.18. Remark. *Properties of blocks.* An edge of a cycle cannot itself be a block, since it is in a larger subgraph with no cut-vertex. Hence an edge is a block if and only if it is a cut-edge; the blocks of a tree are its edges. If a block has more than two vertices, then it is 2-connected. The blocks of a loopless graph are its isolated vertices, its cut-edges, and its maximal 2-connected subgraphs. ■

4.1.19. Proposition. Two blocks in a graph share at most one vertex.

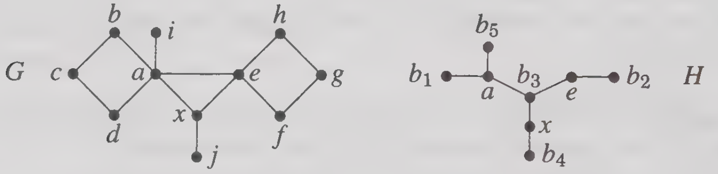
Proof: We use contradiction. Suppose that blocks B_1, B_2 have at least two common vertices. We show that $B_1 \cup B_2$ is a connected subgraph with no cut-vertex, which contradicts the maximality of B_1 and B_2.

When we delete one vertex from B_i, what remains is connected. Hence we retain a path in B_i from every vertex that remains to every vertex of $V(B_1) \cap V(B_2)$ that remains. Since the blocks have at least two common vertices, deleting a single vertex leaves a vertex in the intersection. We retain paths from all vertices to that vertex, so $B_1 \cup B_2$ cannot be disconnected by deleting one vertex. ■

Every edge by itself is a subgraph with no cut-vertex and hence is in a block. We conclude that the blocks of a graph decompose the graph. Blocks in a graph behave somewhat like strong components of a digraph (Definition 1.4.12), but strong components share no vertices (Exercise 1.4.13a). Thus although blocks in a graph decompose the edge set, strong components in a digraph merely partition the vertex set and usually omit edges.

When two blocks of G share a vertex, it must be a cut-vertex of G. The interaction between blocks and cut-vertices is described by a special graph.

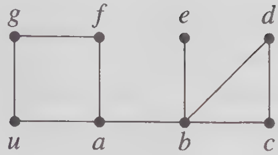
4.1.20.* Definition. The **block-cutpoint graph** of a graph G is a bipartite graph H in which one partite set consists of the cut-vertices of G, and the other has a vertex b_i for each block B_i of G. We include vb_i as an edge of H if and only if $v \in B_i$.



When G is connected, its block-cutpoint graph is a tree (Exercise 34) whose leaves are blocks of G. Thus a graph G that is not a single block has at least two blocks (**leaf blocks**) that each contain exactly one cut-vertex of G.

Blocks can be found using a technique for searching graphs. In **Depth-First Search** (DFS), we explore always from the most recently discovered vertex that has unexplored edges (also called **backtracking**). In contrast, Breadth-First Search (Algorithm 2.3.8) explores from the oldest vertex, so the difference between DFS and BFS is that in DFS we maintain the list of vertices to be searched as a Last-In First-Out "stack" rather than a queue.

4.1.21.* Example. *Depth-First Search.* In the graph below, one depth-first search from u finds the vertices in the order u, a, b, c, d, e, f, g. For both BFS and DFS, the order of discovery depends on the order of exploring edges from a searched vertex. ■



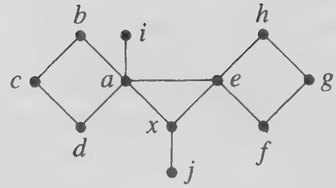
A breath-first or depth-first search from u generates a tree rooted at u; each time exploring a vertex x yields a new vertex v, we include the edge xv. This grows a tree that becomes a spanning tree of the component containing u. Applications of depth-first search rely on a fundamental property of the resulting spanning tree.

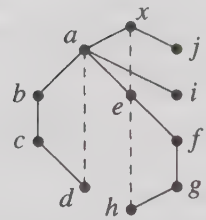
4.1.22.* Lemma. If T is a spanning tree of a connected graph G grown by DFS from u, then every edge of G not in T consists of two vertices v, w such that v lies on the u, w-path in T.

Proof: Let vw be an edge of G, with v encountered before w in the depth-first search. Because vw is an edge, we cannot finish v before w is added to T. Hence w appears somewhere in the subtree formed before finishing v, and the path from w to u contains v. ∎

4.1.23.* Algorithm. (Computing the blocks of a graph)
Input: A connected graph G. (The blocks of a graph are the blocks of its components, which can be found by depth-first search, so we may assume that G is connected.)
Idea: Build a depth-first search tree T of G, discarding portions of T as blocks are identified. Maintain one vertex called ACTIVE.
Initialization: Pick a root $x \in V(H)$; make x ACTIVE; set $T = \{x\}$.
Iteration: Let v denote the current active vertex.
1) If v has an unexplored incident edge vw, then
    1A) If $w \notin V(T)$, then add vw to T, mark vw explored, make w ACTIVE.
    1B) If $w \in V(T)$, then w is an ancestor of v; mark vw explored.
2) If v has no more unexplored incident edges, then
    2A) If $v \neq x$, and w is the parent of v, make w ACTIVE. If no vertex in the current subtree T' rooted at v has an explored edge to an ancestor above w, then $V(T') \cup \{w\}$ is the vertex set of a block; record this information and delete $V(T')$ from T.
    2B) If $v = x$, terminate. ∎

4.1.24.* Example. *Finding blocks.* For the graph below, one depth-first traversal from x visits the other vertices in the order $a, b, c, d, e, f, g, h, i, j$. We find blocks in the order $\{a, b, c, d\}$, $\{e, f, g, h\}$, $\{a, i\}$, $\{x, a, e\}$, $\{x, j\}$. After finding each block, we delete the vertices other than the highest. Exercise 36 requests a proof of correctness. ∎





EXERCISES

4.1.1. (−) Give a proof or a counterexample for each statement below.

a) Every graph with connectivity 4 is 2-connected.

b) Every 3-connected graph has connectivity 3.

c) Every k-connected graph is k-edge-connected.

d) Every k-edge-connected graph is k-connected.

4.1.2. (−) Give a counterexample to the following statement, add a hypothesis to correct it, and prove the corrected statement: If e is a cut-edge of G, then at least one vertex of e is a cut-vertex of G.

4.1.3. (−) Let G be an n-vertex simple graph other than K_n. Prove that if G is not k-connected, then G has a separating set of size $k-1$.

4.1.4. (−) Prove that a graph G is k-connected if and only if $G \vee K_r$ (Definition 3.3.6) is $k+r$-connected.

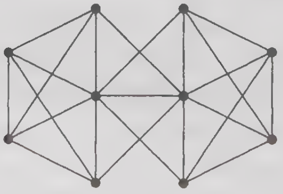
4.1.5. (−) Let G be a connected graph with at least three vertices. Form G' from G by adding an edge with endpoints x, y whenever $d_G(x, y) = 2$. Prove that G' is 2-connected.

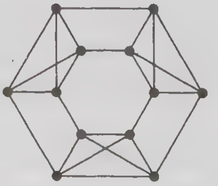
4.1.6. (−) For a graph G with blocks $B_1, \ldots, B_k$, prove that $n(G) = (\sum_{i=1}^{k} n(B_i)) - k + 1$.

4.1.7. (−) Obtain a formula for the number of spanning trees of a connected graph in terms of the numbers of spanning trees of its blocks.

• • • • •

4.1.8. Determine $\kappa(G)$, $\kappa'(G)$, and $\delta(G)$ for each graph G drawn below.





4.1.9. For each choice of integers k, l, m with $0 < k \le l \le m$, construct a simple graph G with $\kappa(G) = k$, $\kappa'(G) = l$, and $\delta(G) = m$. (Chartrand–Harary [1968])

4.1.10. (!) Find (with proof) the smallest 3-regular simple graph having connectivity 1.

4.1.11. Prove that $\kappa'(G) = \kappa(G)$ when G is a simple graph with $\Delta(G) \le 3$.

4.1.12. Let n, k be positive integers with n even, k odd, and $n > k > 1$. Let G be the k-regular simple graph formed by placing n vertices on a circle and making each vertex adjacent to the opposite vertex and to the $(k-1)/2$ nearest vertices in each direction. Prove that $\kappa(G) = k$. (Harary [1962a])

4.1.13. In $K_{m,n}$, let S consist of a vertices from one partite set and b from the other.

a) Compute $\left|[S, \overline{S}]\right|$ in terms of a, b, m, n.

b) Use part (a) to prove numerically that $\kappa'(K_{m,n}) = \min\{m, n\}$.

c) Prove that every set of seven edges in $K_{3,3}$ is a disconnecting set, but no set of seven edges is an edge cut.

4.1.14. (!) Let G be a connected graph in which for every edge e, there are cycles C_1 and C_2 containing e whose only common edge is e. Prove that G is 3-edge-connected. Use this to show that the Petersen graph is 3-edge-connected.

4.1.15. (!) Use Proposition 4.1.12 and Theorem 4.1.11 to prove that the Petersen graph is 3-connected.

4.1.16. Use Proposition 4.1.12 to prove that the Petersen graph has an edge cut of size m if and only if $3 \le m \le 12$. (Hint: Consider $\left|[S, \overline{S}]\right|$ for $1 \le |S| \le 5$.)

4.1.17. Prove that deleting an edge cut of size 3 in the Petersen graph isolates a vertex.

4.1.18. Let G be a triangle-free graph with minimum degree at least 3. Prove that if $n(G) \le 11$, then G is 3-edge-connected. Show that this inequality is sharp by finding a 3-regular bipartite graph with 12 vertices that is not 3-edge-connected. (Galvin)

4.1.19. Prove that $\kappa(G) = \delta(G)$ if G is simple and $\delta(G) \ge n(G) - 2$. Prove that this is best possible for each $n \ge 4$ by constructing a simple n-vertex graph with minimum degree $n-3$ and connectivity less than $n-3$.

4.1.20. (!) Let G be a simple n-vertex graph with $n/2 - 1 \le \delta(G) \le n-2$. Prove that G is k-connected for all k with $k \le 2\delta(G) + 2 - n$. Prove that this is best possible for all $\delta \ge n/2 - 1$ by constructing a simple n-vertex graph with minimum degree δ that is not k-connected for $k = 2\delta + 3 - n$. (Comment: Proposition 1.3.15 is the special case of this when $\delta(G) = (n-1)/2$.)

4.1.21. (+) Let G be a simple n-vertex graph with $n \ge k + l$ and $\delta(G) \ge \frac{n+l(k-2)}{l+1}$. Prove that if $G - S$ has more than l components, then $|S| \ge k$. Prove that the hypothesis on $\delta(G)$ is best possible for $n \ge k + l$ by constructing an appropriate n-vertex graph with minimum degree $\left\lfloor \frac{n+l(k-2)-1}{l+1} \right\rfloor$. (Comment: This generalizes Exercise 4.1.20.)

4.1.22. (!) *Sufficient condition for $k+1$-connected graphs.* (Bondy [1969])

a) Let G be a simple n-vertex graph with vertex degrees $d_1 \le \cdots \le d_n$. Prove that if $d_j \ge j + k$ whenever $j \le n - 1 - d_{n-k}$, then G is $k+1$-connected. (Comment: Exercise 1.3.64 is the special case of this when $k = 0$.)

b) Suppose that $0 \le j + k \le n$. Construct an n-vertex graph G such that $\kappa(G) \le k$ and G has j vertices of degree $j+k-1$, has $n-j-k$ vertices of degree $n-j-1$, and has k vertices of degree $n-1$. In what sense does this show that part (a) is best possible?

4.1.23. (!) Let G be an r-connected graph of even order having no $K_{1,r+1}$ as an induced subgraph. Prove that G has a 1-factor. (Sumner [1974b])

4.1.24. (!) *Degree conditions for $\kappa' = \delta$.* Let G be a simple n-vertex graph. Use Corollary 4.1.13 to prove the following statements.

a) If $\delta(G) \ge \lfloor n/2 \rfloor$, then $\kappa'(G) = \delta(G)$. Prove this best possible by constructing for each $n \ge 3$ a simple n-vertex graph with $\delta(G) = \lfloor n/2 \rfloor - 1$ and $\kappa'(G) < \delta(G)$.

b) If $d(x) + d(y) \geq n - 1$ whenever $x \nleftrightarrow y$, then $\kappa'(G) = \delta(G)$. Prove that this is best possible by constructing for each $n \geq 4$ and $\delta(G) = m \leq n/2 - 1$ an n-vertex graph G with $\kappa'(G) < \delta(G) = m$ in which $d(x) + d(y) \geq n - 2$ whenever $x \nleftrightarrow y$.

4.1.25. (!) $\kappa'(G) = \delta(G)$ *for diameter 2.* Let G be a simple graph with diameter 2, and let $[S, \overline{S}]$ be a minimum edge cut with $|S| \leq |\overline{S}|$.

a) Prove that every vertex of S has a neighbor in $\overline{S}$.

b) Use part (a) and Corollary 4.1.13 to prove that $\kappa'(G) = \delta(G)$. (Plesník [1975])

4.1.26. (!) Let F be a set of edges in G. Prove that F is an edge cut if and only if F contains an even number of edges from every cycle in G. For example, when $G = C_n$, every even subset of the edges is an edge cut, but no odd subset is an edge cut. (Hint: For sufficiency, the task is to show that the components of $G - F$ can be grouped into two nonempty collections so that every edge of F has an endpoint in each collection.)

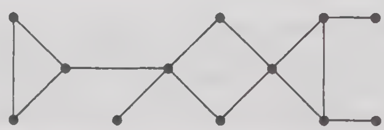
4.1.27. (!) Let $[S, \overline{S}]$ be an edge cut. Prove that there is a set of pairwise edge-disjoint bonds whose union (as edge sets) is $[S, \overline{S}]$. (Note: This is trivial if $[S, \overline{S}]$ is itself a bond.)

4.1.28. (!) Prove that the symmetric difference of two different edge cuts is an edge cut. (Hint: Draw a picture illustrating the two edge cuts and use it to guide the proof.)

4.1.29. (!) Let H be a spanning subgraph of a connected graph G. Prove that H is a spanning tree if and only if the subgraph $H^* = G - E(H)$ is a maximal subgraph that contains no bond. (Comment: See Section 8.2 for a more general context.)

4.1.30. (−) Let G be the simple graph with vertex set $\{1, \ldots, 11\}$ defined by $i \leftrightarrow j$ if and only if i, j have a common factor bigger than 1. Determine the blocks of G.

4.1.31. A **cactus** is a connected graph in which every block is an edge or a cycle. Prove that the maximum number of edges in a simple n-vertex cactus is $\lfloor 3(n-1)/2 \rfloor$. (Hint: $\lfloor x \rfloor + \lfloor y \rfloor \leq \lfloor x + y \rfloor$.)



4.1.32. Prove that every vertex of a graph has even degree if and only if every block is Eulerian.

4.1.33. Prove that a connected graph is k-edge-connected if and only if each of its blocks is k-edge-connected.

4.1.34. (!) *The block-cutpoint graph* (see Definition 4.1.20). Let H be the block-cutpoint graph of a graph G that has a cut-vertex. (Harary–Prins [1966])

a) Prove that H is a forest.

b) Prove that G has at least two blocks each of which contains exactly one cut-vertex of G.

c) Prove that a graph G with k components has exactly $k + \sum_{v \in V(G)} (b(v) - 1)$ blocks, where $b(v)$ is the number of blocks containing v.

d) Prove that every graph has fewer cut-vertices than blocks.

4.1.35. Let H and H' be two maximal k-connected subgraphs of a graph G. Prove that they have at most $k - 1$ common vertices. (Harary–Kodama [1964])

4.1.36. Prove that Algorithm 4.1.23 correctly computes blocks of graphs.

4.1.37. Develop an algorithm to compute the strong components of a digraph. Prove that it works. (Hint: Model the algorithm on Algorithm 4.1.23).

4.2. k-Connected Graphs

A communication network is fault-tolerant if it has alternative paths between vertices: the more disjoint paths, the better. In this section, we prove that this alternative measure of connection is essentially the same as k-connectedness. When $k = 1$, the definition already states that a graph G is 1-connected if and only if each pair of vertices is connected by a path. For larger k the equivalence is more subtle.

2-CONNECTED GRAPHS

We begin by characterizing 2-connected graphs.

4.2.1. Definition. Two paths from u to v are **internally disjoint** if they have no common internal vertex.

4.2.2. Theorem. (Whitney [1932a]) A graph G having at least three vertices is 2-connected if and only if for each pair $u, v \in V(G)$ there exist internally disjoint u, v-paths in G.

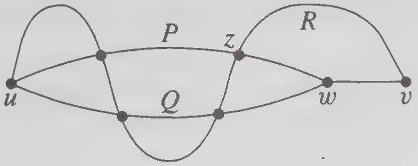
Proof: *Sufficiency.* When G has internally disjoint u, v-paths, deletion of one vertex cannot separate u from v. Since this condition is given for every pair u, v, deletion of one vertex cannot make any vertex unreachable from any other. We conclude that G is 2-connected.

Necessity. Suppose that G is 2-connected. We prove by induction on $d(u, v)$ that G has internally disjoint u, v-paths.

Basis step ($d(u, v) = 1$). When $d(u, v) = 1$, the graph $G - uv$ is connected, since $\kappa'(G) \geq \kappa(G) \geq 2$. A u, v-path in $G - uv$ is internally disjoint in G from the u, v-path formed by the edge uv itself.

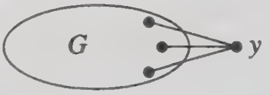
Induction step ($d(u, v) > 1$). Let $k = d(u, v)$. Let w be the vertex before v on a shortest u, v-path; we have $d(u, w) = k - 1$. By the induction hypothesis, G has internally disjoint u, w-paths P and Q. If $v \in V(P) \cup V(Q)$, then we find the desired paths in the cycle $P \cup Q$. Suppose not.

Since G is 2-connected, $G - w$ is connected and contains a u, v-path R. If R avoids P or Q, we are done, but R may share internal vertices with both P and Q. Let z be the last vertex of R (before v) belonging to $P \cup Q$. By symmetry, we may assume that $z \in P$. We combine the u, z-subpath of P with the z, v-subpath of R to obtain a u, v-path internally disjoint from $Q \cup wv$. ■



4.2.3. Lemma. (Expansion Lemma) If G is a k-connected graph, and G' is obtained from G by adding a new vertex y with at least k neighbors in G, then G' is k-connected.

Proof: We prove that a separating set S of G' must have size at least k. If $y \in S$, then $S - \{y\}$ separates G, so $|S| \geq k + 1$. If $y \notin S$ and $N(y) \subseteq S$, then $|S| \geq k$. Otherwise, y and $N(y) - S$ lie in a single component of $G' - S$. Thus again S must separate G and $|S| \geq k$. ■



4.2.4. Theorem. For a graph G with at least three vertices, the following conditions are equivalent (and characterize 2-connected graphs).

A) G is connected and has no cut-vertex.
B) For all $x, y \in V(G)$, there are internally disjoint x, y-paths.
C) For all $x, y \in V(G)$, there is a cycle through x and y.
D) $\delta(G) \geq 1$, and every pair of edges in G lies on a common cycle.

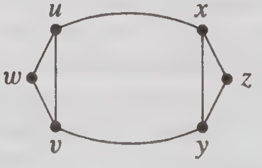
Proof: Theorem 4.2.2 proves A⇔B.

For B⇔C, note that cycles containing x and y correspond to pairs of internally disjoint x, y-paths.

For D⇒C, the condition $\delta(G) \geq 1$ implies that vertices x and y are not isolated; we then apply the last part of D to edges incident to x and y. If there is only one such edge, then we use it and any edge incident to a third vertex.

To complete the proof, we assume that G satisfies the equivalent properties A and C and then derive D. Since G is connected, $\delta(G) \geq 1$. Now consider two edges uv and xy. Add to G the vertices w with neighborhood $\{u, v\}$ and z with neighborhood $\{x, y\}$. Since G is 2-connected, the Expansion Lemma (Lemma 4.2.3) implies that the resulting graph G' is 2-connected.

Hence condition C holds in G', so w and z lie on a cycle C in G'. Since w, z each have degree 2, C must contain the paths u, w, v and x, z, y but not the edges uv or xy. Replacing the paths u, w, v and x, z, y in C with the edges uv and xy yields the desired cycle through uv and xy in G. ■



4.2.5. Definition. In a graph G, **subdivision** of an edge uv is the operation of replacing uv with a path u, w, v through a new vertex w.

4.2.6. Corollary. If G is 2-connected, then the graph G' obtained by subdividing an edge of G is 2-connected.

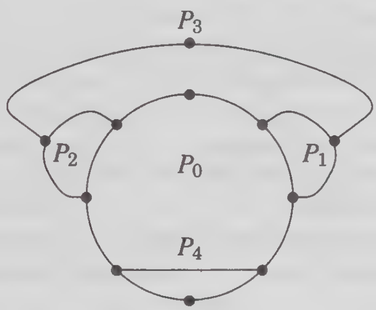
Proof: Let G' be formed from G by adding vertex w to subdivide uv. To show that G' is 2-connected, it suffices to find a cycle through arbitrary edges e, f of G' (by Theorem 4.2.4D).

Since G is 2-connected, any two edges of G lie on a common cycle (Theorem 4.2.4D). When our given edges e, f of G' lie in G, a cycle through them in G is also in G', unless it uses uv, in which case we modify the cycle. Here "modify the cycle" means "replace the edge uv with the u, v-path of length 2 through w".

When $e \in E(G)$ and $f \in \{uw, wv\}$, we modify a cycle passing through e and uv in G. When $\{e, f\} = \{uw, wv\}$, we modify a cycle through uv. ∎

The class of 2-connected graphs has a characterization that expresses the construction of each such graph from a cycle and paths.

4.2.7. Definition. An **ear** of a graph G is a maximal path whose internal vertices have degree 2 in G. An **ear decomposition** of G is a decomposition $P_0, \ldots, P_k$ such that P_0 is a cycle and P_i for $i \geq 1$ is an ear of $P_0 \cup \cdots \cup P_i$.

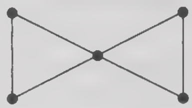


4.2.8. Theorem. (Whitney [1932a]) A graph is 2-connected if and only if it has an ear decomposition. Furthermore, every cycle in a 2-connected graph is the initial cycle in some ear decomposition.

Proof: *Sufficiency.* Since cycles are 2-connected, it suffices to show that adding an ear preserves 2-connectedness. Let u, v be the endpoints of an ear P to be added to a 2-connected graph G. Adding an edge cannot reduce connectivity, so $G + uv$ is 2-connected. A succession of edge subdivisions converts $G + uv$ into the graph $G \cup P$ in which P is an ear; by Corollary 4.2.6, each subdivision preserves 2-connectedness.

Necessity. Given a 2-connected graph G, we build an ear decomposition of G from a cycle C in G. Let $G_0 = C$. Let G_i be a subgraph obtained by successively adding i ears. If $G_i \neq G$, then we can choose an edge uv of $G - E(G_i)$ and an edge $xy \in E(G_i)$. Because G is 2-connected, uv and xy lie on a common cycle C'. Let P be the path in C' that contains uv and exactly two vertices of G_i, one at each end of P. Now P can be added to G_i to obtain a larger subgraph G_{i+1} in which P is an ear. The process ends only by absorbing all of G. ∎

Every 2-connected graph is 2-edge-connected, but the converse does not hold. Recall that the bowtie is the graph consisting of two triangles sharing one common vertex; it is 2-edge-connected but not 2-connected. Since more graphs are 2-edge-connected, decomposition of 2-edge-connected graphs needs a more general operation. The proof is like that of Theorem 4.2.8.



4.2.9. Definition. A **closed ear** in a graph G is a cycle C such that all vertices of C except one have degree 2 in G. A **closed-ear decomposition** of G is a decomposition $P_0, \dots, P_k$ such that P_0 is a cycle and P_i for $i \geq 1$ is either an (open) ear or a closed ear in G.

4.2.10. Theorem. A graph is 2-edge-connected if and only if it has a closed-ear decomposition, and every cycle in a 2-edge-connected graph is the initial cycle in some such decomposition.

Proof: *Sufficiency.* Cut-edges are the edges not on cycles (Theorem 1.2.14), so a connected graph is 2-edge-connected if and only if every edge lies on a cycle. The initial cycle is 2-edge-connected. When we add a closed ear, its edges form a cycle. When we add an open ear P to a connected graph G, a path in G connecting the endpoints of P completes a cycle containing all edges of P. In each case, the new graph also is connected. Thus adding an open or closed ear preserves 2-edge-connectedness.

Necessity. Given a 2-edge-connected graph G, let P_0 be a cycle in G. Consider a closed-ear decomposition $P_0, \dots, P_i$ of a subgraph G_i of G. When $G_i \neq G$, we find an ear to add. Since G is connected, there is an edge $uv \in E(G) - E(G_i)$ with $u \in V(G_i)$. Since G is 2-edge-connected, uv lies on a cycle C. Follow C until it returns to $V(G_i)$, forming up to this point a path or cycle P. Adding P to G_i yields a larger subgraph G_{i+1} in which P is an open or closed ear. The process ends only by absorbing all of G. ■

CONNECTIVITY OF DIGRAPHS

Our results about k-connected and k-edge-connected graphs will apply as well for digraphs, where we use analogous terminology.

4.2.11. Definition. A **separating set** or **vertex cut** of a digraph D is a set $S \subseteq V(D)$ such that $D - S$ is not strongly connected. A digraph is **k-connected** if every vertex cut has at least k vertices. The minimum size of a vertex cut is the **connectivity** $\kappa(D)$.

For vertex sets S, T in a digraph D, let $[S, T]$ denote the set of edges with tail in S and head in T. An **edge cut** is the set $[S, \overline{S}]$ for some $\emptyset \neq$

$S \subset V(D)$. A digraph is k-**edge-connected** if every edge cut has at least k edges. The minimum size of an edge cut is the **edge-connectivity** $\kappa'(D)$.

4.2.12. Remark. Because $|[S, \overline{S}]|$ is the number of edges leaving S, we can restate the definition of edge-connectivity as follows: A graph or digraph G is k-edge-connected if and only if for every nonempty proper vertex subset S, there are at least k edges in G leaving S.

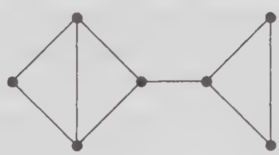
Note that $[S, T]$ is the set of edge from S to T. The meaning of this depends on whether we are discussing a graph or a digraph. In a graph, we take all edges that have endpoints in both sets. In a digraph, we take only the edges with tail in S and head in T. ∎

Strong digraphs are similar to 2-edge-connected graphs.

4.2.13. Proposition. Adding a (directed) ear to a strong digraph produces a larger strong digraph.

Proof: By Remark 4.2.12, a digraph is strong if and only if for every nonempty vertex subset there is a departing edge. If we add an open ear or closed ear P to a strong digraph D, then for every set S with $\emptyset \subset S \subset V(D)$ we already have an edge from S to $V(D) - S$. We need only consider sets that don't intersect $V(D)$ and sets that contain all of $V(D)$ but not all of $V(P)$. For every such set, there is an edge leaving it along P. ∎

When can the streets in a road network all be made one-way without making any location unreachable from some other location? In other words, when does a graph have a strong orientation? The graph below does not. The obvious necessary conditions are sufficient.



4.2.14. Theorem. (Robbins [1939]) A graph has a strong orientation if and only if it is 2-edge-connected.

Proof: *Necessity.* If a graph G is disconnected, then some vertices cannot reach others in any orientation. If G has a cut-edge xy oriented from x to y in an orientation D, then y cannot reach x in D. Hence G must be connected and have no cut-edge.

Sufficiency. When G is 2-edge-connected, it has a closed-ear decomposition. We orient the initial cycle consistently to obtain a strong digraph. As we add each new ear and direct it consistently, Proposition 4.2.13 guarantees that we still have a strong digraph. ∎

Robbins' Theorem generalizes for all k. When G has a k-edge-connected orientation, Remark 4.2.12 implies that G must be $2k$-edge-connected. Nash-Williams [1960] proved that this obvious necessary condition is also sufficient: a graph has a k-edge-connected orientation if and only if it is $2k$-edge-connected. This is easy when G is Eulerian (Exercise 21), but the general case is difficult (see Exercises 36–38). A thorough discussion of this and other orientation theorems appears in Frank [1993].

k-CONNECTED AND k-EDGE-CONNECTED GRAPHS

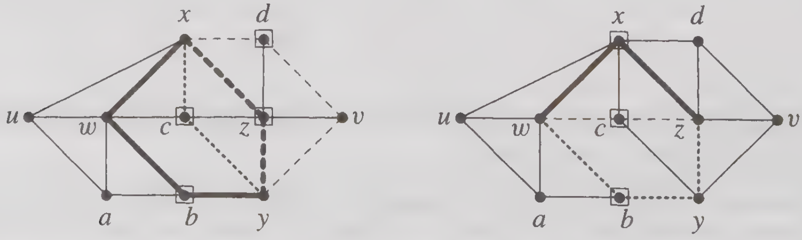
We have introduced two measures of good connection: invulnerability to deletions and multiplicity of alternative paths. Extending Whitney's Theorem, we show that these two notions are the same, for both vertex deletions and edge deletions, and for both graphs and digraphs.

We first discuss the "local" problem of x, y-paths for a fixed pair $x, y \in V(G)$. These definitions hold both for graphs and for digraphs.

4.2.15. Definition. Given $x, y \in V(G)$, a set $S \subseteq V(G) - \{x, y\}$ is an x, y-**separator** or x, y-**cut** if $G - S$ has no x, y-path. Let $\kappa(x, y)$ be the minimum size of an x, y-cut. Let $\lambda(x, y)$ be the maximum size of a set of pairwise internally disjoint x, y-paths. For $X, Y \subseteq V(G)$, an X, Y-**path** is a path having first vertex in X, last vertex in Y, and no other vertex in $X \cup Y$.

An x, y-cut must contain an internal vertex of every x, y-path, and no vertex can cut two internally disjoint x, y-paths. Therefore, always $\kappa(x, y) \geq \lambda(x, y)$. Thus the problems of finding the smallest cut and the largest set of paths are dual problems, like the duality between matching and covering in Chapter 3.

4.2.16. Example. In the graph G below, the set $S = \{b, c, z, d\}$ is an x, y-cut of size 4; thus $\kappa(x, y) \leq 4$. As shown on the left, G has four pairwise internally disjoint x, y-paths; thus $\lambda(x, y) \geq 4$. Since $\kappa(x, y) \geq \lambda(x, y)$ always, we have $\kappa(x, y) = \lambda(x, y) = 4$.



Consider also the pair w, z. As shown on the right, $\kappa(w, z) = \lambda(w, z) = 3$, with $\{b, c, x\}$ being a minimum w, z-cut. The graph G is 3-connected; for every pair $u, v \in V(G)$, we can find three pairwise internally disjoint u, v-paths.

From the equality for internally disjoint paths, we will obtain an analogous equality for edge-disjoint paths. Although $\kappa(w, z) = 3$ above, it takes four edges to break all w, z-paths, and there are four pairwise edge-disjoint w, z-paths. ■

What we call Menger's Theorem states that the local equality $\kappa(x, y) = \lambda(x, y)$ always holds. The global statement for connectivity and analogous results for edge-connectivity and digraphs were observed by others. All are considered forms of Menger's Theorem. More than 15 proofs of Menger's Theorem have been published, some yielding stronger results, some incorrect. (A gap in Menger's original argument was later repaired by König.)

4.2.17. Theorem. (Menger [1927]) If x, y are vertices of a graph G and $xy \notin E(G)$, then the minimum size of an x, y-cut equals the maximum number of pairwise internally disjoint x, y-paths.

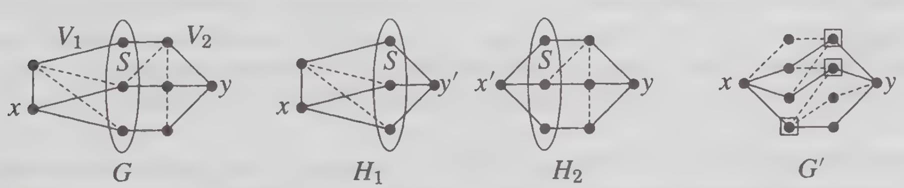
Proof: An x, y-cut must contain an internal vertex from each path in a set of pairwise internally disjoint x, y-paths. These vertices must be distinct, so $\kappa(x, y) \geq \lambda(x, y)$.

To prove equality, we use induction on $n(G)$. Basis step: $n(G) = 2$. Here $xy \notin E(G)$ yields $\kappa(x, y) = \lambda(x, y) = 0$. Induction step: $n(G) > 2$. Let $k = \kappa_G(x, y)$. We construct k pairwise internally disjoint x, y-paths. Note that since $N(x)$ and $N(y)$ are x, y-cuts, no minimum cut properly contains $N(x)$ or $N(y)$.

Case 1: G has a minimum x, y-cut S other than $N(x)$ or $N(y)$. To obtain the k desired paths, we combine x, S-paths and S, y-paths obtained from the induction hypothesis (as formed by solid edges shown below). Let V_1 be the set of vertices on x, S-paths, and let V_2 be the set of vertices on S, y-paths. We claim that $S = V_1 \cap V_2$. Since S is a minimal x, y-cut, every vertex of S lies on an x, y-path, and hence $S \subseteq V_1 \cap V_2$. If $v \in (V_1 \cap V_2)$, then following the x, v-portion of some x, S-path and then the v, y-portion of some S, y-path yields an x, y-path that avoids the x, y-cut S. This is impossible, so $S = V(G_1) \cap V(G_2)$. By the same argument, V_1 omits $N(y) - S$ and V_2 omits $N(x) - S$.

Form H_1 by adding to $G[V_1]$ a vertex y' with edges from S. Form H_2 by adding to $G[V_2]$ a vertex x' with edges to S. Every x, y-path in G starts with an x, S-path (contained in H_1), so every x, y'-cut in H_1 is an x, y-cut in G. Therefore, $\kappa_{H_1}(x, y') = k$, and similarly $\kappa_{H_2}(x', y) = k$.

Since V_1 omits $N(y) - S$ and V_2 omits $N(x) - S$, both H_1 and H_2 are smaller than G. Hence the induction hypothesis yields $\lambda_{H_1}(x, y') = k = \lambda_{H_2}(x', y)$. Since $V_1 \cap V_2 = S$, deleting y' from the k paths in H_1 and x' from the k paths in H_2 yields the desired x, S-paths and S, y-paths in G that combine to form k pairwise internally disjoint x, y-paths in G.



Case 1 Case 2

Case 2. Every minimum x, y-cut is $N(x)$ or $N(y)$. Again we construct the k desired paths. In this case, every vertex outside $\{x \cup N(x) \cup N(y) \cup y\}$ is in no minimum x, y-cut. If G has such a vertex v, then $\kappa_{G-v}(x, y) = k$, and applying the induction hypothesis to $G - v$ yields the desired x, y-paths in G. Also, if there exists $u \in N(x) \cap N(y)$, then u appears in every x, y-cut, and $\kappa_{G-u}(x, y) = k - 1$. Now applying the induction hypothesis to $G - v$ yields $k - 1$ paths to combine with the path x, v, y.

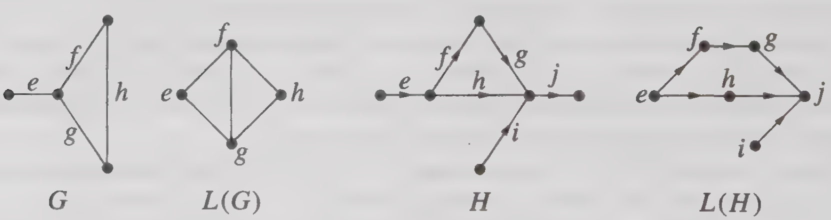
We may thus assume that $N(x)$ and $N(y)$ partition $V(G) - \{x, y\}$. Let G' be the bipartite graph with bipartition $N(x), N(y)$ and edge set $[N(x), N(y)]$. Every x, y-path in G uses some edge from $N(x)$ to $N(y)$, so the x, y-cuts in G are precisely the vertex covers of G'. Hence $\beta(G') = k$. By the König–Egerváry Theorem, G' has a matching of size k. These k edges yield k pairwise internally disjoint x, y-paths of length 3. ■

Case 2 is needed in the proof because when $S = N(x)$, the induction hypothesis cannot be used to obtain the S, y-paths.

The statement of Theorem 4.2.17 makes sense also for digraphs. The proof of the digraph version is exactly the same; we only need to replace $N(x)$ and $N(y)$ with $N^+(x)$ and $N^-(y)$ throughout.

We next develop the analogue of Theorem 4.2.17 for edge-disjoint paths, which we prove by applying Theorem 4.2.17 to a transformed graph. The main part of the transformation is an operation that we will use again in Chapter 7.

4.2.18. Definition. The **line graph** of a graph G, written $L(G)$, is the graph whose vertices are the edges of G, with $ef \in E(L(G))$ when $e = uv$ and $f = vw$ in G. Substituting "digraph" for "graph" in this sentence yields the definition of **line digraph**. For graphs, e and f share a vertex; for digraphs, the head of e must be the tail of f.

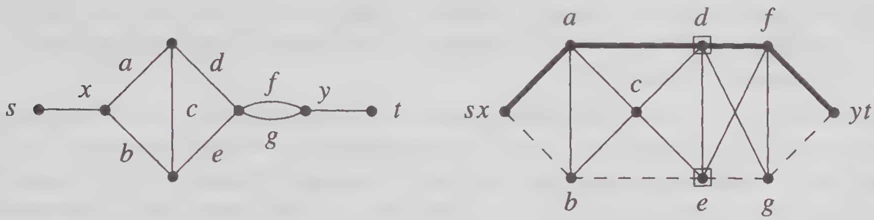


When disconnecting y from x by deleting edges, we use notation analogous to that of Definition 4.2.15: $\lambda'(x, y)$ is the maximum size of a set of pairwise edge-disjoint x, y-paths, and $\kappa'(x, y)$ is the minimum number of edges whose deletion makes y unreachable from x. Elias–Feinstein–Shannon [1956] and Ford–Fulkerson [1956] proved that always $\lambda'(x, y) = \kappa'(x, y)$ (using the methods of Section 4.3). We allow multiple edges and allow $xy \in E(G)$.

4.2.19. Theorem. If x and y are distinct vertices of a graph or digraph G, then the minimum size of an x, y-disconnecting set of edges equals the maximum number of pairwise edge-disjoint x, y-paths.

Proof: Modify G to obtain G' by adding two new vertices s, t and two new edges sx and yt. This does not change $\kappa'(x, y)$ or $\lambda'(x, y)$, and we can think of each path as starting from the edge sx and ending with the edge yt. A set of edges disconnects y from x in G if and only if the corresponding vertices of $L(G')$ form an sx, yt-cut. Similarly, edge-disjoint x, y-paths in G become internally disjoint sx, yt-paths in $L(G')$, and vice versa. Since $x \neq y$, we have no edge from sx to ty in $L(G')$. Applying Theorem 4.2.17 to $L(G')$ yields

$$\kappa'_G(x, y) = \kappa_{L(G')}(sx, yt) = \lambda_{L(G')}(sx, yt) = \lambda'_G(x, y). \quad \blacksquare$$



The global version for k-connected graphs, observed first by Whitney [1932a], is also commonly called Menger's Theorem. The global versions for edges and digraphs appeared in Ford–Fulkerson [1956].

4.2.20. Lemma. Deletion of an edge reduces connectivity by at most 1.

Proof: We discuss only graphs; the argument for digraphs is similar (Exercise 7). Since every separating set of G is a separating set of $G - xy$, we have $\kappa(G - xy) \leq \kappa(G)$. Equality holds unless $G - xy$ has a separating set S that is smaller than $\kappa(G)$ and hence is not a separating set of G. Since $G - S$ is connected, $G - xy - S$ has two components $G[X]$ and $G[Y]$, with $x \in X$ and $y \in Y$. In $G - S$, the only edge joining X and Y is xy.

If $|X| \geq 2$, then $S \cup \{x\}$ is a separating set of G, and $\kappa(G) \leq \kappa(G - xy) + 1$. If $|Y| \geq 2$, then again the inequality holds. In the remaining case, $|S| = n(G) - 2$. Since we have assumed that $|S| < \kappa(G)$, $|S| = n(G) - 2$ implies that $\kappa(G) \geq n(G) - 1$, which holds only for a complete graph. Thus $\kappa(G - xy) = n(G) - 2 = \kappa(G) - 1$, as desired. $\blacksquare$

4.2.21. Theorem. The connectivity of G equals the maximum k such that $\lambda(x, y) \geq k$ for all $x, y \in V(G)$. The edge-connectivity of G equals the maximum k such that $\lambda'(x, y) \geq k$ for all $x, y \in V(G)$. Both statements hold for graphs and for digraphs.

Proof: Since $\kappa'(G) = \min_{x,y \in V(G)} \kappa'(x, y)$, Theorem 4.2.19 immediately yields the claim for edge-connectivity.

For connectivity, we have $\kappa(x, y) = \lambda(x, y)$ for $xy \notin E(G)$, and $\kappa(G)$ is the minimum of these values. We need only show that $\lambda(x, y)$ cannot be smaller than $\kappa(G)$ when $xy \in E(G)$. Certainly deletion of xy reduces $\lambda(x, y)$ by 1, since xy itself is an x, y-path and cannot contribute to any other x, y-path. With this, Theorem 4.2.17, and Lemma 4.2.20, we have

$$\lambda_G(x, y) = 1 + \lambda_{G-xy}(x, y) = 1 + \kappa_{G-xy}(x, y) \geq 1 + \kappa(G - xy) \geq \kappa(G). \quad \blacksquare$$

APPLICATIONS OF MENGER'S THEOREM

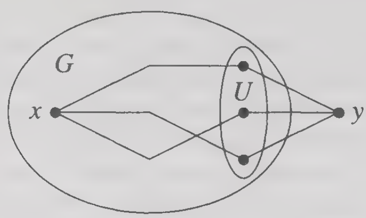
Dirac extended Menger's Theorem to other families of paths.

4.2.22. Definition. Given a vertex x and a set U of vertices, an x, U**-fan** is a set of paths from x to U such that any two of them share only the vertex x.

4.2.23. Theorem. (Fan Lemma, Dirac [1960]). A graph is k-connected if and only if it has at least $k+1$ vertices and, for every choice of x, U with $|U| \geq k$, it has an x, U-fan of size k.

Proof: *Necessity.* Given k-connected graph G, we construct G' from G by adding a new vertex y adjacent to all of U. The Expansion Lemma (Lemma 4.2.3) implies that G' also is k-connected, and then Menger's Theorem yields k pairwise internally disjoint x, y-paths in G'. Deleting y from these paths produces an x, U-fan of size k in G.

Sufficiency. Suppose that G satisfies the fan condition. For $v \in V(G)$ and $U = V(G) - \{v\}$, there is a v, U-fan of size k; thus $\delta(G) \geq k$. Given $w, z \in V(G)$, let $U = N(z)$. Since $|U| \geq k$, we have an w, U-fan of size k; extend each path by adding an edge to z. We obtain k pairwise internally disjoint w, z-paths, so $\lambda(w, z) \geq k$. This holds for all $w, z \in V(G)$, so G is k-connected. ■



The Fan Lemma generalizes considerably. Whenever X and Y are disjoint sets of vertices in a k-connected graph G and we specify integers at X and Y summing to k in each set, there are k pairwise internally disjoint X, Y-paths with the specified number ending at each point (Exercise 28). The Fan Lemma also yields the next result.

4.2.24.* Theorem. (Dirac [1960]) If G is a k-connected graph (with $k \geq 2$), and S is a set of k vertices in G, then G has a cycle including S in its vertex set.

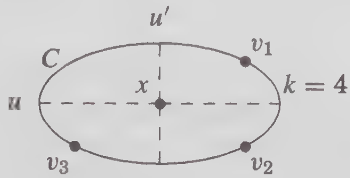
Proof: We use induction on k. Basis step ($k = 2$): Theorem 4.2.2 (or Theorem 4.2.21) implies that any two vertices are connected by two internally disjoint paths, which form a cycle containing them.

Induction step ($k > 2$): With G and S as specified, choose $x \in S$. Since G is also $k-1$-connected, the induction hypothesis implies that all of $S - \{x\}$ lies on a cycle C. Suppose first that $n(C) = k - 1$. Since G is $k-1$-connected, we have an $x, V(C)$-fan of size $k-1$, and the paths of the fan to two consecutive vertices of C enlarge the cycle to include x.

Hence we may assume that $n(C) \geq k$. Since G is k-connected, G has an

$x, V(C)$-fan of size k. We claim that again the fan has two paths forming a detour from C that includes x while keeping $S - \{x\}$. Let $v_1, \dots, v_{k-1}$ be the vertices of $S - \{x\}$ in order on C, and let V_i be the portion of $V(C)$ from v_i up to but not including v_{i+1} (here $v_k = v_1$).

The sets $V_1, \dots, V_{k-1}$ partition $V(C)$ into $k-1$ disjoint sets. Since the $x, V(C)$-fan has k paths, two of them enter $V(C)$ in one of these sets, by the pigeonhole principle. Let u, u' be the vertices where these paths reach C. Replacing the u, u'-portion of C by the x, u-path and x, u'-path in the fan builds a new cycle that contains x and all of $S - \{x\}$. ■



Many applications of Menger's Theorem involve modeling a problem so that the desired objects correspond to paths in a graph or digraph, often by graph transformation arguments. For example, given sets $\mathbf{A} = A_1, \dots, A_m$ with union X, a **system of distinct representatives** (SDR) is a set of distinct elements $x_1, \dots, x_m$ such that $x_i \in A_i$. A necessary and sufficient condition for the existence of an SDR is that $\left|\bigcup_{i\in I} A_i\right| \ge |I|$ for all $I \subseteq [m]$. It is easy to prove this from Hall's Theorem by modeling $\mathbf{A}$ with an appropriate bipartite graph (Exercise 3.1.19). Indeed, Hall's Theorem was originally proved in the language of SDRs and is equivalent to Menger's Theorem (Exercise 23).

Ford and Fulkerson considered a more difficult problem. Let $\mathbf{A} = A_1, \dots, A_m$ and $\mathbf{B} = B_1, \dots, B_m$ be two families of sets. We may ask when there is a **common system of distinct representatives (CSDR)**, meaning a set of m elements that is both an SDR for $\mathbf{A}$ and an SDR for $\mathbf{B}$. They found a necessary and sufficient condition.

4.2.25.* Theorem. (Ford–Fulkerson [1958]) Families $\mathbf{A} = \{A_1, \dots, A_m\}$ and $\mathbf{B} = \{B_1, \dots, B_m\}$ have a common system of distinct representatives (CSDR) if and only if

$$\left|\Big(\bigcup_{i\in I} A_i\Big) \cap \Big(\bigcup_{j\in J} B_j\Big)\right| \ge |I| + |J| - m \quad \text{for each pair } I, J \subseteq [m].$$

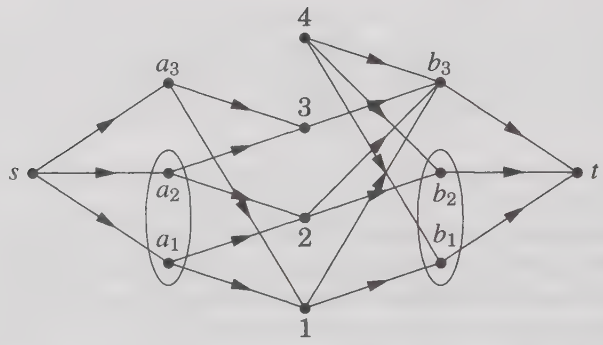
Proof: We create a digraph G with vertices $a_1, \dots, a_m$ and $b_1, \dots, b_m$, plus a vertex for each element in the sets and special vertices s, t. The edges are

$$\begin{array}{ll} \{sa_i\colon A_i \in \mathbf{A}\} & \{a_i x\colon x \in A_i\} \\ \{b_j t\colon B_j \in \mathbf{B}\} & \{x b_j\colon x \in B_j\} \end{array}$$

Each s, t-path selects a member of the intersection of some A_i and some B_j. There is a CSDR if and only if there is a set of m pairwise internally disjoint s, t-paths. By Menger's Theorem, it suffices to show that the stated condition is equivalent to having no s,t-cut of size less than m. Given a set $R \subseteq V(G) - \{s,t\}$, let $I = \{a_i\} - R$ and $J = \{b_j\} - R$. The set R is an s,t-cut if and only if $(\bigcup_{i\in I} A_i) \cap (\bigcup_{j\in J} B_j) \subseteq R$. For an s,t-cut R, we thus have

$$|R| \ge \left|(\bigcup_{i\in I} A_i) \cap (\bigcup_{j\in J} B_j)\right| + (m - |I|) + (m - |J|).$$

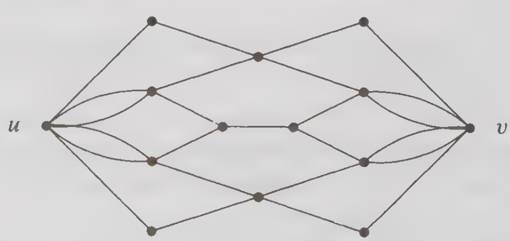
This lower bound is at least m for every s,t-cut if and only if the stated condition holds. ■



4.2.26.* Example. *Digraph for CSDR.* In the example above, the elements are $\{1,2,3,4\}$, $\mathbf{A} = \{12, 23, 31\}$, and $\mathbf{B} = \{14, 24, 1234\}$. Suppose that $R \cap \{a_i\} = \{a_1, a_2\}$ and $R \cap \{b_j\} = \{b_1, b_2\}$. In the argument, we set $I = \{a_3\}$ and $J = \{b_3\}$, and we observe that R is an s,t-cut if and only if it also contains $\{1,3\}$, which equals $(\bigcup_{i\in I} A_i) \cap (\bigcup_{j\in J} B_j)$. ■

EXERCISES

4.2.1. $(-)$ Determine $\kappa(u,v)$ and $\kappa'(u,v)$ in the graph drawn below. (Hint: Use the dual problems to give short proofs of optimality.)



4.2.2. $(-)$ Prove that if G is 2-edge-connected and G' is obtained from G by subdividing an edge of G, then G' is 2-edge-connected. Use this to prove that every graph having a closed-ear decomposition is 2-edge-connected. (Comment: This is an alternative proof of sufficiency for Theorem 4.2.10.)

4.2.3. (−) Let G be the digraph with vertex set $[12]$ in which $i \to j$ if and only if i divides j. Determine $\kappa(1, 12)$ and $\kappa'(1, 12)$.

4.2.4. (−) Prove or disprove: If P is a u, v-path in a 2-connected graph G, then there is a u, v-path Q that is internally disjoint from P.

4.2.5. (−) Let G be a simple graph, and let $H(G)$ be the graph with vertex set $V(G)$ such that $uv \in E(H)$ if and only if u, v appear on a common cycle in G. Characterize the graphs G such that H is a clique.

4.2.6. (−) Use results of this section to prove that a simple graph G is 2-connected if and only if G can be obtained from C_3 by a sequence of edge additions and edge subdivisions.

• • • • •

4.2.7. Let xy be an edge in a digraph G. Prove that $\kappa(G - xy) \ge \kappa(G) - 1$.

4.2.8. Prove that a simple graph G is 2-connected if and only if for every triple (x, y, z) of distinct vertices, G has an x, z-path through y. (Chein [1968])

4.2.9. Prove that a graph G with at least four vertices is 2-connected if and only if for every pair X, Y of disjoint vertex subsets with $|X|, |Y| \ge 2$, there exist two completely disjoint paths P_1, P_2 in G such that each has an endpoint in X and an endpoint in Y and no internal vertex in X or Y.

4.2.10. A **greedy ear decomposition** of a 2-connected graph is an ear decomposition that begins with a longest cycle and iteratively adds a longest ear from the remaining graph. Use a greedy ear decomposition to prove that every 2-connected claw-free graph G has $\lfloor n(G)/3 \rfloor$ pairwise-disjoint copies of P_3. (Kaneko–Kelmans–Nishimura [2000])

4.2.11. (!) For a connected graph G with at least three vertices, prove that the following statements are equivalent (use of Menger's Theorem is permitted).

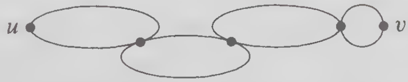
A) G is 2-edge-connected.
B) Every edge of G appears in a cycle.
C) G has a closed trail containing any specified pair of edges.
D) G has a closed trail containing any specified pair of vertices.

4.2.12. (!) Use Menger's Theorem to prove that $\kappa(G) = \kappa'(G)$ when G is 3-regular (Theorem 4.1.11).

4.2.13. (!) Let G be a 2-edge-connected graph. Define a relation R on $E(G)$ by $(e, f) \in R$ if $e = f$ or if $G - e - f$ is disconnected. (Lovász [1979, p277])

a) Prove that $(e, f) \in R$ if and only if e, f belong to the same cycles.
b) Prove that R is an equivalence relation on $E(G)$.
c) For each equivalence class F, prove that F is contained in a cycle.
d) For each equivalence class F, prove that $G - F$ has no cut-edge.

4.2.14. (!) A u, v-**necklace** is a list of cycles $C_1, \ldots, C_k$ such that $u \in C_1$, $v \in C_k$, consecutive cycles share one vertex, and nonconsecutive cycles are disjoint. Use induction on $d(u, v)$ to prove that a graph G is 2-edge-connected if and only if for all $u, v \in V(G)$ there is a u, v-necklace in G.



4.2.15. (+) Let v be a vertex of a 2-connected graph G. Prove that v has a neighbor u such that $G - u - v$ is connected. (Chartrand–Lesniak [1986, p51])

4.2.16. (+) Let G be a 2-connected graph. Prove that if T_1, T_2 are two spanning trees of G, then T_1 can be transformed into T_2 by a sequence of operations in which a leaf is removed and reattached using another edge of G.

4.2.17. Determine the smallest graph with connectivity 3 having a pair of nonadjacent vertices linked by four pairwise internally disjoint paths.

4.2.18. Let G be a graph without isolated vertices. Prove that if G has no even cycles, then every block of G is an edge or an odd cycle.

4.2.19. (!) *Membership in common cycles.*

a) Prove that two distinct edges lie in the same block of a graph if and only if they belong to a common cycle.

b) Given $e, f, g \in E(G)$, suppose that G has a cycle through e and f and a cycle through f and g. Prove that G also has a cycle through e and g. (Comment: This problem implies that for graphs without cut-edges, "belong to a common cycle" is an equivalence relation whose equivalence classes are the edge sets of blocks.)

4.2.20. Prove that the hypercube Q_k is k-connected by constructing k pairwise internally disjoint x, y-paths for each vertex pair $x, y \in V(Q_k)$.

4.2.21. (!) Let G be a $2k$-edge-connected graph with at most two vertices of odd degree. Prove that G has a k-edge-connected orientation. (Nash-Williams [1960])

4.2.22. (!) Suppose that $\kappa(G) = k$ and diam $G = d$. Prove that $n(G) \geq k(d-1)+2$ and $\alpha(G) \geq \lceil (1+d)/2 \rceil$. For each $k \geq 1$ and $d \geq 2$, construct a graph for which equality holds in both bounds.

4.2.23. (!) Use Menger's Theorem ($\kappa(x, y) = \lambda(x, y)$ when $xy \notin E(G)$) to prove the König–Egerváry Theorem ($\alpha'(G) = \beta(G)$ when G is bipartite).

4.2.24. (!) Let G be a k-connected graph, and let S, T be disjoint subsets of $V(G)$ with size at least k. Prove that G has k pairwise disjoint S, T-paths.

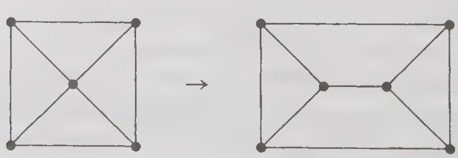
4.2.25. (*) Show that Theorem 4.2.24 is best possible by constructing for each k a k-connected graph having $k+1$ vertices that do not lie on a cycle.

4.2.26. For $k \geq 2$, prove that a graph with at least $k+1$ vertices is k-connected if and only if for every $T \subseteq S \subseteq V(G)$ with $|S| = k$ and $|T| = 2$, there is a cycle in G that contains T and avoids $S - T$. (Lick [1973])

4.2.27. A **vertex k-split** of a graph G is a graph H obtained from G by replacing one vertex $x \in V(G)$ by two adjacent vertices x_1, x_2 such that $d_H(x_i) \geq k$ and that $N_H(x_1) \cup N_H(x_2) = N_G(x) \cup \{x_1, x_2\}$.

a) Prove that every vertex k-split of a k-connected graph is k-connected.

b) Conclude that any graph obtained from a "wheel" $W_n = K_1 \vee C_{n-1}$ (Definition 3.3.6) by a sequence of edge additions and vertex 3-splits on vertices of degree at least 4 is 3-connected. (Comment: Tutte [1961b] proved also that every 3-connected graph arises in this way. The characterization does not extend easily for $k > 3$.)



4.2.28. (!) Let X and Y be disjoint sets of vertices in a k-connected graph G. Let $u(x)$ for $x \in X$ and $w(y)$ for $y \in Y$ be nonnegative integers such that $\sum_{x \in X} u(x) = \sum_{y \in Y} w(y) = k$. Prove that G has k pairwise internally disjoint X, Y-paths so that $u(x)$ of them start at x and $w(y)$ of them end at y, for $x \in X$ and $y \in Y$.

4.2.29. Given a graph G, let D be the digraph obtained by replacing each edge with two oppositely-directed edges having the same endpoints (thus D is the symmetric digraph with underlying graph G). Assume that for all $x, y \in V(D)$ both $\kappa'_D(x, y) = \lambda'_D(x, y)$ and $\kappa_D(x, y) = \lambda_D(x, y)$ hold, the latter applying only when $x \not\to y$. Use this hypothesis to prove that also $\kappa'_G(x, y) = \lambda'_G(x, y)$ and $\kappa_G(x, y) = \lambda_G(x, y)$, the latter for $x \not\leftrightarrow y$.

4.2.30. (!) Prove that applying the expansion operation of Example 1.3.26 to a 3-connected graph yields a 3-connected graph. Obtain the Petersen graph from K_4 by expansions. (Comment: Tutte [1966a] proved that a 3-regular graph is 3-connected if and only if it arises from K_4 by a sequence of these operations.)

4.2.31. Let G be a k-connected simple graph.

a) Let C and D be two cycles in G of maximum length. For $k = 2$ and $k = 3$, prove that C and D share at least k vertices. (Hint: If they don't, construct a longer cycle.)

b) For each $k \geq 2$, construct a k-connected graph that has distinct longest cycles with only k common vertices. (Hint: $K_{2,4}$ works for $k = 2$.)

4.2.32. *Graph splices.* Let G_1 and G_2 be disjoint k-connected graphs with $k \geq 2$. Choose $v_1 \in V(G_1)$ and $v_2 \in V(G_2)$. Let B be a bipartite graph with partite sets $N_{G_1}(v_1)$ and $N_{G_2}(v_2)$ that has no isolated vertex and has a matching of size at least k. Prove that $(G_1 - v_1) \cup (G_2 - v_2) \cup B$ is k-connected.

4.2.33. $(*)$ Prove Hall's Theorem from Theorem 4.2.25.

4.2.34. A k-connected graph G is **minimally k-connected** if for every $e \in E(G)$, the graph $G - e$ is not k-connected. Halin [1969] proved that $\delta(G) = k$ when G is minimally k-connected. Use ear decomposition to prove this for $k = 2$. Conclude that a minimally 2-connected graph G with at least 4 vertices has at most $2n(G) - 4$ edges, with equality only for $K_{2,n-2}$. (Dirac [1967])

4.2.35. Prove that if G is 2-connected, then $G - xy$ is 2-connected if and only if x and y lie on a cycle in $G - xy$. Conclude that a 2-connected graph is minimally 2-connected if and only if every cycle is an induced subgraph. (Dirac [1967], Plummer [1968])

4.2.36. (!) For $S \subseteq V(G)$, let $d(S) = \left|[S, \overline{S}]\right|$. Let X and Y be nonempty proper vertex subsets of G. Prove that $d(X \cap Y) + d(X \cup Y) \leq d(X) + d(Y)$. (Hint: Draw a picture and consider contributions from various types of edges.)

4.2.37. $(+)$ A k-edge-connected graph G is **minimally k-edge-connected** if for every $e \in E(G)$ the graph $G - e$ is not k-edge-connected. Prove that $\delta(G) = k$ when G is minimally k-edge-connected. (Hint: Consider a minimal set S such that $\left|[S, \overline{S}]\right| = k$. If $|S| \neq 1$, use $G - e$ for some $e \in E(G[S])$ to obtain another set T with $\left|[T, \overline{T}]\right| = k$ such that S, T contradict Exercise 4.2.36.) (Mader [1971]; see also Lovász [1979, p285])

4.2.38. Mader [1978] proved the following: "If z is a vertex of a graph G such that $d_G(z) \notin \{0, 1, 3\}$ and z is incident to no cut-edge, then z has neighbors x and y such that $\kappa_{G-xz-yz+xy}(u, v) = \kappa_G(u, v)$ for all $u, v \in V(G) - \{z\}$." Use Mader's Theorem and Exercise 4.2.37 to prove Nash-Williams' Orientation Theorem: every $2k$-edge-connected graph has a k-edge-connected orientation. (Comment: A weaker version of Mader's Theorem, given in Lovász [1979, p286–288], also yields Nash-Williams' Theorem in the same way.)

4.3. Network Flow Problems

Consider a network of pipes where valves allow flow in only one direction. Each pipe has a capacity per unit time. We model this with a vertex for each junction and a (directed) edge for each pipe, weighted by the capacity. We also assume that flow cannot accumulate at a junction. Given two locations s, t in the network, we may ask "what is the maximum flow (per unit time) from s to t?"

This question arises in many contexts. The network may represent roads with traffic capacities, or links in a computer network with data transmission capacities, or currents in an electrical network. There are applications in industrial settings and to combinatorial min-max theorems. The seminal book on the subject is Ford–Fulkerson [1962]. More recently, Ahuja–Magnanti–Orlin [1993] presents a thorough treatment of network flow problems.

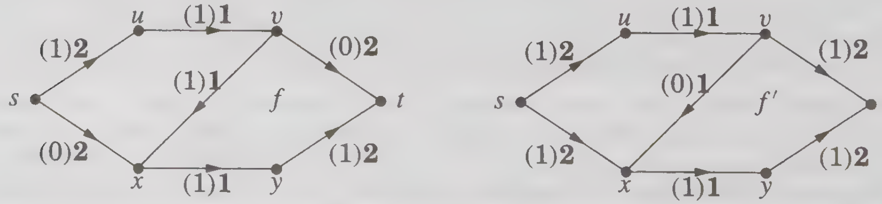
4.3.1. Definition. A **network** is a digraph with a nonnegative **capacity** $c(e)$ on each edge e and a distinguished **source vertex** s and **sink vertex** t. Vertices are also called **nodes**. A **flow** f assigns a value $f(e)$ to each edge e. We write $f^+(v)$ for the total flow on edges leaving v and $f^-(v)$ for the total flow on edges entering v. A flow is **feasible** if it satisfies the **capacity constraints** $0 \le f(e) \le c(e)$ for each edge and the **conservation constraints** $f^+(v) = f^-(v)$ for each node $v \notin \{s, t\}$.

MAXIMUM NETWORK FLOW

We consider first the problem of maximizing the net flow into the sink.

4.3.2. Definition. The **value** $\text{val}(f)$ of a flow f is the net flow $f^-(t) - f^+(t)$ into the sink. A **maximum flow** is a feasible flow of maximum value.

4.3.3. Example. The **zero flow** assigns flow 0 to each edge; this is feasible. In the network below we illustrate a nonzero feasible flow. Each capacities are shown in bold, flow values in parentheses. Our flow f assigns $f(sx) = f(vt) = 0$, and $f(e) = 1$ for every other edge e. This is a feasible flow of value 1.



A path from the source to the sink with excess capacity would allow us to increase flow. In this example, no path remains with excess capacity, but the flow f' with $f'(vx) = 0$ and $f'(e) = 1$ for $e \neq vx$ has value 2. The flow f is "maximal" in that no other feasible flow can be found by increasing the flow on some edges, but f is not a maximum flow.

We need a more general way to increase flow. In addition to traveling forward along edges with excess capacity, we allow traveling backward (against the arrow) along edges where the flow is nonzero. In this example, we can travel from s to x to v to t. Increasing the flow by 1 on sx and vt and decreasing it by one on vx changes f into f'. ■

4.3.4. Definition. When f is a feasible flow in a network N, an f**-augmenting path** is a source-to-sink path P in the underlying graph G such that for each $e \in E(P)$,

a) if P follows e in the forward direction, then $f(e) < c(e)$.

b) if P follows e in the backward direction, then $f(e) > 0$.

Let $\epsilon(e) = c(e) - f(e)$ when e is forward on P, and let $\epsilon(e) = f(e)$ when e is backward on P. The **tolerance** of P is $\min_{e \in E(P)} \epsilon(e)$.

As in Example 4.3.3, an f-augmenting path leads to a flow with larger value. The definition of f-augmenting path ensures that the tolerance is positive; this amount is the increase in the flow value.

4.3.5. Lemma. If P is an f-augmenting path with tolerance z, then changing flow by $+z$ on edges followed forward by P and by $-z$ on edges followed backward by P produces a feasible flow f' with $\mathrm{val}(f') = \mathrm{val}(f) + z$.

Proof: The definition of tolerance ensures that $0 \le f'(e) \le c(e)$ for every edge e, so the capacity constraints hold. For the conservation constraints we need only check vertices of P, since flow elsewhere has not changed.

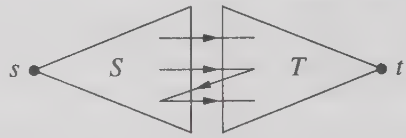
The edges of P incident to an internal vertex v of P occur in one of the four ways shown below. In each case, the change to the flow out of v is the same as the change to the flow into v, so the net flow out of v remains 0 in f'.

Finally, the net flow into the sink t increases by z. ■

The flow on backward edges did not disappear; it was redirected. In effect, the augmentation in Example 4.3.3 cuts the flow path and extends each portion to become a new flow path. We will soon describe an algorithm to find augmenting paths.

Meanwhile, we would like a quick way to know when our present flow is a maximum flow. In Example 4.3.3, the central edges seem to form a "bottleneck"; we only have capacity 2 from the left half of the network to the right half. This observation will give us a PROOF that the flow value can be no larger.

4.3.6. Definition. In a network, a **source/sink cut** $[S, T]$ consists of the edges from a **source set** S to a **sink set** T, where S and T partition the set of nodes, with $s \in S$ and $t \in T$. The **capacity** of the cut $[S, T]$, written $\mathrm{cap}(S, T)$, is the total of the capacities on the edges of $[S, T]$.



Keep in mind that in a digraph $[S, T]$ denotes the set of edges with tail in S and head in T. Thus the capacity of a cut $[S, T]$ is completely unaffected by edges from T to S.

Given a cut $[S, T]$, every s, t-path uses at least one edge of $[S, T]$, so intuition suggests that the value of a feasible flow should be bounded by $\mathrm{cap}(S, T)$. To make this precise, we extend the notion of net flow to sets of nodes. Let $f^+(U)$ denote the total flow on edges leaving U, and let $f^-(U)$ be the total flow on edges entering U. The net flow out of U is then $f^+(U) - f^-(U)$.

4.3.7. Lemma. If U is a set of nodes in a network, then the net flow out of U is the sum of the net flows out of the nodes in U. In particular, if f is a feasible flow and $[S, T]$ is a source/sink cut, then the net flow out of S and net flow into T equal $\mathrm{val}(f)$.

Proof: The stated claim is that

$$f^+(U) - f^-(U) = \textstyle\sum_{v \in U}[f^+(v) - f^-(v)].$$

We consider the contribution of the flow $f(xy)$ on an edge xy to each side of the formula. If $x, y \in U$, then $f(xy)$ is not counted on the left, but it contributes positively (via $f^+(x)$) and negatively (via $f^-(y)$) on the right. If $x, y \notin U$, then $f(xy)$ contributes to neither sum. If $xy \in [U, \overline{U}]$, then it contributes positively to each sum. If $xy \in [\overline{U}, U]$, then it contributes negatively to each sum. Summing over all edges yields the equality.

When $[S, T]$ is a source/sink cut and f is a feasible flow, net flow from nodes of S sums to $f^+(s) - f^-(s)$, and net flow from nodes of T sums to $f^+(t) - f^-(t)$, which equals $-\mathrm{val}(f)$. Hence the net flow across any source/sink cut equals both the net flow out of s and the net flow into t. ■

4.3.8. Corollary. (Weak duality) If f is a feasible flow and $[S, T]$ is a source/sink cut, then $\mathrm{val}(f) \le \mathrm{cap}(S, T)$.

Proof: By the lemma, the value of f equals the net flow out of S. Thus

$$\mathrm{val}(f) = f^+(S) - f^-(S) \le f^+(S),$$

since the flow into S is no less than 0. Since the capacity constraints require $f^+(S) \le \mathrm{cap}(S, T)$, we obtain $\mathrm{val}(f) \le \mathrm{cap}(S, T)$. ■

Among source/sink cuts, one with minimum capacity yields the best bound on the value of a flow. This defines the **minimum cut** problem. The max flow and min cut problems on a network are dual optimization problems.† Given a flow with value α and a cut with value α, the duality inequality in Corollary 4.3.8 PROVES that the cut is a minimum cut and the flow is a maximum flow.

If every instance has solutions with the same value to both the max problem and the min problem ("strong duality"), then a short proof of optimality always exists. This does not hold for all dual pairs of problems (recall matching and covering in general graphs), but it holds for max flow and min cut.

The Ford–Fulkerson algorithm seeks an augmenting path to increase the flow value. If it does not find such a path, then it finds a cut with the same value (capacity) as this flow; by Corollary 4.3.8, both are optimal. If no infinite sequence of augmentations is possible, then the iteration leads to equality between the maximum flow value and the minimum cut capacity.

4.3.9. Algorithm. (Ford–Fulkerson labeling algorithm)
Input: A feasible flow f in a network.
Output: An f-augmenting path or a cut with capacity val(f).
Idea: Find the nodes reachable from s by paths with positive tolerance. Reaching t completes an f-augmenting path. During the search, R is the set of nodes labeled *Reached*, and S is the subset of R labeled *Searched*.
Initialization: $R = \{s\}$, $S = \emptyset$.
Iteration: Choose $v \in R - S$.

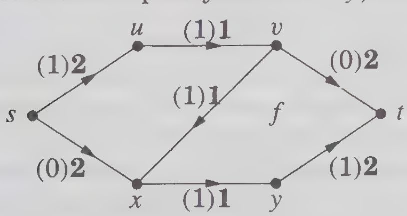
For each exiting edge vw with $f(vw) < c(vw)$ and $w \notin R$, add w to R.

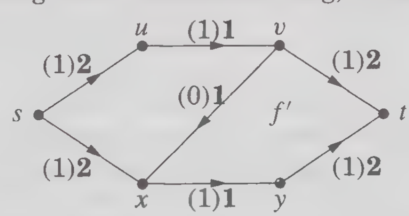
For each entering edge uv with $f(uv > 0)$ and $u \notin R$, add u to R.

Label each vertex added to R as "reached", and record v as the vertex reaching it. After exploring all edges at v, add v to S.

If the sink t has been reached (put in R), then trace the path reaching t to report an f-augmenting path and terminate. If $R = S$, then return the cut $[S, \overline{S}]$ and terminate. Otherwise, iterate. ■

4.3.10. Example. On the left below is the network of Example 4.3.3 with the flow f. We run the labeling algorithm. First we search from s and find excess capacity to u and x, labeling them reached. Now we have $u, v \in R - S$. There is no excess capacity on uv or xy, so searching from u reaches nothing, and also





†The precise notion of "dual problem" comes from linear programming. For our purposes, dual problems are a maximization problem and a minimization problem such that $a \leq b$ whenever a and b are the values of feasible solutions to the max problem and min problem, respectively. See Section 8.1 for further discussion.

searching from x does not reach y. However, there is nonzero flow on vx. Thus we label v from x. Now v is the only element of $R - S$, and searching from v reaches t. We labeled t from v, v from x, and x from s, so we have found the augmenting path s, x, v, t.

The tolerance on this path is 1, so the augmentation increases the flow value by 1. In the new flow f' shown on the right, every edge has unit flow except $f'(vx) = 0$. When we run the labeling algorithm again, we have excess capacity on su and sx and can label $\{u, x\}$, but from these nodes we can label no others. We terminate with $R = S = \{s, u, x\}$. The capacity of the resulting cut $[S, \overline{S}]$ is 2, which equals $\mathrm{val}(f')$ and proves that f' is a maximum flow. ■

Repeated use of the labeling algorithm allows us to solve the maximum flow problem and prove the strong duality relationship.

4.3.11. Theorem. (Max-flow Min-cut Theorem—Ford and Fulkerson [1956]) In every network, the maximum value of a feasible flow equals the minimum capacity of a source/sink cut.

Proof: In the max-flow problem, the zero flow ($f(e) = 0$ for all e) is always a feasible flow and gives us a place to start. Given a feasible flow, we apply the labeling algorithm. It iteratively adds vertices to S (each vertex at most once) and terminates with $t \in R$ ("breakthrough") or with $S = R$.

In the breakthrough case, we have an f-augmenting path and increase the flow value. We then repeat the labeling algorithm. When the capacities are rational, each augmentation increases the flow by a multiple of $1/a$, where a is the least common multiple of the denominators, so after finitely many augmentations the capacity of some cut is reached. The labeling algorithm then terminates with $S = R$.

When terminating this way, we claim that $[S, T]$ is a source/sink cut with capacity $\mathrm{val}(f)$, where $T = \overline{S}$ and f is the present flow. It is a cut because $s \in S$ and $t \notin R = S$. Since applying the labeling algorithm to the flow f introduces no node of T into R, no edge from S to T has excess capacity, and no edge from T to S has nonzero flow in f. Hence $f^+(S) = \mathrm{cap}(S, T)$ and $f^-(S) = 0$.

Since the net flow out of any set containing the source but not the sink is $\mathrm{val}(f)$, we have proved

$$\mathrm{val}(f) = f^+(S) - f^-(S) = f^+(S) = \mathrm{cap}(S, T). \quad ■$$

This proof of Theorem 4.3.11 requires rational capacities; otherwise, Algorithm 4.3.9 may yield augmenting paths forever! Ford and Fulkerson provided an example of this with only ten vertices (see Papadimitriou–Steiglitz [1982, p126-128]). Edmonds and Karp [1972] modified the labeling algorithm to use at most $(n^3 - n)/4$ augmentations in an n-vertex network and work for all real capacities. As in the bipartite matching problem (Theorem 3.2.22), this is done by searching always for shortest augmenting paths. Faster algorithms are now known; again we cite Ahuja–Magnanti–Orlin [1993] for a thorough discussion.

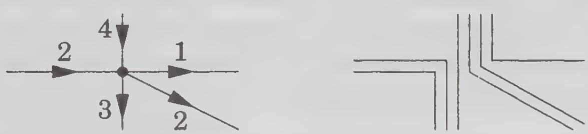
INTEGRAL FLOWS

In combinatorial applications, we typically have integer capacities and want a solution in which the flow on each edge is an integer.

4.3.12. Corollary. (Integrality Theorem) If all capacities in a network are integers, then there is a maximum flow assigning integral flow to each edge. Furthermore, some maximum flow can be partitioned into flows of unit value along paths from source to sink.

Proof: In the labeling algorithm of Ford and Fulkerson, the change in flow value when an augmenting path is found is always a flow value or the difference between a flow value and a capacity. When these are integers, the difference is also an integer. Starting with the zero flow, this implies that there is no first time when a noninteger flow appears.

The algorithm thus produces a maximum flow with integer flow on each edge. At each internal node, we now match units of entering flow to units of exiting flow. This forms s,t-paths and perhaps cycles. If a cycle arises, then we decrease flow on its edges by 1 to eliminate it without changing the flow value. This leaves $\mathrm{val}(f)$ paths from s to t, each corresponding to a unit of flow. ■



The integrality theorem yields paths of unit flow. In applications, we build networks where these units of flow have meaning.

The next two remarks show that the Max-flow Min-cut Theorem for networks with integer capacities is almost the same statement as Menger's Theorem for edge-disjoint paths in digraphs.

4.3.13. Remark. *Menger from Max-flow Min-cut.* When x, y are vertices in a digraph D, we can view D as a network with source x and sink y and capacity 1 on every edge. Capacity 1 ensures that units of flow from x to y correspond to pairwise edge-disjoint x, y-paths in D. Thus a flow of value k yields a set of k such paths.

Similarly, every source/sink partition S, T defines a set of edges whose deletion makes y unreachable from x: the set $[S, T]$. Since every capacity is 1, the size of this set is $\mathrm{cap}(S, T)$.

The paths and the edge cut we have obtained might not be optimal, but by the Max-flow Min-cut Theorem we have

$$\lambda'_D(x, y) \geq \max \mathrm{val}(f) = \min \mathrm{cap}(S, T) \geq \kappa'_D(x, y).$$

Since always $\kappa'(x, y) \geq \lambda'(x, y)$, equality now holds. ■

4.3.14. Remark. *Max-flow Min-cut from Menger.* To show that Menger's Theorem implies the Max-flow Min-cut Theorem for rational capacities, we take an arbitrary network and transform it into a digraph where we apply Menger's Theorem. By multiplying all capacities by the least common denominator, we may assume that the capacities are integers.

Given a network N with integer capacities, we form a digraph D by splitting each edge of capacity j into j edges with the same endpoints. For N, duality yields $\max \mathrm{val}(f) \le \min \mathrm{cap}(S, T)$. This time we want to use Menger's Theorem on D to obtain the reverse inquality, so in contrast to Remark 4.3.13 our desired computation is

$$\max \mathrm{val}(f) \ge \lambda'_D(s,t) = \kappa'_D(s,t) \ge \min \mathrm{cap}(S, T).$$

A set of $\lambda'(s,t)$ pairwise edge-disjoint s,t-paths in D collapses into a flow of value $\lambda'(s,t)$ in N, since the number of copies of each edge in D equals the capacity of the edge in N. Thus $\max \mathrm{val}(f) \ge \lambda'(s,t)$.

Now, let F be a set of $\kappa'(s,t)$ edges disconnecting t from s in D. If $e \in F$, then the minimality of F implies that $D - (F - e)$ has an s,t-path P through e. If some other copy e' of the edge $e = uv$ is not in F, then P can be rerouted along e' to obtain an s,t-path in $D - F$. Therefore, F contains all copies or no copies of each multiple edge in D. Hence $\kappa'(s,t)$ is the sum of the capacities on a set of edges that disconnects t from s in N. Letting S be the set of vertices reachable from s in $D - F$, we have $\mathrm{cap}(S, T) = \kappa'(s,t)$. The minimum cut has at most this capacity, so $\min \mathrm{cap}(S, T) \le \kappa'(s,t)$, and we have proved all the needed inequalities. ■

For combinatorial applications, Menger's Theorem may yield simpler proofs than the Max-flow Min-cut Theorem (compare Theorem 4.2.25 with ??). Nevertheless, our proof of Menger's Theorem in Section 4.2 is awkward to implement algorithmically. For large-scale computations, network flow and the Ford–Fulkerson labeling algorithm are more appropriate. Indeed, most algorithms that compute connectivity in graphs and digraphs use network flow methods (Stoer–Wagner [1994] presents a different approach).

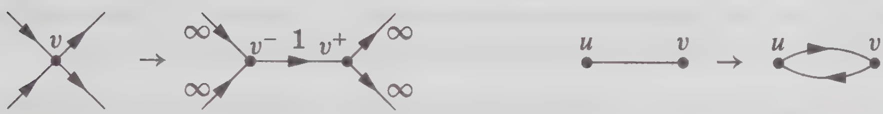
We present other network models for combinatorial problems. For example, the other local versions of Menger's Theorem can also be obtained directly.

4.3.15. Remark. *Other transformations.* For each version of Menger's Theorem, we encode the path problem using network flows with integer capacities.

To obtain a network model for the problem of internally disjoint paths in a digraph D, we must prevent two units of flow from passing through a vertex. This can be done by replacing each vertex v with two vertices v^-, v^+ that inherit the entering and exiting edges at v. By adding an edge of unit capacity from v^- to v^+, we obtain the effect of limiting flow through v to one unit. By putting very large capacity (essentially infinite) on the edges that were in D, we ensure that a minimum cut will count only edges of the form v^-v^+.

To obtain a network model for the problem of edge-disjoint paths in a graph G, we must permit flow to pass either way in an edge. This can be done by replacing each edge uv with two directed edges uv and vu. When the network sends unit flow in both directions, in effect the edge is not being used at all.

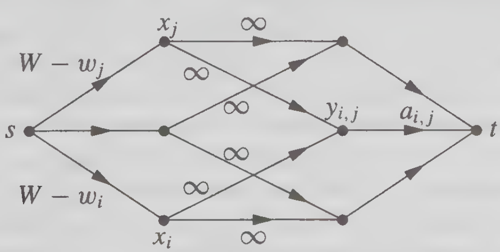
In each case, a flow in the network provides a set of paths, and a minimum cut leads to a separating set of vertices or edges. As in Remark 4.3.13, duality then gives us the desired equality in Menger's Theorem. To model the problem of internally disjoint paths in a graph, we need both of these transformations. Exercises 5–7 request the details of these proofs. ■



4.3.16. Application. *Baseball Elimination Problem* (Schwartz [1966]). At some time during the season, we may wonder whether team X can still win the championship. In other words, can winners be assigned for the remaining games so that no team ends with more victories than X? If so, then such an assignment exists with X winning all its remaining games, reaching W wins. We want to know whether winners can be chosen for other games so that no team obtains more than W wins. To test this, we create a network where units of flow correspond to the remaining games.

Let $X_1, \dots, X_n$ be the other teams. Include nodes $x_1, \dots, x_n$ for the n teams, nodes $y_{i,j}$ for the $\binom{n}{2}$ pairs of teams, and a source s and sink t. Put an edge from s to each team node and an edge from each pair node to t. Each pair node $y_{i,j}$ is entered by edges from x_i and x_j.

The capacities model the constraints. The capacity on edge $y_{i,j}t$ is $a_{i,j}$, the number of remaining games between X_i and X_j. Given that X_i has won w_i games already, the capacity on edge sx_i is $W - w_i$ to keep X in contention. The capacity on edges $x_i y_{i,j}$ and $x_j y_{i,j}$ is ∞ (the number of games x_i can win from x_j is constrained by the capacity on $y_{i,j}t$).



By the integrality theorem, a maximum flow breaks into flow units. Each unit corresponds to one game; the first edge specifies the winner, and the last edge specifies the pair. The network has a flow of value $\sum_{i,j} a_{i,j}$ if and only if *all* remaining games can be played with no team exceeding W wins; this is the condition for X remaining in contention.

By the Max-flow Min-cut Theorem, there is a flow of value $\sum a_{i,j}$ if and only if every cut has capacity at least $\sum a_{i,j}$. Let S, T be a cut with finite capacity, and let $Z = \{i: x_i \in T\}$. Since $c(x_i y_{i,j}) = \infty$, we cannot have $x_i \in S$ and $y_{i,j} \in T$; thus $y_{i,j} \in S$ whenever i or j is not in Z. To minimize capacity, we put $y_{i,j} \in T$ whenever $\{i, j\} \subseteq Z$. Now $\text{cap}(S, T) = \sum_{i \in Z}(W - w_i) + \sum_{\{i,j\} \not\subseteq Z} a_{i,j}$. The condition that every cut have capacity at least $\sum a_{i,j}$ becomes

$$\sum_{i \in Z}(W - w_i) \geq \sum_{\{i,j\} \subseteq Z} a_{i,j} \qquad \text{for all } Z \subseteq [n].$$

Note that this condition is obviously necessary; it states that we need enough leeway in the total wins among teams indexed by Z in order to accommodate winners for all the games among these teams. We have proved TONCAS. ■

Combinatorial applications of network flow usually involve showing that the desired configuration exists if and only if a related network has a large enough flow. As in Application 4.3.16, the Max-flow Min-cut Theorem then yields a necessary and sufficient condition for its existence. Other examples include most of Exercises 5– and also Exercise 13 and Theorems 4.3.17–4.3.18.

SUPPLIES AND DEMANDS (optional)

Next we consider a more general network model. We allow multiple sources and sinks, and also we associate with each source x_i a **supply** $\sigma(x_i)$ and with each sink y_j a **demand** $\partial(y_j)$. To the capacity constraints for edges and conservation constraints for internal nodes, we add **transportation constraints** for the sources and sinks.

$$f^+(x_i) - f^-(x_i) \leq \sigma(x_i) \text{ for each source } x_i$$
$$f^-(y_j) - f^+(y_j) \geq \partial(y_j) \text{ for each sink } y_j$$

The resulting configuration is a **transportation network**. With positive values for the demands, the zero flow is not feasible. We seek a feasible flow satisfying these additional constraints. The "supply/demand" terminology suggests the constraints; we must satisfy the demands at the sinks without exceeding the available supply at any source. This model is appropriate when a company has multiple distribution centers (sources) and retail outlets (sinks).

Let X and Y denote the sets of sources and sinks, respectively. Let $\sigma(A) = \sum_{v \in A} \sigma(v)$ and $\partial(B) = \sum_{v \in B} \partial(v)$ denote the total supply or demand at a set $A \subseteq X$ or $B \subseteq Y$. For a set F of edges, let $c(F) = \sum_{e \in F} c(e)$. Given a set T of vertices, the **net demand** $\partial(Y \cap T) - \sigma(X \cap T)$ must be satisfied by flow from the remaining vertices. Hence it is necessary that $c([\overline{T}, T])$ be at least this large. Satisfying this for every set T is also sufficient for a feasible flow (TONCAS).

4.3.17. Theorem. (Gale [1957]) In a transportation network N with sources X and sinks Y, a feasible flow exists if and only if

$$c([S, T]) \geq \partial(Y \cap T) - \sigma(X \cap T)$$

for every partition of the vertices of N into sets S and T.

Proof: We have already observed the necessity of the condition. For sufficiency, construct a new network N' by adding a supersource s and a supersink t, with an edge of capacity $\sigma(x_i)$ from s to each $x_i \in X$ and an edge of capacity $\partial(y_j)$ from each $y_j \in Y$ to t. The transportation network N has a feasible flow if and only if N' has a flow saturating each edge to t (a flow of value $\partial(Y)$).

By the Ford–Fulkerson Theorem, we know that N' has a flow of value $\partial(Y)$ if and only if $\mathrm{cap}(S \cup s, T \cup t) \geq \partial(Y)$ for each partition S, T of $V(N)$. The cut $[S \cup s, T \cup t]$ in N' consists of $[S, T]$ from N, plus edges from s to T and edges from S to t in N'. Hence

$$\mathrm{cap}(S \cup s, T \cup t) = c(S, T) + \sigma(T \cap X) + \partial(S \cap Y).$$

We now have $\mathrm{cap}(S \cup s, T \cup t) \geq \partial(Y)$ if and only if

$$c(S, T) + \sigma(X \cap T) \geq \partial(Y) - \partial(Y \cap S) = \partial(Y \cap T),$$

which is the condition assumed. ■

For specific instances, the construction of N' is the key point, because we produce a feasible flow in N (when it exists) by running the Ford–Fulkerson algorithm on the network N'. When costs (per unit flow) are attached to the edges, we have the Min-cost Flow Problem, which generalizes the Transportation Problem of Application 3.2.14. Solution algorithms for the Min-cost Flow Problem appear in Ford–Fulkerson [1962] and in Ahuja–Magnanti–Orlin [1993].

We discuss several applications of Gale's condition. A pair of integer lists $p = (p_1, \dots, p_m)$ and $q = (q_1, \dots, q_n)$ is **bigraphic** (Exercise 1.4.31) if there is a simple X, Y-bigraph such that the vertices of X have degrees $p_1, \dots, p_m$ and the vertices of Y have degrees $q_1, \dots, q_n$. Clearly $\sum p_i = \sum q_j$ is necessary, but this condition is not sufficient. To test whether (p, q) is bigraphic, we create a network in which units of flows will correspond to edges in the desired graph. The result is a bipartite analogue of the Erdős–Gallai condition for graphic sequences (Exercise 3.3.28).

4.3.18. Theorem. (Gale [1957], Ryser [1957]) If p, q are lists of nonnegative integers with $p_1 \geq \cdots \geq p_m$ and $q_1 \geq \cdots \geq q_n$, then (p, q) is bigraphic if and only if $\sum_{i=1}^{m} \min\{p_i, k\} \geq \sum_{j=1}^{k} q_j$ for $1 \leq k \leq n$.

Proof: *Necessity*. Let G be a simple X, Y-bigraph realizing (p, q). Consider the edges incident to a set of k vertices in Y. Because G is simple, each $x_i \in X$ is incident to at most k of these edges, and also x_i is incident to at most p_i of these edges. Hence $\sum_{i=1}^{m} \min\{p_i, k\}$ is an upper bound on the number of edges incident to any k vertices of Y, such as those with degrees $q_1, \dots, q_k$.

Sufficiency. Given (p, q), create a network N with an edge of capacity 1 from x_i to y_j for each i, j, and let $\sigma(x_i) = p_i$ and $\partial(y_j) = q_j$. Unit capacity prevents multiple edges, and (p, q) is realizable if and only if N has a feasible flow.

It suffices to show that the stated condition on p and q implies the condition of Theorem 4.3.17. For $S \subseteq V(N)$, let $I(S) = \{i\colon x_i \in S\}$ and $J(S) = \{j\colon y_j \in S\}$. For a partition S, T of $V(N)$, we now have $\sigma(X \cap T) = \sum_{i \in I(T)} p_i$ and $\partial(Y \cap T) = \sum_{j \in J(T)} q_j$, and we have $c([S, T]) = |I(S)| \cdot |J(T)|$.

Letting $k = |J(T)|$, this last quantity becomes

$$c([S,T]) = |I(S)|\,k = \sum_{i\in I(S)} k \geq \sum_{i\in I(S)} \min\{p_i, k\}.$$

Also $\sum_{i\in I(T)} p_i \geq \sum_{i\in I(T)} \min\{p_i, k\}$, and $\sum_{j\in J(T)} q_j \leq \sum_{j=1}^{k} q_j$. Combining these inequalities, the condition $\sum_{i=1}^{m} \min\{p_i, k\} \geq \sum_{j=1}^{k} q_j$ implies $c([S,T]) \geq \partial(Y\cap T) - \sigma(X\cap T)$. Since this holds for each partition S, T, the network has a feasible flow, which yields the desired bipartite graph. ■

We can extend the maximum flow problem by imposing a nonnegative lower bound on the permitted flow in each edge. The capacity constraint remains as an upper bound, so we require $l(e) \leq f(e) \leq u(e)$ for the flow $f(e)$. We still impose conservation constraints on the internal nodes. If we have a feasible flow, then an easy modification of the Ford–Fulkerson labeling algorithm allows us to find a maximum (or minimum) feasible flow (Exercise 4). The difficulty is finding an initial feasible flow. First we present an application.

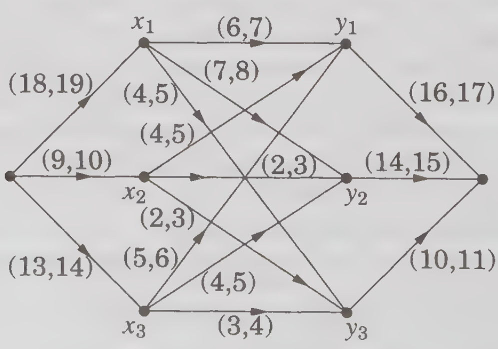
4.3.19. Application. *Matrix rounding* (Bacharach [1966]). We may want to round the entries of a data matrix up or down to integers. We also want to present integers for the row sums and column sums. The sum of each rounded row or column should be a rounding of the original sum. The resulting integer matrix, if it exists, is a **consistent rounding**.

We can represent the consistent rounding problem as a feasible flow problem. Establish vertices $x_1, \ldots, x_n$ for the rows and vertices $y_1, \ldots, y_n$ for the columns of the matrix. Add a source s and a sink t. Add edges sx_i, x_iy_j, y_jt for all values of i and j. If the matrix has entries $a_{i,j}$ with row-sums $r_1, \ldots, r_n$ and column-sums $s_1, \ldots, s_n$, set

$$\begin{array}{lll} l(sx_i) = \lfloor r_i \rfloor & l(x_iy_j) = \lfloor a_{i,j} \rfloor & l(y_jt) = \lfloor c_j \rfloor \\ u(sx_i) = \lceil r_i \rceil & u(x_iy_j) = \lceil a_{i,j} \rceil & u(y_jt) = \lceil c_j \rceil \end{array}$$

We test for a feasible flow by transforming again to an ordinary maximum flow problem. With these two transformations, we can use network flow to test for the existence of a consistent rounding. ■

$$\begin{pmatrix} 6.3 & 7.6 & 4.6 \\ 4.7 & 2.3 & 2.8 \\ 5.5 & 4.5 & 3.5 \end{pmatrix}$$



4.3.20. Solution. *Circulations and flows with lower bounds.* In a maximum flow problem with upper and lower bounds on edge capacities, the zero flow is not feasible, so the Ford–Fulkerson labeling algorithm has no place to start. We must first obtain a feasible flow, after which an easy modification of the labeling algorithm applies (Exercise 4).

The first step is to add an edge of infinite capacity from the sink to the source. The resulting network has a feasible flow with conservation at *every* node (called a **circulation**) if and only if the original network has a feasible flow. In a circulation problem, there is no source or sink.

Next, we convert a feasible circulation problem C into a maximum flow problem N by introducing supplies or demands at the nodes and adding a source and sink to satisfy the supplies and demands. Given the flow constraints $l(e) \le f(e) \le u(e)$, let $c(e) = u(e) - l(e)$ for each edge e. For each vertex v, let

$$\begin{aligned} l^-(v) &= \textstyle\sum_{e\in[V(C)-v,v]} l(e), \\ l^+(v) &= \textstyle\sum_{e\in[v,V(C)-v]} l(e), \\ b(v) &= l^-(v) - l^+(v). \end{aligned}$$

Since each $l(uv)$ contributes to $l^+(u)$ and $l^-(v)$, we have $\sum b(v) = 0$. A feasible circulation f must satisfy the flow constraints at each edge and satisfy $f^+(v) - f^-(v) = 0$ at each node. Letting $f'(e) = f(e) - l(e)$, we find that f is a feasible circulation in C if and only if f' satisfies $0 \le f'(e) \le c(e)$ on each edge and $f'^+(v) - f'^-(v) = b(v)$ at each vertex.

This transforms the feasible circulation problem into a flow problem with supplies and demands. If $b(v) \ge 0$, then v supplies flow $|b(v)|$ to the network; otherwise v demands $|b(v)|$. To restore conservation constraints, we add a source s with an edge of capacity $b(v)$ to each v with $b(v) \ge 0$, and we add a sink t with an edge of capacity $-b(v)$ from each v with $b(v) < 0$. This completes the construction of N.

Let α be the total capacity on the edges leaving s; since $\sum b(v) = 0$, the edges entering t also have total capacity α. Now C has a feasible circulation f if and only if N has a flow of value α (saturating all edges out of s or into t). ■

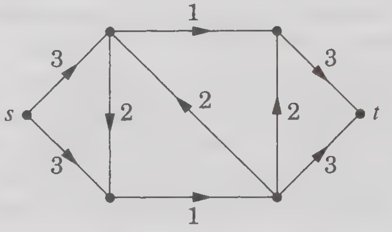
4.3.21. Corollary. A network D with conservation constraints at every node has a feasible circulation if and only if $\sum_{e\in[S,\overline{S}]} l(e) \le \sum_{e\in[\overline{S},S]} u(e)$ for every $S \subseteq V(D)$.

Proof: We can stop before the last step in the discussion of Solution 4.3.20 and interpret our problem with supplies and demands in the model of Theorem 4.3.17. Since $\sum b(v) = 0$, the only way to satisfy all the demands is to use up all the supply. Hence there is a circulation if and only if the supply/demand problem with supplies $\sigma(v) = b(v)$ for $\{v \in V(D) : b(v) \ge 0\}$ and demands $\partial(v) = -b(v)$ for $\{v \in V(D) : b(v) < 0\}$ has a solution.

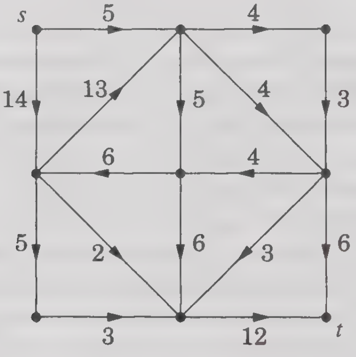
Theorem 4.3.17 characterizes when this problem has a solution. Translated back into the lower and upper bounds on flow in the original problem (Exercise 22), the criterion of Theorem 4.3.17 becomes $\sum_{e\in[S,\overline{S}]} l(e) \le \sum_{e\in[\overline{S},S]} u(e)$ for every $S \subseteq V(D)$. ■

EXERCISES

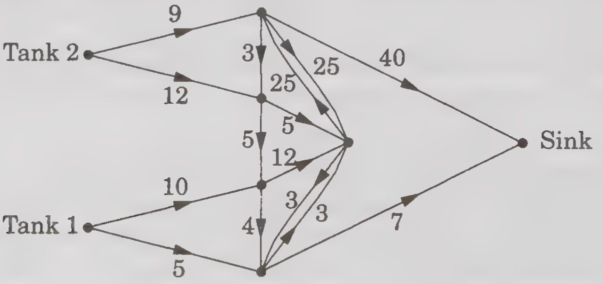
4.3.1. (−) In the network below, list all integer-valued feasible flows and select a flow of maximum value (this illustrates the advantage of duality over exhaustive search). Prove that this flow is a maximum flow by exhibiting a cut with the same value. Determine the number of source/sink cuts. (Comment: There is a nonzero flow with value 0.)



4.3.2. (−) In the network below, find a maximum flow from s to t. Prove that your answer is optimal by using the dual problem, and explain why this proves optimality.



4.3.3. (−) A kitchen sink draws water from two tanks according to the network of pipes with capacities per unit time shown below. Find the maximum flow. Prove that your answer is optimal by using the dual problem, and explain why this proves optimality.



• • • • •

4.3.4. Let N be a network with edge capacity and node conservation constraints plus lower bound constraints $l(e)$ on the flow in edges, meaning that $f(e) \geq l(e)$ is required. If an initial feasible flow is given, how can the Ford–Fulkerson labeling algorithm be modified to search for a maximum feasible flow in this network?

4.3.5. (!) Use network flows to prove Menger's Theorem for internally-disjoint paths in digraphs: $\kappa(x, y) = \lambda(x, y)$ when xy is not an edge. (Hint: Use the first transformation suggested in Remark 4.3.15.)

4.3.6. (!) Use network flows to prove Menger's Theorem for edge-disjoint paths in graphs: $\kappa'(x, y) = \lambda'(x, y)$. (Hint: Use the second transformation suggested in Remark 4.3.15.)

4.3.7. (!) Use network flows to prove Menger's Theorem for nonadjacent vertices in graphs: $\kappa(x, y) = \lambda(x, y)$. (Hint: Use both transformations suggested in Remark 4.3.15.)

4.3.8. Let G be a directed graph with $x, y \in V(G)$. Suppose that capacities are specified *not* on the edges of G, but rather on the *vertices* (other than x, y); for each vertex there is a fixed limit on the total flow through it. There is no restriction on flows in edges. Show how to use ordinary network flow theory to determine the maximum value of a feasible flow from x to y in the vertex-capacitated graph G.

4.3.9. Use network flows to prove that a graph G is connected if and only if for every partition of $V(G)$ into two nonempty sets S, T, there is an edge with one endpoint in S and one endpoint in T. (Comment: Chapter 1 contains an easy direct proof of the conclusion, so this is an example of "using a sledgehammer to squash a bug".)

4.3.10. (!) Use network flows to prove the König–Egerváry Theorem ($\alpha'(G) = \beta(G)$ if G is bipartite).

4.3.11. Show that the Augmenting Path Algorithm for bipartite graphs (Algorithm 3.2.1) is a special case of the Ford–Fulkerson Labeling Algorithm.

4.3.12. Let $[S, \overline{S}]$ and $[T, \overline{T}]$ be source/sink cuts in a network N.

a) Prove that $\text{cap}(S \cup T, \overline{S \cup T}) + \text{cap}(S \cap T, \overline{S \cap T}) \le \text{cap}([S, \overline{S}]) + \text{cap}(T, \overline{T})$. (Hint: Draw a picture and consider contributions from various types of edges.)

b) Suppose that $[S, \overline{S}]$ and $[T, \overline{T}]$ are minimum cuts. Conclude from part (a) that $[S \cup T, \overline{S \cup T}]$ and $[S \cap T, \overline{S \cap T}]$ are also minimum cuts. Conclude also that no edge between $S - T$ and $T - S$ has positive capacity.

4.3.13. (!) Several companies send representatives to a conference; the ith company sends m_i representatives. The organizers of the conference conduct simultaneous networking groups; the jth group can accommodate up to n_j participants. The organizers want to schedule all the participants into groups, but the participants from the same company must be in different groups. The groups need not all be filled.

a) Show how to use network flows to test whether the constraints can be satisfied.

b) Let p be the number of companies, and let q be the number of groups, indexed so that $m_1 \ge \cdots \ge m_p$ and $n_1 \le \cdots \le n_q$. Prove that there exists an assignment of participants to groups that satisfies all the constraints if and only if, for all $0 \le k \le p$ and $0 \le l \le q$, it holds that $k(q - l) + \sum_{j=1}^{l} n_j \ge \sum_{i=1}^{k} m_i$.

4.3.14. In a large university with k academic departments, we must appoint an important committee. One professor will be chosen from each department. Some professors have joint appointments in two or more departments, but each must be the designated representative of at most one department. We must use equally many assistant professors, associate professors, and full professors among the chosen representatives (assume that k is divisible by 3). How can the committee be found? (Hint: Build a network in which units of flow correspond to professors chosen for the committee and capacities enforce the various constraints. Explain how to use the network to test whether such a committee exists and find it if it does.) (Hall [1956])

4.3.15. Let G be a weighted graph. Let the *value* of a spanning tree be the minimum weight of its edges. Let the *cap* from a edge cut $[S, \overline{S}]$ be the maximum weight of its edges. Prove that the maximum value of a spanning tree of G equals the minimum cap of an edge cut in G. (Ahuja–Magnanti–Orlin [1993, p538])

4.3.16. (+) Let x be a vertex of maximum outdegree in a tournament T. Prove that T has a spanning directed tree rooted at x such that every vertex has distance at most 2 from x and every vertex other than x has outdegree at most 2. (Hint: Create a network to model the desired paths to the non-successors of x, and show that every cut has enough capacity. Comment: This strengthens Proposition 1.4.30 about kings in tournaments; no vertex need be an intermediate vertex for more than two others.) (Lu [1996])

4.3.17. (∗−) Use the Gale–Ryser Theorem (Theorem 4.3.18) to determine whether there is a simple bipartite graph in which the vertices in one partite set have degrees (5, 4, 4, 2, 1) and the vertices in the other partite set also have degrees (5, 4, 4, 2, 1).

4.3.18. (∗−) Given list $r = (r_1, \ldots, r_n)$ and $s = (s_1, \ldots, s_n)$, obtain necessary and sufficient conditions for the existence of a digraph D with vertices $v_1, \ldots, v_n$ such that each ordered pair occurs at most once as an edge and $d^+(v_i) = r_i$ and $d^-(v_i) = s_i$ for all i.

4.3.19. (∗−) Find a consistent rounding of the data in the matrix below. Is it unique? (Every entry must be 0 or 1.)

$$\begin{pmatrix} .55 & .6 & .6 \\ .55 & .65 & .7 \\ .6 & .65 & .7 \end{pmatrix}$$

4.3.20. (∗) Prove that every two-by-two matrix can be consistently rounded.

4.3.21. (∗) Suppose that every entry in an n-by-n matrix is strictly between $1/n$ and $1/(n-1)$. Describe all consistent roundings.

4.3.22. (∗) Complete the details of proving Corollary 4.3.21, proving the necessary and sufficient condition for a circulation in a network with lower and upper bounds.

4.3.23. (∗!) A $(k + l)$-regular graph G is (k, l)-**orientable** if it can be oriented so that each indegree is k or l.

a) Prove that G is (k, l)-orientable if and only if there is a partition X, Y of $V(G)$ such that for every $S \subseteq V(G)$,

$$(k - l)(|X \cap S| - |Y \cap S|) \le \left|[S, \overline{S}]\right|.$$

(Hint: Use Theorem 4.3.17.)

b) Conclude that if G is (k, l)-orientable and $k > l$, then G is also $(k - 1, l + 1)$-orientable. (Bondy–Murty [1976, p210–211])

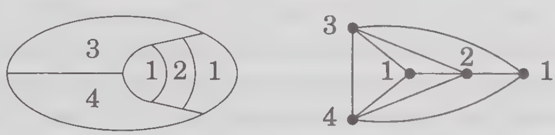
Chapter 5

Coloring of Graphs

5.1. Vertex Coloring and Upper Bounds

The committee-scheduling example (Example 1.1.11) used graph coloring to model avoidance of conflicts. Similarly, in a university we want to assign time slots for final examinations so that two courses with a common student have different slots. The number of slots needed is the chromatic number of the graph in which two courses are adjacent if they have a common student.

Coloring the regions of a map with different colors on regions with common boundaries is another example; we return to it in Chapter 6. The map on the left below has five regions, and four colors suffice. The graph on the right models the "common boundary" relation and the corresponding coloring. Labeling of vertices is our context for coloring problems.



DEFINITIONS AND EXAMPLES

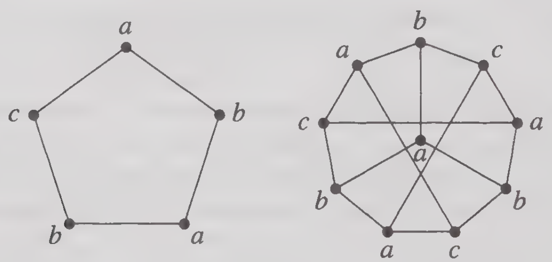
Graph coloring takes its name from the map-coloring application. We assign labels to vertices. When the numerical value of the labels is unimportant, we call them "colors" to indicate that they may be elements of any set.

5.1.1. Definition. A k-**coloring** of a graph G is a labeling $f\colon V(G) \to S$, where $|S| = k$ (often we use $S = [k]$). The labels are **colors**; the vertices of one color form a **color class**. A k-coloring is **proper** if adjacent vertices have different labels. A graph is k-**colorable** if it has a proper k-coloring. The **chromatic number** $\chi(G)$ is the least k such that G is k-colorable.

5.1.2. Remark. In a proper coloring, each color class is an independent set, so G is k-colorable if and only if $V(G)$ is the union of k independent sets. Thus "k-colorable" and "k-partite" have the same meaning. (The usage of the two terms is slightly different. Often "k-partite" is a structural hypothesis, while "k-colorable" is the result of an optimization problem.)

Graphs with loops are uncolorable; we cannot make the color of a vertex different from itself. Therefore, **in this chapter all graphs are loopless.** Also, multiple edges are irrelevant; extra copies don't affect colorings. Thus we usually think in terms of simple graphs when discussing colorings, and we will name edges by their endpoints. Most of the statements made without restriction to simple graphs remain valid when multiple edges are allowed. ■

5.1.3. Example. Since a graph is 2-colorable if and only if it is bipartite, C_5 and the Petersen graph have chromatic number at least 3. Since they are 3-colorable, as shown below, they have chromatic number exactly 3. ■



5.1.4. Definition. A graph G is k**-chromatic** if $\chi(G) = k$. A proper k-coloring of a k-chromatic graph is an **optimal coloring**. If $\chi(H) < \chi(G) = k$ for every proper subgraph H of G, then G is **color-critical** or k**-critical**.

5.1.5. Example. *k-critical graphs for small k.* Properly coloring a graph needs at least two colors if and only if the graph has an edge. Thus K_2 is the only 2-critical graph (similarly, K_1 is the only 1-critical graph). Since 2-colorable is the same as bipartite, the characterization of bipartite graphs implies that the 3-critical graphs are the odd cycles.

We can test 2-colorability of a graph G by computing distances from a vertex x (in each component). Let $X = \{u \in V(G) \colon d(u,x) \text{ is even}\}$, and let $Y = \{u \in V(G) \colon d(u,x) \text{ is odd}\}$. The graph G is bipartite if and only if X, Y is a bipartition, meaning that $G[X]$ and $G[Y]$ are independent sets.

No good characterization of 4-critical graphs or test for 3-colorability is known. Appendix B discusses the computational ramifications. ■

5.1.6. Definition. The **clique number** of a graph G, written $\omega(G)$, is the maximum size of a set of pairwise adjacent vertices (clique) in G.

We have used $\alpha(G)$ for the independence number of G; the usage of $\omega(G)$ is analogous. The letters α and ω are the first and last in the Greek alphabet.

This is consistent with viewing independent sets and cliques as the beginning and end of the "evolution" of a graph (see Section 8.5).

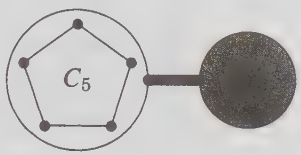
5.1.7. Proposition. For every graph G, $\chi(G) \geq \omega(G)$ and $\chi(G) \geq \frac{n(G)}{\alpha(G)}$.

Proof: The first bound holds because vertices of a clique require distinct colors. The second bound holds because each color class is an independent set and thus has at most $\alpha(G)$ vertices. ■

Both bounds in Proposition 5.1.7 are tight when G is a complete graph.

5.1.8. Example. *$\chi(G)$ may exceed $\omega(G)$.* For $r \geq 2$, let $G = C_{2r+1} \vee K_s$ (the join of C_{2r+1} and K_s—see Definition 3.3.6). Since C_{2r+1} has no triangle, $\omega(G) = s+2$.

Properly coloring the induced cycle requires at least three colors. The s-clique needs s colors. Since every vertex of the induced cycle is adjacent to every vertex of the clique, these s colors must differ from the first three, and $\chi(G) \geq s+3$. We conclude that $\chi(G) > \omega(G)$. ■

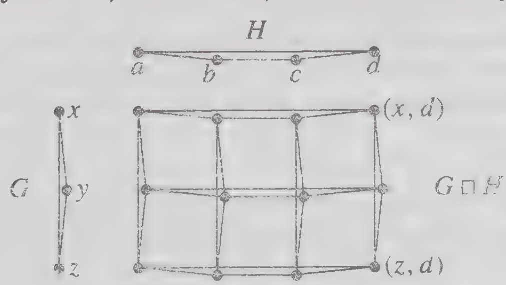


Exercises 23–30 discuss the chromatic number for special families of graphs. We can also ask how it behaves under graph operations. For the disjoint union, $\chi(G+H) = \max\{\chi(G), \chi(H)\}$. For the join, $\chi(G \vee H) = \chi(G) + \chi(H)$. Next we introduce another combining operation.

5.1.9. Definition. The **cartesian product** of G and H, written $G \square H$, is the graph with vertex set $V(G) \times V(H)$ specified by putting (u, v) adjacent to (u', v') if and only if (1) $u = u'$ and $vv' \in E(H)$, or (2) $v = v'$ and $uu' \in E(G)$.

5.1.10. Example. The cartesian product operation is symmetric; $G \square H \cong H \square G$. Below we show $C_3 \square C_4$. The hypercube is another familiar example: $Q_k = Q_{k-1} \square K_2$ when $k \geq 1$. The m-by-n **grid** is the cartesian product $P_m \square P_n$.

In general, $G \square H$ decomposes into copies of H for each vertex of G and copies of G for each vertex of H (Exercise 10). We use $\square$ instead of $\times$ to avoid confusion with other product operations, reserving $\times$ for the cartesian product of vertex sets. The symbol $\square$, due to Rödl, evokes the identity $K_2 \square K_2 = C_4$. ■

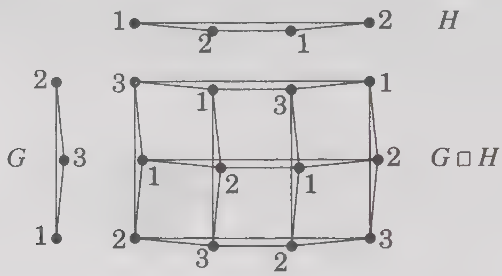


5.1.11. Proposition. (Vizing [1963], Aberth [1964]) $\chi(G\square H)=\max\{\chi(G),\chi(H)\}$.

Proof: The cartesian product $G\square H$ contains copies of G and H as subgraphs, so $\chi(G\square H)\ge\max\{\chi(G),\chi(H)\}$.

Let $k=\max\{\chi(G),\chi(H)\}$. To prove the upper bound, we produce a proper k-coloring of $G\square H$ using optimal colorings of G and H. Let g be a proper $\chi(G)$-coloring of G, and let h be a proper $\chi(H)$-coloring of H. Define a coloring f of $G\square H$ by letting $f(u,v)$ be the congruence class of $g(u)+h(v)$ modulo k. Thus f assigns colors to $V(G\square H)$ from a set of size k.

We claim that f properly colors $G\square H$. If (u,v) and (u',v') are adjacent in $G\square H$, then $g(u)+h(v)$ and $g(u')+h(v')$ agree in one summand and differ by between 1 and k in the other. Since the difference of the two sums is between 1 and k, they lie in different congruence classes modulo k. ■



The cartesian product allows us to compute chromatic numbers by computing independence numbers, because a graph G is m-colorable if and only if the cartesian product $G\square K_m$ has an independent set of size $n(G)$ (Exercise 31).

UPPER BOUNDS

Most upper bounds on the chromatic number come from algorithms that produce colorings. For example, assigning distinct colors to the vertices yields $\chi(G)\le n(G)$. This bound is best possible, since $\chi(K_n)=n$, but it holds with equality only for complete graphs. We can improve a "best-possible" bound by obtaining another bound that is always at least as good. For example, $\chi(G)\le n(G)$ uses nothing about the structure of G; we can do better by coloring the vertices in some order and always using the "least available" color.

5.1.12. Algorithm. (Greedy coloring)
The **greedy coloring** relative to a vertex ordering $v_1,\dots,v_n$ of $V(G)$ is obtained by coloring vertices in the order $v_1,\dots,v_n$, assigning to v_i the smallest-indexed color not already used on its lower-indexed neighbors. ■

5.1.13. Proposition. $\chi(G)\le\Delta(G)+1$.

Proof: In a vertex ordering, each vertex has at most $\Delta(G)$ earlier neighbors, so the greedy coloring cannot be forced to use more than $\Delta(G)+1$ colors. This proves constructively that $\chi(G)\le\Delta(G)+1$. ■

The bound $\Delta(G) + 1$ is the worst upper bound that greedy coloring could produce (although optimal for cliques and odd cycles). Choosing the vertex ordering carefully yields improvements. We can avoid the trouble caused by vertices of high degree by putting them at the beginning, where they won't have many earlier neighbors (see Exercise 36 for a better ordering).

5.1.14. Proposition. (Welsh–Powell [1967]) If a graph G has degree sequence $d_1 \geq \cdots \geq d_n$, then $\chi(G) \leq 1 + \max_i \min\{d_i, i-1\}$.

Proof: We apply greedy coloring to the vertices in nonincreasing order of degree. When we color the ith vertex v_i, it has at most $\min\{d_i, i-1\}$ earlier neighbors, so at most this many colors appear on its earlier neighbors. Hence the color we assign to v_i is at most $1 + \min\{d_i, i-1\}$. This holds for each vertex, so we maximize over i to obtain the upper bound on the maximum color used. ■

The bound in Proposition 5.1.14 is always at most $1 + \Delta(G)$, so this is always at least as good as Proposition 5.1.13. It gives the optimal upper bound in Example 5.1.8, while $1 + \Delta(G)$ does not.

In Proposition 5.1.14, we use greedy coloring with a well-chosen ordering. In fact, every graph G has some vertex ordering for which the greedy algorithm uses only $\chi(G)$ colors (Exercise 33). Usually it is hard to find such an ordering.

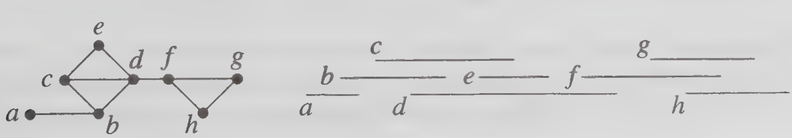
Our next example introduces a class of graphs where such an ordering is easy to find. The ordering produces a coloring that achieves equality in the bound $\chi(G) \geq \omega(G)$.

5.1.15. Example. *Register allocation and interval graphs.* A computer program stores the values of its variables in memory. For arithmetic computations, the values must be entered in easily accessed locations called *registers*. Registers are expensive, so we want to use them efficiently. If two variables are never used simultaneously, then we can allocate them to the same register. For each variable, we compute the first and last time when it is used. A variable is *active* during the interval between these times.

We define a graph whose vertices are the variables. Two vertices are adjacent if they are active at a common time. The number of registers needed is the chromatic number of this graph. The time when a variable is active is an interval, so we we obtain a special type of representation for the graph.,

An **interval representation** of a graph is a family of intervals assigned to the vertices so that vertices are adjacent if and only if the corresponding intervals intersect. A graph having such a representation is an **interval graph**.

For the vertex ordering a, b, c, d, e, f, g, h of the interval graph below, greedy coloring assigns $1, 2, 1, 3, 2, 1, 2, 3$, respectively, which is optimal. Greedy colorings relative to orderings starting $a, d, \ldots$ use four colors. ■



5.1.16. Proposition. If G is an interval graph, then $\chi(G) = \omega(G)$.

Proof: Order the vertices according to the left endpoints of the intervals in an interval representation. Apply greedy coloring, and suppose that x receives k, the maximum color assigned. Since x does not receive a smaller color, the left endpoint a of its interval belongs also to intervals that already have colors 1 through $k-1$. These intervals all share the point a, so we have a k-clique consisting of x and neighbors of x with colors 1 through $k-1$. Hence $\omega(G) \geq k \geq \chi(G)$. Since $\chi(G) \geq \omega(G)$ always, this coloring is optimal. ∎

5.1.17.* Remark. The greedy coloring algorithm runs rapidly. It is "on-line" in the sense that it produces a proper coloring even if it sees only one new vertex at each step and must color it with no option to change earlier colors. For a random vertex ordering in a random graph (see Section 8.5), greedy coloring almost always uses only about twice as many colors as the minimum, although with a bad ordering it may use many colors on a tree (Exercise 34). ∎

We began with greedy coloring to underscore the constructive aspect of upper bounds on chromatic number. Other bounds follow from the properties of k-critical graphs but don't produce proper colorings: every k-chromatic graph *has* a k-critical subgraph, but we have no good algorithm for *finding* one. We derive the next bound using critical subgraphs; it can also be proved using greedy coloring (Exercise 36).

5.1.18. Lemma. If H is a k-critical graph, then $\delta(H) \geq k-1$.

Proof: Let x be a vertex of H. Because H is k-critical, $H - x$ is $k-1$-colorable. If $d_H(x) < k-1$, then the $k-1$ colors used on $H - x$ do not all appear on $N(x)$. We can assign x a color not used on $N(x)$ to obtain a proper $k-1$-coloring of H. This contradicts our hypothesis that $\chi(H) = k$. We conclude that $d_H(x) \geq k-1$ (for each $x \in V(H)$). ∎

5.1.19. Theorem. (Szekeres–Wilf [1968]) If G is a graph, then $\chi(G) \leq 1 + \max_{H \subseteq G} \delta(H)$.

Proof: Let $k = \chi(G)$, and let H' be a k-critical subgraph of G. Lemma 5.1.18 yields $\chi(G) - 1 = \chi(H') - 1 \leq \delta(H') \leq \max_{H \subseteq G} \delta(H)$. ∎

The next bound involves orientations (see also Exercises 43–45).

5.1.20. Example. If G is bipartite, then the orientation of G that directs every edge from one partite set to the other has no path (in the directed sense) of length more than 1. The next theorem thus implies that $\chi(G) \leq 2$.

Every orientation of an odd cycle must somewhere have two consecutive edges in the same direction. Thus each orientation has a path of length at least two, and the theorem confirms that an odd cycle is 3-chromatic. ∎

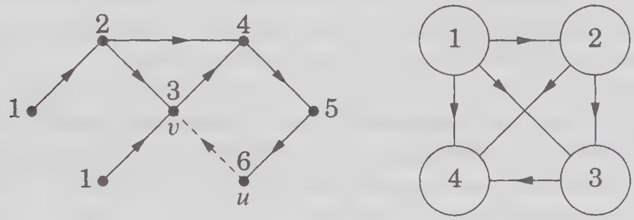
5.1.21. Theorem. Gallai–Roy–Vitaver Theorem (Gallai [1968], Roy [1967], Vitaver [1962]) If D is an orientation of G with longest path length $l(D)$, then $\chi(G) \leq 1 + l(D)$. Furthermore, equality holds for some orientation of G.

Proof: Let D be an orientation of G. Let D' be a maximal subdigraph of D that contains no cycle (in the example below, uv is the only edge of D not in D'). Note that D' includes all vertices of G. Color $V(G)$ by letting $f(v)$ be 1 plus the length of the longest path in D' that ends at v.

Let P be a path in D', and let u be the first vertex of P. Every path in D' ending at u has no other vertex on P, since D' is acyclic. Therefore, each path ending at u (including the longest such path) can be lengthened along P. This implies that f strictly increases along each path in D'.

The coloring f uses colors 1 through $1+l(D')$ on $V(D')$ (which is also $V(G)$). We claim that f is a proper coloring of G. For each $uv \in E(D)$, there is a path in D' between its endpoints (since uv is an edge of D' or its addition to D' creates a cycle). This implies that $f(u) \neq f(v)$, since f increases along paths of D'.

To prove the second statement, we construct an orientation D^* such that $l(D^*) \leq \chi(G) - 1$. Let f be an optimal coloring of G. For each edge uv in G, orient it from u to v in D^* if and only if $f(u) < f(v)$. Since f is a proper coloring, this defines an orientation. Since the labels used by f increase along each path in D^*, and there are only $\chi(G)$ labels in f, we have $l(D^*) \leq \chi(G) - 1$. ■



D' and D $\qquad$ D^*

BROOKS' THEOREM

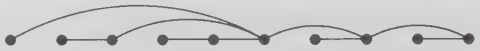
The bound $\chi(G) \leq 1 + \Delta(G)$ holds with equality for complete graphs and odd cycles. By choosing the vertex ordering more carefully, we can show that these are essentially the only such graphs. This implies, for example, that the Petersen graph is 3-colorable, without finding an explicit coloring. To avoid unimportant complications, we phrase the statement only for connected graphs. It extends to all graphs because the chromatic number of a graph is the maximum chromatic number of its components. Many proofs are known; we present a modification of the proof by Lovász [1975].

5.1.22. Theorem. (Brooks [1941]) If G is a connected graph other than a complete graph or an odd cycle, then $\chi(G) \leq \Delta(G)$.

Proof: Let G be a connected graph, and let $k = \Delta(G)$. We may assume that $k \geq 3$, since G is a complete graph when $k \leq 1$, and G is an odd cycle or is bipartite when $k = 2$, in which case the bound holds.

Our aim is to order the vertices so that each has at most $k-1$ lower-indexed neighbors; greedy coloring for such an ordering yields the bound.

When G is not k-regular, we can choose a vertex of degree less than k as v_n. Since G is connected, we can grow a spanning tree of G from v_n, assigning indices in decreasing order as we reach vertices. Each vertex other than v_n in the resulting ordering $v_1, \ldots, v_n$ has a higher-indexed neighbor along the path to v_n in the tree. Hence each vertex has at most $k-1$ lower-indexed neighbors, and the greedy coloring uses at most k colors.



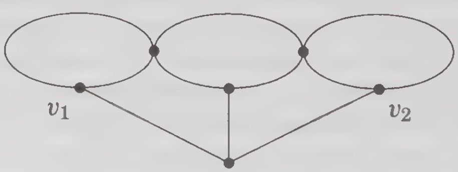
In the remaining case, G is k-regular. Suppose first that G has a cut-vertex x, and let G' be a subgraph consisting of a component of $G-x$ together with its edges to x. The degree of x in G' is less than k, so the method above provides a proper k-coloring of G'. By permuting the names of colors in the subgraphs resulting in this way from components of $G-x$, we can make the colorings agree on x to complete a proper k-coloring of G.

We may thus assume that G is 2-connected. In every vertex ordering, the last vertex has k earlier neighbors. The greedy coloring idea may still work if we arrange that two neighbors of v_n get the same color.

In particular, suppose that some vertex v_n has neighbors v_1, v_2 such that $v_1 \not\leftrightarrow v_2$ and $G - \{v_1, v_2\}$ is connected. In this case, we index the vertices of a spanning tree of $G - \{v_1, v_2\}$ using $3, \ldots, n$ such that labels increase along paths to the root v_n. As before, each vertex before v_n has at most $k-1$ lower indexed neighbors. The greedy coloring also uses at most $k-1$ colors on neighbors of v_n, since v_1 and v_2 receive the same color.

Hence it suffices to show that every 2-connected k-regular graph with $k \geq 3$ has such a triple v_1, v_2, v_n. Choose a vertex x. If $\kappa(G-x) \geq 2$, let v_1 be x and let v_2 be a vertex with distance 2 from x. Such a vertex v_2 exists because G is regular and is not a complete graph; let v_n be a common neighbor of v_1 and v_2.

If $\kappa(G-x) = 1$, let $v_n = x$. Since G has no cut-vertex, x has a neighbor in every leaf block of $G-x$. Neighbors v_1, v_2 of x in two such blocks are nonadjacent. Also, $G - \{x, v_1, v_2\}$ is connected, since blocks have no cut-vertices. Since $k \geq 3$, vertex x has another neighbor, and $G - \{v_1, v_2\}$ is connected. ■



5.1.23.* Remark. The bound $\chi(G) \leq \Delta(G)$ can be improved when G has no large clique (Exercise 50). Brooks' Theorem implies that the complete graphs and odd cycles are the only $k-1$-regular k-critical graphs (Exercise 47). Gallai

[1963b] strengthened this by proving that in the subgraph of a k-critical graph induced by the vertices of degree $k-1$, every block is a clique or an odd cycle.

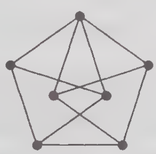
Brooks' Theorem states that $\chi(G) \le \Delta(G)$ whenever $3 \le \omega(G) \le \Delta(G)$. Borodin and Kostochka [1977] conjectured that $\omega(G) < \Delta(G)$ implies $\chi(G) < \Delta(G)$ if $\Delta(G) \ge 9$ (examples show that the condition $\Delta(G) \ge 9$ is needed). Reed [1999] proved that this is true when $\Delta(G) \ge 10^{14}$.

Reed [1998] also conjectured that the chromatic number is bounded by the average of the trivial upper and lower bounds; that is, $\chi(G) \le \left\lceil \frac{\Delta(G)+1+\omega(G)}{2} \right\rceil$. ■

Because the idea of partitioning to satisfy constraints is so fundamental, there are many, many variations and generalizations of graph coloring. In Chapter 7 we consider coloring the edges of a graph. Sticking to vertices, we could allow color classes to induce subgraphs other than independent sets ("generalized coloring"—Exercises 49–53). We could restrict the colors allowed to be used on each vertex ("list coloring"—Section 8.4). We could ask questions involving numerical values of the colors (Exercise 54). We have only touched the tip of the iceberg on coloring problems.

EXERCISES

5.1.1. (−) Compute the clique number, the independence number, and the chromatic number of the graph below. Does either bound in Proposition 5.1.7 prove optimality for some proper coloring? Is the graph color-critical?



5.1.2. (−) Prove that the chromatic number of a graph equals the maximum of the chromatic numbers of its components.

5.1.3. (−) Let $G_1, \dots, G_k$ be the blocks of a graph G. Prove that $\chi(G) = \max_i \chi(G_i)$.

5.1.4. (−) Exhibit a graph G with a vertex v so that $\chi(G-v) < \chi(G)$ and $\chi(\overline{G}-v) < \chi(\overline{G})$.

5.1.5. (−) Given graphs G and H, prove that $\chi(G+H) = \max\{\chi(G), \chi(H)\}$ and that $\chi(G \vee H) = \chi(G) + \chi(H)$.

5.1.6. (−) Suppose that $\chi(G) = \omega(G) + 1$, as in Example 5.1.8. Let $H_1 = G$ and $H_k = H_{k-1} \vee G$ for $k > 1$. Prove that $\chi(H)_k = \omega(H)_k + k$.

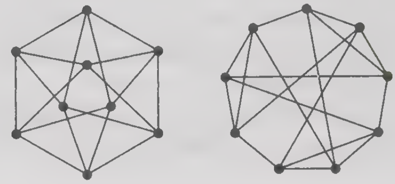
5.1.7. (−) Construct a graph G that is neither a clique nor an odd cycle but has a vertex ordering relative to which greedy coloring uses $\Delta(G)+1$ colors.

5.1.8. (−) Prove that $\max_{H \subseteq G} \delta(H) \le \Delta(G)$ to explain why Theorem 5.1.19 is better than Proposition 5.1.13. Determine all graphs G such that $\max_{H \subseteq G} \delta(H) = \Delta(G)$.

5.1.9. (−) Draw the graph $K_{1,3} \,\square\, P_3$ and exhibit an optimal coloring of it. Draw $C_5 \,\square\, C_5$ and find a proper 3-coloring of it with color classes of sizes 9, 8, 8.

5.1.10. (−) Prove that $G \square H$ decomposes into $n(G)$ copies of H and $n(H)$ copies of G.

5.1.11. (−) Prove that each graph below is isomorphic to $C_3 \square C_3$.



5.1.12. (−) Prove or disprove: Every k-chromatic graph G has a proper k-coloring in which some color class has $\alpha(G)$ vertices.

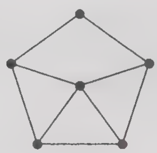
5.1.13. (−) Prove or disprove: If $G = F \cup H$, then $\chi(G) \le \chi(F) + \chi(H)$.

5.1.14. (−) Prove or disprove: For every graph G, $\chi(G) \le n(G) - \alpha(G) + 1$.

5.1.15. (−) Prove or disprove: If G is a connected graph, then $\chi(G) \le 1 + a(G)$, where $a(G)$ is the average of the vertex degrees in G.

5.1.16. (−) Use Theorem 5.1.21 to prove that every tournament has a spanning path. (Rédei [1934])

5.1.17. (−) Use Lemma 5.1.18 to prove that $\chi(G) \le 4$ for the graph G below.



5.1.18. (−) Determine the number of colors needed to label $V(K_n)$ such that each color class induces a subgraph with maximum degree at most k.

5.1.19. (−) Find the error in the false argument below for Brooks' Theorem (Theorem 5.1.22).

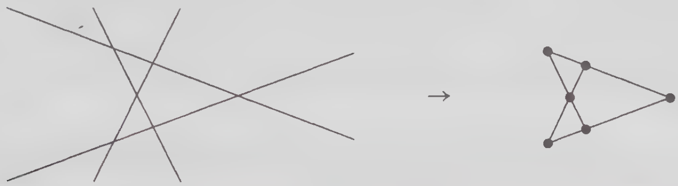
"We use induction on $n(G)$; the statement holds when $n(G) = 1$. For the induction step, suppose that G is not a complete graph or an odd cycle. Since $\kappa(G) \le \delta(G)$, the graph G has a separating set S of size at most $\Delta(G)$. Let $G_1, \ldots, G_m$ be the components of $G - S$, and let $H_i = G[V(G_i) \cup S]$. By the induction hypothesis, each H_i is $\Delta(G)$-colorable. Permute the names of the colors used on these subgraphs to agree on S. This yields a proper $\Delta(G)$-coloring of G."

• • • • •

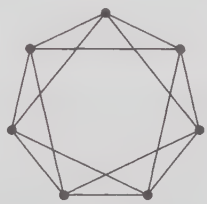
5.1.20. (!) Let G be a graph whose odd cycles are pairwise intersecting, meaning that every two odd cycles in G have a common vertex. Prove that $\chi(G) \le 5$.

5.1.21. Suppose that every edge of a graph G appears in at most one cycle. Prove that every block of G is an edge, a cycle, or an isolated vertex. Use this to prove that $\chi(G) \le 3$.

5.1.22. (!) Given a set of lines in the plane with no three meeting at a point, form a graph G whose vertices are the intersections of the lines, with two vertices adjacent if they appear consecutively on one of the lines. Prove that $\chi(G) \le 3$. (Hint: This can be solved by using the Szekeres–Wilf Theorem or by using greedy coloring with an appropriate vertex ordering. Comment: The conclusion may fail when three lines are allowed to share a point.) (H. Sachs)



5.1.23. (!) Place n points on a circle, where $n \geq k(k+1)$. Let $G_{n,k}$ be the $2k$-regular graph obtained by joining each point to the k nearest points in each direction on the circle. For example, $G_{n,1} = C_n$, and $G_{7,2}$ appears below. Prove that $\chi(G_{n,k}) = k+1$ if $k+1$ divides n and $\chi(G_{n,k}) = k+2$ if $k+1$ does not divide n. Prove that the lower bound on n cannot be weakened, by proving that $\chi(G_{k(k+1)-1,k}) > k+2$ if $k \geq 2$.



5.1.24. (+) Let G be any 20-regular graph with 360 vertices formed in the following way. The vertices are evenly-spaced around a circle. Vertices separated by 1 or 2 degrees are nonadjacent. Vertices separated by 3, 4, 5 or 6 degrees are adjacent. No information is given about other adjacencies (except that G is 20-regular). Prove that $\chi(G) \leq 19$. (Hint: Color successive vertices in order around the circle.) (Pritikin)

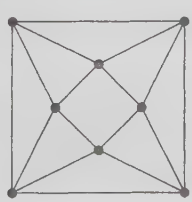
5.1.25. (+) Let G be the **unit-distance graph** in the plane; $V(G) = \mathbb{R}^2$, and two points are adjacent if their Euclidean distance is 1 (this is an infinite graph). Prove that $4 \leq \chi(G) \leq 7$. (Hint: For the upper bound, present an explicit coloring by regions, paying attention to the boundaries.) (Hadwiger [1945, 1961], Moser–Moser [1961])

5.1.26. Given finite sets $S_1, \ldots, S_m$, let $U = S_1 \times \cdots \times S_m$. Define a graph G with vertex set U by putting $u \leftrightarrow v$ if and only if u and v differ in every coordinate. Determine $\chi(G)$.

5.1.27. Let H be the complement of the graph in Exercise 5.1.26. Determine $\chi(H)$.

5.1.28. Consider a traffic signal controlled by two switches, each of which can be set in n positions. For each setting of the switches, the traffic signal shows one of its n possible colors. Whenever the setting of *both* switches changes, the color changes. Prove that the color shown is determined by the position of one of the switches. Interpret this in terms of the chromatic number of some graph. (Greenwell–Lovász [1974])

5.1.29. For the graph G below, compute $\chi(G)$ and find a $\chi(G)$-critical subgraph.



5.1.30. (+) Let $S = \binom{[n]}{2}$ denote the collection of 2-sets of the n-element set $[n]$. Define the graph G_n by $V(G_n) = S$ and $E(G_n) = \{(ij, jk)\colon 1 \le i < j < k \le n\}$ (disjoint pairs, for example, are nonadjacent). Prove that $\chi(G_n) = \lceil \lg n \rceil$. (Hint: Prove that G_n is r-colorable if and only if $[r]$ has at least n distinct subsets. Comment: G_n is called the **shift graph** of K_n.) (attributed to A. Hajnal)

5.1.31. (!) Prove that a graph G is m-colorable if and only if $\alpha(G \,\square\, K_m) \ge n(G)$. (Berge [1973, p379–80])

5.1.32. (!) Prove that a graph G is 2^k-colorable if and only if G is the union of k bipartite graphs. (Hint: This generalizes Theorem 1.2.23.)

5.1.33. (!) Prove that every graph G has a vertex ordering relative to which greedy coloring uses $\chi(G)$ colors.

5.1.34. (!) For all $k \in \mathbb{N}$, construct a tree T_k with maximum degree k and an ordering σ of $V(T_k)$ such that greedy coloring relative to the ordering σ uses $k+1$ colors. (Hint: Use induction and construct the tree and ordering simultaneously. Comment: This result shows that the performance ratio of greedy coloring to optimal coloring can be as bad as $(\Delta(G)+1)/2$.) (Bean [1976])

5.1.35. Let G be a graph having no induced subgraph isomorphic to P_4. Prove that for every vertex ordering, greedy coloring produces an optimal coloring of G. (Hint: Suppose that the algorithm uses k colors for the ordering $v_1, \ldots, v_n$, and let i be the smallest integer such that G has a clique consisting of vertices assigned colors i through k in this coloring. Prove that $i = 1$. Comment: P_4-free graphs are also called **cographs**.)

5.1.36. Given a vertex ordering $\sigma = v_1, \ldots, v_n$ of a graph G, let $G_i = G[\{v_1, \ldots, v_i\}]$ and $f(\sigma) = 1 + \max_i d_{G_i}(v_i)$. Greedy coloring relative to σ yields $\chi(G) \le f(\sigma)$. Define σ^* by letting v_n be a minimum degree vertex of G and letting v_i for $i < n$ be a minimum degree vertex of $G - \{v_{i+1}, \ldots, v_n\}$. Show that $f(\sigma^*) = 1 + \max_{H \subseteq G} \delta(H)$, and thus that σ^* minimizes $f(\sigma)$. (Halin [1967], Matula [1968], Finck–Sachs [1969], Lick–White [1970])

5.1.37. Prove that $V(G)$ can be partitioned into $1 + \max_{H \subseteq G} \delta(H)/r$ classes such that every subgraph whose vertices lie in a single class has a vertex of degree less than r. (Hint: Consider ordering σ^* of Exercise 5.1.36. Comment: This generalizes Theorem 5.1.19. See also Chartrand–Kronk [1969] when $r = 2$.)

5.1.38. (!) Prove that $\chi(G) = \omega(G)$ when $\overline{G}$ is bipartite. (Hint: Phrase the claim in terms of $\overline{G}$ and apply results on bipartite graphs.)

5.1.39. (!) Prove that every k-chromatic graph has at least $\binom{k}{2}$ edges. Use this to prove that if G is the union of m complete graphs of order m, then $\chi(G) \le 1 + m\sqrt{m-1}$. (Comment: This bound is near tight, but the Erdős–Faber–Lovász Conjecture (see Erdős [1981]) asserts that $\chi(G) = m$ when the complete graphs are pairwise edge-disjoint.)

5.1.40. Prove that $\chi(G) \cdot \chi(\overline{G}) \ge n(G)$, use this to prove that $\chi(G) + \chi(\overline{G}) \ge 2\sqrt{n(G)}$, and provide a construction achieving these bounds whenever $\sqrt{n(G)}$ is an integer. (Nordhaus–Gaddum [1956], Finck [1968])

5.1.41. (!) Prove that $\chi(G) + \chi(\overline{G}) \le n(G) + 1$. (Hint: Use induction on $n(G)$.) (Nordhaus–Gaddum [1956])

5.1.42. (!) *Looseness of* $\chi(G) \ge n(G)/\alpha(G)$. Let G be an n-vertex graph, and let $c = (n+1)/\alpha(G)$. Use Exercise 5.1.41 to prove that $\chi(G) \cdot \chi(\overline{G}) \le (n+1)^2/4$, and use this to prove that $\chi(G) \le c(n+1)/4$. For each odd n, construct a graph such that $\chi(G) = c(n+1)/4$. (Nordhaus–Gaddum [1956], Finck [1968])

5.1.43. (!) *Paths and chromatic number in digraphs.*

a) Let $G = F \cup H$. Prove that $\chi(G) \le \chi(F)\chi(H)$.

b) Consider an orientation D of G and a function $f\colon V(G) \to \mathbb{R}$. Use part (a) and Theorem 5.1.21 to prove that if $\chi(G) > rs$, then D has a path $u_0 \to \cdots \to u_r$ with $f(u_0) \le \cdots \le f(u_r)$ or a path $v_0 \to \cdots \to v_s$ with $f(v_0) > \cdots > f(v_s)$.

c) Use part (b) to prove that every sequence of $rs + 1$ distinct real numbers has an increasing subsequence of size $r + 1$ or a decreasing subsequence of size $s + 1$. (Erdős–Szekeres [1935])

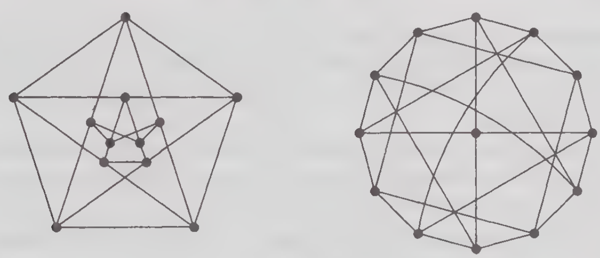
5.1.44. (!) *Minty's Theorem* (Minty [1962]). An **acyclic orientation** of a loopless graph is an orientation having no cycle. For each acyclic orientation D of G, let $r(D) = \max_C \lceil a/b \rceil$, where C is a cycle in G and a, b count the edges of C that are forward in D or backward in D, respectively. Fix a vertex $x \in V(G)$, and let W be a walk in G beginning at x. Let $g(W) = a - b \cdot r(D)$, where a is the number of steps along W that are forward edges in D and b is the number that are backward in D. For each $y \in V(G)$, let $g(y)$ be the maximum of $g(W)$ such that W is an x, y-walk (assume that G is connected).

a) Prove that $g(y)$ is finite and thus well-defined, and use $g(y)$ to obtain a proper $1 + r(D)$-coloring of G. Thus G is $1 + r(D)$-colorable.

b) Prove that $\chi(G) = \min_{D \in \mathbf{D}}$, where $\mathbf{D}$ is the set of acyclic orientations of G.

5.1.45. (+) Use Minty's Theorem (Exercise 5.1.44) to prove Theorem 5.1.21. (Hint: Prove that $l(D)$ is maximized by some acyclic orientation of G.)

5.1.46. (+) Prove that the 4-regular triangle-free graphs below are 4-chromatic. (Hint: Consider the maximum independent sets. Comment: Chvátal [1970] showed that the graph on the left is the smallest triangle-free 4-regular 4-chromatic graph.)



5.1.47. (!) Prove that Brooks' Theorem is equivalent to the following statement: every $k-1$-regular k-critical graph is a complete graph or an odd cycle.

5.1.48. Let G be a simple graph with n vertices and m edges and maximum degree at most 3. Suppose that no component of G is a complete graph on 4 vertices. Prove that G contains a bipartite subgraph with at least $m - n/3$ edges. (Hint: Apply Brooks' Theorem, and then show how to delete a few edges to change a proper 3-coloring of G into a proper 2-coloring of a large subgraph of G.)

5.1.49. (−) Prove that the Petersen graph can be 2-colored so that the subgraph induced by each color class consists of isolated edges and vertices.

5.1.50. (!) *Improvement of Brooks' Theorem.*

a) Given a graph G, let $k_1, \ldots, k_t$ be nonnegative integers with $\sum k_i \ge \Delta(G) - t + 1$. Prove that $V(G)$ can be partitioned into sets $V_1, \ldots, V_t$ so that for each i, the subgraph G_i induced by V_i has maximum degree at most k_i. (Hint: Prove that the partition minimizing $\sum e(G_i)/k_i$ has the desired property.) (Lovász [1966])

b) For $4 \le r \le \Delta(G)+1$, use part (a) to prove that $\chi(G) \le \left\lceil \frac{r-1}{r}(\Delta(G)+1) \right\rceil$ when G has no r-clique. (Borodin–Kostochka [1977], Catlin [1978], Lawrence [1978])

5.1.51. (!) Let G be an k-colorable graph, and let P be a set of vertices in G such that $d(x, y) \ge 4$ whenever $x, y \in P$. Prove that every coloring of P with colors from $[k+1]$ extends to a proper $k+1$ coloring of G. (Albertson–Moore [1999])

5.1.52. Prove that every graph G can be $\lceil (\Delta(G)+1)/j \rceil$-colored so that each color class induces a subgraph having no j-edge-connected subgraph. For $j > 1$, prove that no smaller number of classes suffices when G is a j-regular j-edge-connected graph or is a complete graph with order congruent to 1 modulo j. (Comment: For $j = 1$, the restriction reduces to ordinary proper coloring.) (Matula [1973])

5.1.53. (+) Let $G_{n,k}$ be the $2k$-regular graph of Exercise 5.1.23. For $k \le 4$, determine the values of n such that $G_{n,k}$ can be 2-colored so that each color class induces a subgraph with maximum degree at most k. (Weaver–West [1994])

5.1.54. Let f be a proper coloring of a graph G in which the colors are natural numbers. The **color sum** is $\sum_{v \in V(G)} f(v)$. Minimizing the color sum may require using more than $\chi(G)$ colors. In the tree below, for example, the best proper 2-coloring has color sum 12, while there is a proper 3-coloring with color sum 11. Construct a sequence of trees in which the kth tree T_k use k colors in a proper coloring that minimizes the color sum. (Kubicka–Schwenk [1989])

5.1.55. (+) *Chromatic number is bounded by one plus longest odd cycle length.*

a) Let G be a 2-connected nonbipartite graph containing an even cycle C. Prove that there exist vertices x, y on C and an x, y-path P internally disjoint from C such that $d_C(x, y) \ne d_P(x, y) \bmod 2$.

b) Let G be a simple graph with no odd cycle having length at least $2k+1$. Prove that if $\delta(G) \ge 2k$, then G has a cycle of length at least $4k$. (Hint: Consider the neighbors of an endpoint of a maximal path.)

c) Let G be a 2-connected nonbipartite graph with no odd cycle longer than $2k-1$. Prove that $\chi(G) \le 2k$. (Erdős–Hajnal [1966])

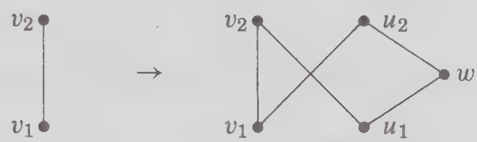
5.2. Structure of k-chromatic Graphs

We have observed that $\chi(H) \ge \omega(H)$ for all H. When equality holds in this bound for G and all its induced subgraphs (as for interval graphs), we say that G is **perfect**; we discuss such graphs in Sections 5.3 and 8.1. Our concern with the bound $\chi(G) \ge \omega(G)$ in this section is how *bad* it can be. Almost always $\chi(G)$ is much larger than $\omega(G)$, in a sense discussed precisely in Section 8.5. (The average values of $\omega(G)$, $\alpha(G)$, and $\chi(G)$ over all graphs with vertex set $[n]$ are very close to $2\lg n$, $2\lg n$, and $n/(2\lg n)$, respectively. Hence $\omega(G)$ is generally a bad lower bound on $\chi(G)$, and $n/\alpha(G)$ is generally a good lower bound.)

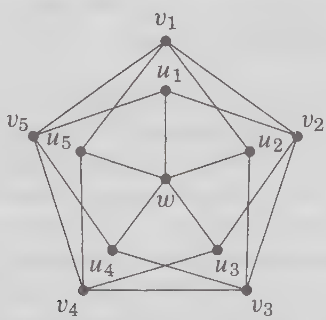
GRAPHS WITH LARGE CHROMATIC NUMBER

The bound $\chi(G) \geq \omega(G)$ can be tight, but it can also be very loose. There have been many constructions of graphs without triangles that have arbitrarily large chromatic number. We present one such construction here; others appear in Exercises 12–13.

5.2.1. Definition. From a simple graph G, **Mycielski's construction** produces a simple graph G' containing G. Beginning with G having vertex set $\{v_1, \ldots, v_n\}$, add vertices $U = \{u_1, \ldots, u_n\}$ and one more vertex w. Add edges to make u_i adjacent to all of $N_G(v_i)$, and finally let $N(w) = U$.



5.2.2. Example. From the 2-chromatic graph K_2, one iteration of Mycielski's construction yields the 3-chromatic graph C_5, as shown above. Below we apply the construction to C_5, producing the 4-chromatic **Grötzsch graph**. ■



5.2.3. Theorem. (Mycielski [1955]) From a k-chromatic triangle-free graph G, Mycielski's construction produces a $k+1$-chromatic triangle-free graph G'.

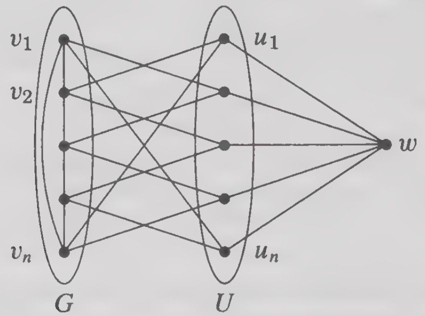
Proof: Let $V(G) = \{v_1, \ldots, v_n\}$, and let G' be the graph produced from it by Mycielski's construction. Let $u_1, \ldots, u_n$ be the copies of $v_1, \ldots, v_n$, with w the additional vertex. Let $U = \{u_1, \ldots, u_n\}$.

By construction, U is an independent set in G'. Hence the other vertices of any triangle containing u_i belong to $V(G)$ and are neighbors of v_i. This would complete a triangle in G, which can't exist. We conclude that G' is triangle-free.

A proper k-coloring f of G extends to a proper $k+1$-coloring of G' by setting $f(u_i) = f(v_i)$ and $f(w) = k+1$; hence $\chi(G') \leq \chi(G) + 1$. We prove equality by showing that $\chi(G) < \chi(G')$. To prove this we consider any proper coloring of G' and obtain from it a proper coloring of G using fewer colors.

Let g be a proper k-coloring of G' By changing the names of colors, we may assume that $g(w) = k$. This restricts g to $\{1, \ldots, k-1\}$ on U. On $V(G)$, it may use all k colors. Let A be the set of vertices in G on which g uses color k; we change the colors used on A to obtain a proper $k-1$-coloring of G.

For each $v_i \in A$, we change the color of v_i to $g(u_i)$. Because all vertices of A have color k under g, no two edges of A are adjacent. Thus we need only check edges of the form $v_i v'$ with $v_i \in A$ and $v' \in V(G) - A$. If $v' \leftrightarrow v_i$, then by construction also $v' \leftrightarrow u_i$, which yields $g(v') \neq g(u_i)$. Since we change the color on v_i to $g(u_i)$, our change does not violate the edge $v_i v'$. We have shown that the modified coloring of $V(G)$ is a proper $k-1$-coloring of G. ■



If G is color-critical, then the graph G' resulting from Mycielski's construction is also color-critical (Exercise 9).

5.2.4.* Remark. Starting with $G_2 = K_2$, iterating Mycielski's construction produces a sequence $G_2, G_3, G_4, \ldots$ of graphs. The first three are K_2, C_5, and the Grötzsch graph. These are the smallest triangle-free 2-chromatic, 3-chromatic, and 4-chromatic graphs. The graphs then grow rapidly: $n(G_k) = 2n(G_{k-1}) + 1$. With $n(G_2) = 2$, this yields $n(G_k) = 3 \cdot 2^{k-2} - 1$ (exponential growth).

Let $f(k)$ be the minimum number of vertices in a triangle-free k-chromatic graph. Using probabilistic (non-constructive) methods, Erdős [1959] proved that $f(k) \le ck^{2+\epsilon}$, where ϵ is any positive constant and c depends on ϵ but not on k. Using Ramsey numbers (Section 8.3), it is now known (non-constructively) that there are constants c_1, c_2 such that $c_1 k^2 \log k \le f(k) \le c_2 k^2 \log k$. Exercise 15 develops a quadratic lower bound.

Blanche Descartes† [1947, 1954] constructed color-critical graphs with girth 6 (Exercise 13). Using probabilistic methods, Erdős [1959] proved that graphs exist with chromatic number at least k and girth at least g (Theorem 8.5.11). Later, explicit constructions were found (Lovász [1968a], Nešetřil–Rödl [1979], Lubotzsky–Phillips–Sarnak [1988], Kriz [1989]).

By all these constructions, forbidding K_r from G does not place a bound on $\chi(G)$. Gyárfás [1975] and Sumner [1981]) conjectured that forbidding a fixed clique and a fixed forest as an *induced* subgraph does bound the chromatic number. Exercise 11 proves this when the forest is $2K_2$. (See also Kierstead–Penrice [1990, 1994], Kierstead [1992, 1997], Kierstead–Rödl [1996]) ■

†This pseudonym was used by W.T. Tutte and also by three others.

EXTREMAL PROBLEMS AND TURÁN'S THEOREM

Perhaps extremal questions can shed some light on the structure of k-chromatic graphs. For example, which are the smallest and largest k-chromatic graphs with n vértices?

5.2.5. Proposition. Every k-chromatic graph with n vertices has at least $\binom{k}{2}$ edges. Equality holds for a complete graph plus isolated vertices.

Proof: An optimal coloring of a graph has an edge with endpoints of colors i and j for each pair i, j of colors. Otherwise, colors i and j could be combined into a single color class and use fewer colors. Since there are $\binom{k}{2}$ distinct pairs of colors, there must be $\binom{k}{2}$ distinct edges. ■

Exercise 6 asks for the minimum size among connected k-chromatic graphs with n vertices.

The maximization problem is more interesting (of course, it makes sense only when restricted to simple graphs). Given a proper k-coloring, we can continue to add edges without increasing the chromatic number as long as two vertices in different color classes are nonadjacent. Thus we may restrict our attention to graphs without such pairs.

5.2.6. Definition. A **complete multipartite graph** is a simple graph G whose vertices can be partitioned into sets so that $u \leftrightarrow v$ if and only if u and v belong to different sets of the partition. Equivalently, every component of $\overline{G}$ is a complete graph. When $k \geq 2$, we write $K_{n_1,\ldots,n_k}$ for the complete k-partite graph with partite sets of sizes $n_1, \ldots, n_k$ and complement $K_{n_1} + \cdots + K_{n_k}$.

We use this notation only for $k > 1$, since K_n denotes a complete graph. A complete k-partite graph is k-chromatic; the partite sets are the color classes in the only proper k-coloring. Also, since a vertex in a partite set of size t has degree $n(G) - t$, the edges can be counted using the degree-sum formula (Exercise 18). Which distribution of vertices to partite sets maximizes $e(G)$?

5.2.7. Example. *The Turán graph.* The **Turán graph** $T_{n,r}$ is the complete r-partite graph with n vertices whose partite sets differ in size by at most 1. By the pigeonhole principle (see Appendix A), some partite set has size at least $\lceil n/r \rceil$ and some has size at most $\lfloor n/r \rfloor$. Therefore, differing by at most 1 means that they all have size $\lfloor n/r \rfloor$ or $\lceil n/r \rceil$.

Let $a = \lfloor n/r \rfloor$. After putting a vertices in each partite set, $b = n - ra$ remain, so $T_{n,r}$ has b partite sets of size $a + 1$ and $r - b$ partite sets of size a. Thus the defining condition on $T_{n,r}$ specifies a single isomorphism class. ■

5.2.8. Lemma. Among simple r-partite (that is, r-colorable) graphs with n vertices, the Turán graph is the unique graph with the most edges.

Proof: As noted before Definition 5.2.6, we need only consider complete r-partite graphs. Given a complete r-partite graph with partite sets differing by more than 1 in size, we move a vertex v from the largest class (size i) to the smallest class (size j). The edges not involving v are the same as before, but v gains $i-1$ neighbors in its old class and loses j neighbors in its new class. Since $i-1 > j$, the number of edges increases. Hence we maximize the number of edges only by equalizing the sizes as in $T_{n,r}$. ■

We used the idea of this local alteration previously in Theorem 1.3.19 and in Theorem 1.3.23; we are finding the largest r-partite subgraph of K_n.

What happens if we have more edges and thus force chromatic number at least $r+1$? We have seen that there are graphs with chromatic number $r+1$ that have no triangles. Nevertheless, if we go beyond the maximum number of edges in an r-colorable graph with n vertices, then we are forced not only to use $r+1$ colors but also to have K_{r+1} as a subgraph.

This famous result of Turán generalizes Theorem 1.3.23 and is viewed as the origin of extremal graph theory.

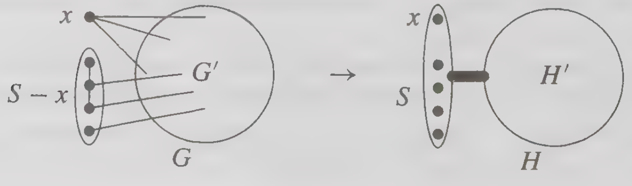
5.2.9. Theorem. (Turán [1941]) Among the n-vertex simple graphs with no $r+1$-clique, $T_{n,r}$ has the maximum number of edges.

Proof: The Turán graph $T_{n,r}$, like every r-colorable graph, has no $r+1$-clique, since each partite set contributes at most one vertex to each clique. If we can prove that the maximum is achieved by an r-partite graph, then Lemma 5.2.8 implies that the maximum is achieved by $T_{n,r}$. Thus it suffices to prove that if G has no $r+1$-clique, then there is an r-partite graph H with the same vertex set as G and at least as many edges.

We prove this by induction on r. When $r=1$, G and H have no edges. For the induction step, consider $r>1$. Let G be an n-vertex graph with no $r+1$-clique, and let $x \in V(G)$ be a vertex of degree $k = \Delta(G)$. Let G' be the subgraph of G induced by the neighbors of x. Since x is adjacent to every vertex in G' and G has no $r+1$-clique, the graph G' has no r-clique. We can thus apply the induction hypothesis to G'; this yields an $r-1$-partite graph H' with vertex set $N(x)$ such that $e(H') \ge e(G')$.

Let H be the graph formed from H' by joining all of $N(x)$ to all of $S = V(G) - N(x)$. Since S is an independent set, H is r-partite. We claim that $e(H) \ge e(G)$. By construction, $e(H) = e(H') + k(n-k)$. We also have $e(G) \le e(G') + \sum_{v\in S} d_G(v)$, since the sum counts each edge of G once for each endpoint it has outside $V(G')$. Since $\Delta(G) = k$, we have $d_G(v) \le k$ for each $v \in S$, and $|S| = n-k$, so $\sum_{v\in S} d_G(v) \le k(n-k)$. As desired, we have

$$e(G) \le e(G') + (n-k)k \le e(H') + k(n-k) = e(H)$$ ■

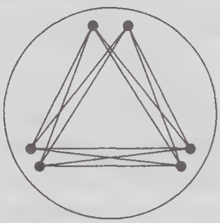


In fact, the Turán graph is the unique extremal graph (Exercise 21). Exercises 16–24 pertain to Turán's Theorem, including alternative proofs, the value of $e(T_{n,r})$, and applications. The argument used in Theorem 1.3.23 was simply one instance of the induction step in Theorem 5.2.9.

Turán's theorem applies to extremal problems when some condition forbids cliques of a given order; we describe a geometric application from Bondy–Murty [1976, p113–115].

5.2.10.* Example. *Distant pairs of points.* In a circular city of diameter 1, we might want to locate n police cars to maximize the number of pairs that are far apart, say separated by distance more than $d = 1/\sqrt{2}$. If six cars occupy equally spaced points on the circle, then the only pairs not more than d apart are the consecutive pairs around the outside: there are nine good pairs.

Instead, putting two cars each near the vertices of an equilateral triangle with side-length $\sqrt{3}/2$ yields three bad pairs and twelve good pairs. (This may not be the socially best criterion!) In general, with $\lceil n/3 \rceil$ or $\lfloor n/3 \rfloor$ cars near each vertex of this triangle, the good pairs correspond to edges of the tripartite Turán graph. We show next that this construction is best. ■

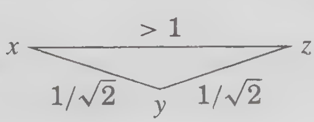


5.2.11.* Application. In a set of n points in the plane with no pair more than distance 1 apart, the maximum number of pairs separated by distance more than $1/\sqrt{2}$ is $\lfloor n^2/3 \rfloor$.

Proof: Draw a graph G on these points by making points adjacent when the distance between them exceeds $1/\sqrt{2}$. By Turán's Theorem and the construction in Example 5.2.10, it suffices to show that G has no K_4.

Among any four points, some three form an angle of at least $90°$: if the four form a convex quadrilateral, then the interior angles sum to $360°$, and if one point is inside the triangle formed by the others, then with them it forms three angles summing to $360°$.

Suppose that G has a 4-vertex clique with points w, x, y, z, where $\angle xyz \geq 90°$. Since the lengths of xy and yz exceed $1/\sqrt{2}$, xz is longer than the hypotenuse of a right triangle with legs of length $1/\sqrt{2}$. Hence the distance between x and z exceeds 1, which contradicts the hypothesis. ■



Even without the full structural statement of Turán's Theorem, one can prove directly a rough bound on the number of edges in an n-vertex graph with no K_{r+1} (Exercise 16). Turning this around yields a sharp lower bound on the chromatic number of a graph in terms of the number of vertices and number of edges (Exercise 17).

COLOR-CRITICAL GRAPHS

The Turán graph solves a problem that is somehow opposite to understanding what forces chromatic number k. It considers the maximal graphs that *avoid* needing k colors instead of the minimal graphs that *do* need k colors.

Every k-chromatic graph has a k-critical subgraph, since we can continue discarding edges and isolated vertices without reducing the chromatic number until we reach a point where every such deletion reduces the chromatic number. Thus knowing the k-critical graphs could help us test for $k-1$-colorability. We begin with elementary properties of k-critical graphs.

5.2.12. Remark. A graph G with no isolated vertices is color-critical if and only if $\chi(G-e) < \chi(G)$ for every $e \in E(G)$. Hence when we prove that a connected graph is color-critical, we need only compare it with subgraphs obtained by deleting a single edge. ■

5.2.13. Proposition. Let G be a k-critical graph.

a) For $v \in V(G)$, there is a proper k-coloring of G in which the color on v appears nowhere else, and the other $k-1$ colors appear on $N(v)$.

b) For $e \in E(G)$, every proper $k-1$-coloring of $G-e$ gives the same color to the two endpoints of e.

Proof: (a) Given a proper $k-1$-coloring f of $G-v$, adding color k on v alone completes a proper k-coloring of G. The other colors must all appear on $N(v)$, since otherwise assigning a missing color to v would complete a proper $k-1$-coloring of G.

(b) If some proper $k-1$-coloring of $G-e$ gave distinct colors to the endpoints of e, then adding e would yield a proper $k-1$-coloring of G. ■

For any graph G, Proposition 5.2.13a holds for every $v \in V(G)$ such that $\chi(G-v) < \chi(G) = k$, and Proposition 5.2.13b holds for every $e \in E(G)$ such that $\chi(G-e) < \chi(G) = k$.

5.2.14. Example. The graph $C_5 \vee K_s$ of Example 5.1.8 is color-critical. In general, the join of two color-critical graphs is always color-critical. This is easy to prove using Remark 5.2.12 and Proposition 5.2.13 by considering cases for the deletion of an edge; the deleted edge e may belong to G or H or have an endpoint in each (Exercise 3). ■

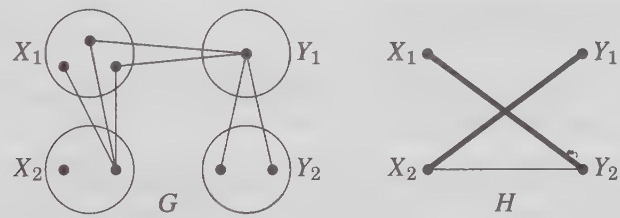
We proved in Lemma 5.1.18 that $\delta(G) \geq k-1$ when G is a k-critical graph. We can strengthen this to $\kappa'(G) \geq k-1$ by using the König–Egerváry Theorem.

5.2.15. Lemma. (Kainen) Let G be a graph with $\chi(G) > k$, and let X, Y be a partition of $V(G)$. If $G[X]$ and $G[Y]$ are k-colorable, then the edge cut $[X, Y]$ has at least k edges.

Proof: Let $X_1, \ldots, X_k$ and $Y_1, \ldots, Y_k$ be the partitions of X and Y formed by the color classes in proper k-colorings of $G[X]$ and $G[Y]$. If there is no edge between X_i and Y_j, then $X_i \cup Y_j$ is an independent set in G. We show that if $|[X, Y]| < k$, then we can combine color classes from $G[X]$ and $G[Y]$ in pairs to form a proper k-coloring of G.

Form a bipartite graph H with vertices $X_1, \ldots, X_k$ and $Y_1, \ldots, Y_k$, putting $X_iY_j \in E(H)$ if in G there is *no edge* between the set X_i and the set Y_j. If $|[X, Y]| < k$, then H has more than $k(k-1)$ edges. Since m vertices can cover at most km edges in a subgraph of $K_{k,k}$, $E(H)$ cannot be covered by $k-1$ vertices. By the König–Egerváry Theorem, H therefore has a perfect matching M.

In G, we give color i to all of X_i and all of the set Y_j to which it is matched by M. Since there are no edges joining X_i and Y_j, doing this for all i produces a proper k-coloring of G, which contradicts the hypothesis that $\chi(G) > k$. Hence we conclude that $|[X, Y]| \geq k$. ■

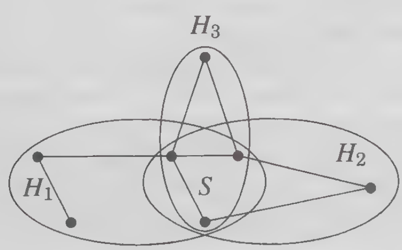


5.2.16. Theorem. (Dirac [1953]) Every k-critical graph is $k-1$-edge-connected.

Proof: Let G be a k-critical graph, and let $[X, Y]$ be a minimum edge cut. Since G is k-critical, $G[X]$ and $G[Y]$ are $k-1$-colorable. Applied with $k-1$ as the parameter, Lemma 5.2.15 then states that $|[X, Y]| \geq k-1$. ■

Although a k-critical graph must be $k-1$-edge-connected, it need not be $k-1$-connected; Exercise 32 shows how to construct k-critical graphs that have connectivity 2. Nevertheless, we can restrict the behavior of small vertex cutsets in k-critical graphs.

5.2.17. Definition. Let S be a set of vertices in a graph G. An **S-lobe** of G is an induced subgraph of G whose vertex set consists of S and the vertices of a component of $G - S$.



For every $S \subseteq V(G)$, the graph G is the union of its S-lobes. We use this to prove a statement about vertex cutsets in k-critical graphs that will be useful in the next theorem. Exercise 33 strengthens the result when $|S| = 2$.

5.2.18. Proposition. If G is k-critical, then G has no cutset consisting of pairwise adjacent vertices. In particular, if G has a cutset $S = \{x, y\}$, then $x \not\leftrightarrow y$ and G has an S-lobe H such that $\chi(H + xy) = k$.

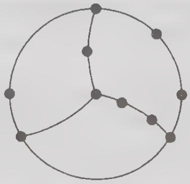
Proof: Let S be a cutset in a k-critical graph G. Let $H_1, \ldots, H_t$ be the S-lobes of G. Since each H_i is a proper subgraph of a k-critical graph, each H_i is $k-1$-colorable. If each H_i has a proper $k-1$-coloring giving distinct colors to the vertices of S, then the names of the colors in these $k-1$-colorings can be permuted to agree on S. The colorings then combine to form a $k-1$-coloring of G, which is impossible.

Hence some S-lobe H has no proper $k-1$-coloring with distinct colors on S. This implies that S is not a clique. If $S = \{x, y\}$, then every $k-1$-coloring of H assigns the same color to x and y, and hence $H + xy$ is not $k-1$-colorable. ■

FORCED SUBDIVISIONS

We need not have a k-clique to have chromatic number k, but perhaps we must have some weakened form of a k-clique.

5.2.19. Definition. An H-**subdivision** (or **subdivision of** H) is a graph obtained from a graph H by successive edge subdivisions (Definition 5.2.19). Equivalently, it is a graph obtained from H by replacing edges with pairwise internally disjoint paths.



a subdivision of K_4

5.2.20. Theorem. (Dirac [1952a]) Every graph with chromatic number at least 4 contains a K_4-subdivision.

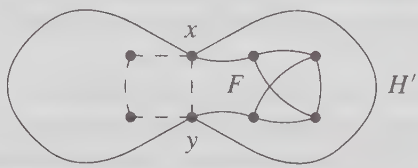
Proof: We use induction on $n(G)$.

Basis step: $n(G) = 4$. The graph G can only be K_4 itself.

Induction step: $n(G) > 4$. Since $\chi(G) \geq 4$, we may let H be a a 4-critical subgraph of G. By Proposition 5.2.18, H has no cut-vertex. If $\kappa(H) = 2$ and $S = \{x, y\}$ is a cutset of size 2, then by Proposition 5.2.18 $x \not\leftrightarrow y$ and H has an S-lobe H' such that $\chi(H' + xy) \geq 4$. Since $n(H') < n(G)$, we can apply the induction hypothesis to obtain a K_4-subdivision in H'.

This K_4-subdivision F appears also in G unless it contains xy (see figure below). In this case, we modify F to obtain a K_4-subdivision in G by replacing the edge xy with an x, y-path through another S-lobe of H. Such a path exists because the minimality of the cutset S implies that each vertex of S has a neighbor in each component of $H - S$.

Hence we may assume that H is 3-connected. Select a vertex $x \in V(G)$. Since $H - x$ is 2-connected, it has a cycle C of length at least 3. (Let x be the central vertex and C the outside cycle in the figure above.) Since H is 3-connected, the Fan Lemma (Theorem 4.2.23) yields an $x, V(C)$-fan of size 3 in H. These three paths, together with C, form a K_4-subdivision in H. ■



5.2.21.* Remark. Hajós [1961] conjectured that every k-chromatic graph contains a subdivision of K_k. For $k = 2$, the statement says that every 2-chromatic graph has a nontrivial path. For $k = 3$, it says that every 3-chromatic graph has a cycle. Theorem 5.2.20 proves it for $k = 4$, and it is open for $k \in \{5, 6\}$.

Hajós' Conjecture is false for $k \geq 7$ (Catlin [1979]—see Exercise 40). Hadwiger [1943] proposed a weaker conjecture: every k-chromatic graph has a subgraph that becomes K_k via edge contractions. This is weaker because a K_k-subdivision is a special subgraph of this type. For $k = 4$, Hadwiger's Conjecture is equivalent to Theorem 5.2.20. For $k = 5$, it is equivalent to the Four Color Theorem (Chapter 6). For $k = 6$, it was proved using the Four Color Theorem by Robertson, Seymour, and Thomas [1993]. For $k \geq 7$, it remains open. ■

Some results about k-critical graphs extend to the larger class of graphs with $\delta(G) \geq k - 1$. For example, every graph with minimum degree at least 3 has a K_4-subdivision (Exercise 38); this strengthens Theorem 5.2.20.

Dirac [1965] and Jung [1965] proved that sufficiently large chromatic number forces a K_k-subdivision in G. Mader improved this by weakening the hypothesis and generalizing the conclusion: for a simple graph F, every simple graph G with $\delta(G) \geq 2^{e(F)}$ contains a subdivision of F. The threshold $2^{e(F)}$ is larger than necessary but permits a short proof.

5.2.22.* Lemma. (Mader [1967], see Thomassen [1988]) If G is a simple graph with minimum degree at least $2k$, then G contains disjoint subgraphs G' and H such that 1) H is connected, 2) $\delta(G') \geq k$, and 3) each vertex of G' has a neighbor in H.

Proof: We may assume that G is connected. Let $G \cdot H'$ denote the graph obtained from G by contracting the edges of a connected subgraph H' and delete extra copies of multiple edges. In $G \cdot H'$, the set $V(H')$ becomes a single vertex. Consider all connected subgraphs H' of G such that $G \cdot H'$ has at least

$k(n(G) - n(H') + 1)$ edges. Since $\delta(G) \geq 2k$, every 1-vertex subgraph of G is such a subgraph. Since such subgraphs exist, we may choose H to be a maximal subgraph with this property.

Let S be the set of vertices outside H with neighbors in H, and let $G' = G[S]$. We need only show that $\delta(G') \geq k$. Each $x \in V(G')$ has a neighbor $y \in V(H)$. In $G \cdot (H \cup xy)$, the edges incident to x in G' collapse onto edges from $V(G')$ to H that appear in $G \cdot H$, and the edge xy contracts. Hence $e(G \cdot H) - e(G \cdot (H \cup xy)) = d_{G'}(x) + 1$. By the choice of H, this difference is more than k, and hence $\delta(G') \geq k$. ■

5.2.23.* Theorem. (Mader [1967], see Thomassen [1988]) If F and G are simple graphs with $e(F) = m$ and $\delta(F) \geq 1$, then $\delta(G) \geq 2^m$ implies that G contains a subdivision of F.

Proof: We use induction on m. The claim is trivial for $m \leq 1$. Consider $m \geq 2$. By Lemma 5.2.22, we may choose disjoint subgraphs H and G' in G such that H is connected, $\delta(G') \geq 2^{m-1}$, and every vertex of G' has a neighbor in H.

If F has an edge $e = xy$ such that $\delta(F - e) \geq 1$, then the induction hypothesis yields a subdivision J of $F - e$ in G'. A path through H can be added between the vertices of J representing x and y to complete a subdivision of F.

If $\delta(F - e) = 0$ for all $e \in E(F)$, then every edge of F is incident to a leaf. Now F is a forest of stars, and $\delta(G) \geq 2^m \geq 2m$ allows us to find F itself in G; we leave this claim to Exercise 42. ■

5.2.24.* Remark. The case when F is a complete graph remains of particular interest. Let $f(k)$ be the minimum d such that every graph with minimum degree at least d contains a K_k-subdivision. Theorem 5.2.23 yields $f(k) \leq 2^{\binom{k}{2}}$. Komlós–Szemerédi [1996] and Bollobás–Thomason [1998] proved that $f(k) < ck^2$ for some constant c (the latter shows $c \leq 256$). Since $K_{m,m-1}$ has no K_{2k}-subdivision when $m = k(k+1)/2$ (Exercise 41), we have $f(k) > k^2/8$.

Exercise 38 yields $f(4) = 3$. Furthermore, $f(5) = 6$. The icosahedron (Exercise 7.3.8) yields $f(5) \geq 6$, since this graph is 5-regular and has no K_5-subdivision. On the other hand, Mader [1998] proved Dirac's conjecture [1964] that every n-vertex graph with at least $3n - 5$ edges contains a K_5-subdivision. By the degree-sum formula, $\delta(G) \geq 6$ yields at least $3n$ edges; hence $f(5) \leq 6$.

Finally, we note that Scott [1997] proved a subdivision version of the Gyárfas–Sumner Conjecture (Remark 5.2.4) for each tree T and integer k: If G has with no k-clique but $\chi(G)$ is sufficiently large, then G contains a subdivision of T as an *induced* subgraph. ■

EXERCISES

5.2.1. (−) Let G be a graph such that $\chi(G - x - y) = \chi(G) - 2$ for all pairs x, y of distinct vertices. Prove that G is a complete graph. (Comment: Lovász conjectured that the conclusion also holds when the condition is imposed only on pairs of adjacent vertices.)

5.2.2. (−) Prove that a simple graph is a complete multipartite graph if and only if it has no 3-vertex induced subgraph with one edge.

5.2.3. (−) The results below imply that there is no k-critical graph with $k+1$ vertices.

a) Let x and y be vertices in a k-critical graph G. Prove that $N(x) \subseteq N(y)$ is impossible. Conclude that no k-critical graph has $k+1$ vertices.

b) Prove that $\chi(G \vee H) = \chi(G) + \chi(H)$, and that $G \vee H$ is color-critical if and only if both G and H are color-critical. Conclude that $C_5 \vee K_{k-3}$, with $k+2$ vertices, is k-critical.

5.2.4. For $n \in \mathbb{N}$, let G be the graph with vertex set $\{v_0, \ldots, v_{3n}\}$ defined by $v_i \leftrightarrow v_j$ if and only if $|i-j| \le 2$ and $i+j$ is not divisible by 6.

a) Determine the blocks of G.

b) Prove that adding the edge v_0v_{3n} to G creates a 4-critical graph.

5.2.5. (−) Find a subdivision of K_4 in the Grötzsch graph (Example 5.2.2).

• • • • •

5.2.6. Determine the minimum number of edges in a connected n-vertex graph with chromatic number k. (Hint: Consider a k-critical subgraph.) (Eršov–Kožuhin [1962]—see Bhasker–Samad–West [1994] for higher connectivity.)

5.2.7. (!) Given an optimal coloring of a k-chromatic graph, prove that for each color i there is a vertex with color i that is adjacent to vertices of the other $k-1$ colors.

5.2.8. Use properties of color-critical graphs to prove Proposition 5.1.14 again: $\chi(G) \le 1 + \max_i \min\{d_i, i-1\}$, where $d_1 \ge \cdots \ge d_n$ are the vertex degrees in G.

5.2.9. (!) Prove that if G is a color-critical graph, then the graph G' generated from it by applying Mycielski's construction is also color-critical.

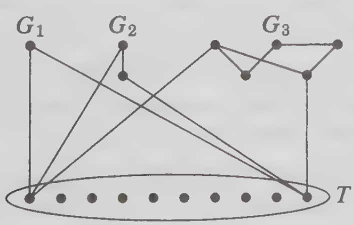
5.2.10. Given a graph G with vertex set $v_1, \ldots, v_n$, let G' be the graph generated from G by Mycielski's construction. Let H be a subgraph of G. Let G'' be the graph obtained from G' by adding the edges $\{u_iu_j \colon v_iv_j \in E(H)\}$. Prove that $\chi(G'') = \chi(G) + 1$ and that $\omega(G'') = \max\{\omega(G), \omega(H) + 1\}$. (Pritikin)

5.2.11. (!) Prove that if G has no induced $2K_2$, then $\chi(G) \le \binom{\omega(G)+1}{2}$. (Hint: Use a maximum clique to define a collection of $\binom{\omega(G)}{2} + \omega(G)$ independent sets that cover the vertices. Comment: This is a special case of the Gyárfás–Sumner Conjecture—Remark 5.2.4) (Wagon [1980])

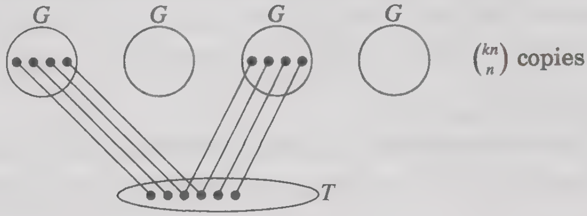
5.2.12. (!) Let $G_1 = K_1$. For $k > 1$, construct G_k as follows. To the disjoint union $G_1 + \cdots + G_{k-1}$, and add an independent set T of size $\prod_{i=1}^{k-1} n(G_i)$. For each choice of $(v_1, \ldots, v_{k-1})$ in $V(G_1) \times \cdots \times V(G_{k-1})$, let one vertex of T have neighborhood $\{v_1, \ldots, v_{k-1}\}$. (In the sketch of G_4 below, neighbors are shown for only two elements of T.)

a) Prove that $\omega(G_k) = 2$ and $\chi(G_k) = k$. (Zykov [1949])

b) Prove that G_k is k-critical. (Schäuble [1969])



5.2.13. (+) Let G be a k-chromatic graph with girth 6 and order n. Construct G' as follows. Let T be an independent set of kn new vertices. Take $\binom{kn}{n}$ pairwise disjoint copies of G, one for each way to choose an n-set $S \subset T$. Add a matching between each copy of G and its corresponding n-set S. Prove that the resulting graph has chromatic number $k+1$ and girth 6. (Comment: Since C_6 is 2-chromatic with girth 6, the process can start and these graphs exist.) (Blanche Descartes [1947, 1954])



5.2.14. *Chromatic number and cycle lengths.*

a) Let v be a vertex in a graph G. Among all spanning trees of G, let T be one that maximizes $\sum_{u \in V(G)} d_T(u, v)$. Prove that every edge of G joins vertices belonging to a path in T starting at v.

b) Prove that if $\chi(G) > k$, then G has a cycle whose length is one more than a multiple of k. (Hint: Use the tree T of part (a) to define a k-coloring of G.) (Tuza)

5.2.15. (!) Prove that a triangle-free graph with n vertices is colorable with $2\sqrt{n}$ colors. (Comment: Thus every k-chromatic triangle-free graph has at least $k^2/4$ vertices.)

5.2.16. (!) Prove that every n-vertex simple graph with no $r+1$-clique has at most $(1-1/r)n^2/2$ edges. (Hint: This can be proved using Turán's Theorem or by induction on r without Turán's Theorem.)

5.2.17. (!) Let G be a simple n-vertex graph with m edges.

a) Prove that $\omega(G) \ge \lceil n^2/(n^2-2m) \rceil$ and that this bound is sharp. (Hint: Use Exercise 5.2.16. Comment: This also yields $\chi(G) \ge \lceil n^2/(n^2-2m) \rceil$.) (Myers–Liu [1972])

b) Prove that $\alpha(G) \ge \lceil n/(d+1) \rceil$, where d is the average vertex degree of G. (Hint: Use part (a).) (Erdős–Gallai [1961])

5.2.18. The Turán graph $T_{n,r}$ (Example 5.2.7) is the complete r-partite graph with b partite sets of size $a+1$ and $r-b$ partite sets of size a, where $a = \lfloor n/r \rfloor$ and $b = n - ra$.

a) Prove that $e(T_{n,r}) = (1-1/r)n^2/2 - b(r-b)/(2r)$.

b) Since $e(G)$ must be an integer, part (a) implies $e(T_{n,r}) \le \lfloor (1-1/r)n^2/2 \rfloor$. Determine the smallest r such that strict inequality occurs for some n. For this value of r, determine all n such that $e(T_{n,r}) < \lfloor (1-1/r)n^2/2 \rfloor$.

5.2.19. (+) Let $a = \lfloor n/r \rfloor$. Compare the Turán graph $T_{n,r}$ with the graph $\overline{K}_a + K_{n-a}$ to prove directly that $e(T_{n,r}) = \binom{n-a}{2} + (r-1)\binom{a+1}{2}$.

5.2.20. Given positive integers n and k, let $q = \lfloor n/k \rfloor$, $r = n - qk$, $s = \lfloor n/(k+1) \rfloor$, and $t = n - s(k+1)$. Prove that $\binom{q}{2}k + rq \ge \binom{s}{2}(k+1) + ts$. (Hint: Consider the complement of the Turán graph.) (Richter [1993])

5.2.21. Prove that among the n-vertex simple graphs with no $r+1$-clique, the Turán graph $T_{n,r}$ is the *unique* graph having the maximum number of edges. (Hint: Examine the proof of Theorem 5.2.9 more carefully.)

5.2.22. A circular city with diameter four miles will get 18 cellular-phone power stations. Each station has a transmission range of six miles. Prove that no matter where in the city the stations are placed, at least two will each be able to transmit to at least five others. (Adapted from Bondy–Murty [1976, p115])

5.2.23. (!) *Turán's proof of Turán's Theorem*, including uniqueness (Turán [1941]).

a) Prove that a maximal simple graph with no $r+1$-clique has an r-clique.

b) Prove that $e(T_{n,r}) = \binom{r}{2} + (n-r)(r-1) + e(T_{n-r,r})$.

c) Use parts (a) and (b) to prove Turán's Theorem by induction on n, including the characterization of graphs acheiving the bound.

5.2.24. (+) Let $t_r(n) = e(T_{n,r})$. Let G be a graph with n vertices that has $t_r(n) - k$ edges and at least one $r+1$-clique, where $k \geq 0$. Prove that G has at least $f_r(n) + 1 - k$ cliques of order $r+1$, where $f_r(n) = n - \lceil n/r \rceil - r$. (Hint: Prove that a graph with exactly one $r+1$-clique has at most $t_r(n) - f_r(n)$ edges.) (Erdős [1964], Moon [1965c])

5.2.25. *Partial analogue of Turán's Theorem for $K_{2,m}$.*

a) Prove that if G is simple and $\sum_{v \in V(G)} \binom{d(v)}{2} > (m-1)\binom{n}{2}$, then G contains $K_{2,m}$. (Hint: View $K_{2,m}$ as two vertices with m common neighbors.)

b) Prove that $\sum_{v \in V(G)} \binom{d(v)}{2} \geq e(2e/n - 1)$, where G has e edges.

c) Use parts (a) and (b) to prove that a graph with more than $\frac{1}{2}(m-1)^{1/2}n^{3/2} + n/4$ edges contains $K_{2,m}$.

d) Application: Given n points in the plane, prove that the distance is exactly 1 for at most $\frac{1}{\sqrt{2}}n^{3/2} + n/4$ pairs. (Bondy–Murty [1976, p111–112])

5.2.26. For $n \geq 4$, prove that every n-vertex graph with more than $\frac{1}{2}n\sqrt{n-1}$ edges has girth at most 4. (Hint: Use the methods of Exercise 5.2.25

5.2.27. (+) For $n \geq 6$, prove that the maximum number of edges in a simple m-vertex graph not having two edge-disjoint cycles is $n+3$. (Pósa)

5.2.28. (+) For $n \geq 6$, prove that the maximum number of edges in a simple n-vertex graph not having two disjoint cycles is $3n-6$. (Pósa)

5.2.29. (!) Let G be a claw-free graph (no induced $K_{1,3}$).

a) Prove that the subgraph induced by the union of any two color classes in a proper coloring of G consists of paths and even cycles.

b) Prove that if G has a proper coloring using exactly k colors, then G has a proper k-coloring where the color classes differ in size by at most one. (Niessen–Kind [2000])

5.2.30. (+) Prove that if G has a proper coloring g in which every color class has at least two vertices, then G has an optimal coloring f in which every color class has at least two vertices. (Hint: If f has a color class with only one vertex, use g to make an alteration in f. The proof can be given algorithmically or by induction on $\chi(G)$.) (Gallai [1963c])

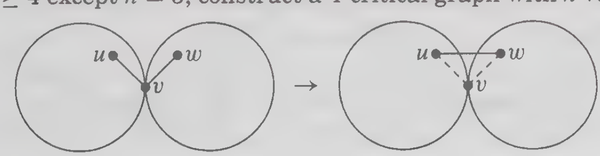
5.2.31. Let G be a connected k-chromatic graph that is not a complete graph or a cycle of length congruent to 3 modulo 6. Prove that every proper k-coloring of G has two vertices of the same color with a common neighbor. (Tomescu)

5.2.32. (!) *The Hajós construction* (Hajós [1961]).

a) Let G and H be k-critical graphs sharing only vertex v, with $vu \in E(G)$ and $vw \in E(H)$. Prove that $(G - vu) \cup (H - vw) \cup uw$ is k-critical.

b) For all $k \geq 3$, use part (a) to obtain a k-critical graph other than K_k.

c) For all $n \geq 4$ except $n = 5$, construct a 4-critical graph with n vertices.



5.2.33. Let G be a k-critical graph having a separating set $S = \{x, y\}$. By Proposition 5.2.18, $x \not\leftrightarrow y$. Prove that G has exactly two S-lobes and that they can be named G_1, G_2 such that $G_1 + xy$ is k-critical and $G_2 \cdot xy$ is k-critical (here $G_2 \cdot xy$ denotes the graph obtained from G_2 by adding xy and then contracting it).

5.2.34. (!) Let G be a 4-critical graph having a separating set S of size 4. Prove that $G[S]$ has at most four edges. (Pritikin)

5.2.35. (+) *Alternative proof that k-critical graphs are $k-1$-edge-connected.*

a) Let G be a k-critical graph, with $k \geq 3$. Prove that for every $e, f \in E(G)$ there is a $k-1$-critical subgraph of G containing e but not f. (Toft [1974])

b) Use part (a) and induction on k to prove Dirac's Theorem that every k-critical graph is $k-1$-edge-connected. (Toft [1974])

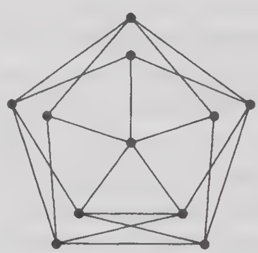
5.2.36. (+) Prove that if G is k-critical and every $k-1$-critical subgraph of G is isomorphic to K_{k-1}, then $G = K_k$ (if $k \geq 4$) (Hint: Use Toft's critical graph lemma—Exercise 5.2.35a.) (Stiebitz [1985])

5.2.37. A graph G is **vertex-color-critical** if $\chi(G - v) < \chi(G)$ for all $v \in V(G)$.

a) Prove that every color-critical graph is vertex-color-critical.

b) Prove that every 3-chromatic vertex-color-critical graph is color-critical.

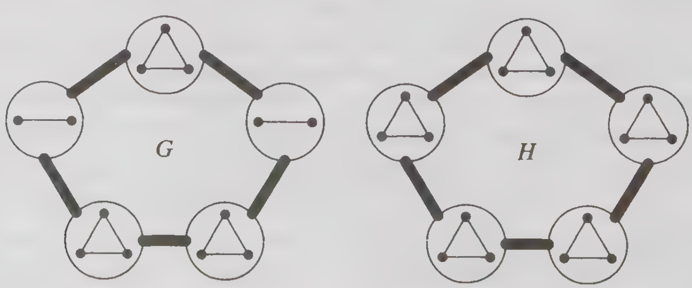
c) Prove that the graph below is vertex-color-critical but not color-critical. (Comment: This is *not* the Grötzsch graph.)



5.2.38. (!) Prove that every simple graph with minimum degree at least 3 contains a K_4-subdivision. (Hint: Prove a stronger result—every nontrivial simple graph with at most one vertex of degree less than 3 contains a K_4-subdivision. The proof of Theorem 5.2.20 already shows that every 3-connected graph contains a K_4-subdivision.) (Dirac [1952a])

5.2.39. (!) Given that $\delta(G) \geq 3$ forces a K_4-subdivision in G, prove that the maximum number of edges in a simple n-vertex graph with no K_4-subdivision is $2n - 3$.

5.2.40. Thick edges below indicate that every vertex in one circle is adjacent to every vertex in the other. Prove that $\chi(G) = 7$ but G has no K_7-subdivision. Prove that $\chi(H) = 8$ but H has no K_8-subdivision. (Catlin [1979])



5.2.41. Let $m = k(k+1)/2$. Prove that $K_{m,m-1}$ has no K_{2k}-subdivision.

5.2.42. $(+)$ Let F be a forest with m edges. Let G be a simple graph such that $\delta(G) \ge m$ and $n(G) \ge n(F)$. Prove that G contains F as a subgraph. (Hint: Delete one leaf from each nontrivial component of F to obtain F'. Let R be the set of neighbors of the deleted vertices. Map R onto an m-set $X \subseteq V(G)$ that minimizes $e(G[X])$. Extend X to a copy of F'. Use Hall's Theorem to show that X can be matched into the remaining vertices to complete a copy of F.) (Brandt [1994])

5.2.43. $(+)$ Let G be a k-chromatic graph. It follows from Lemma 5.1.18 and Proposition 2.1.8 that G contains every k-vertex tree as a subgraph. Strengthen this to a labeled analogue: if f is a proper k-coloring of G and T is a tree with vertex set $\{w_1, \dots, w_k\}$, then there is an adjacency-preserving map $\phi\colon V(T) \to V(G)$ such that $f(\phi(w_i)) = i$ for all i. (Gyárfás–Szemerédi–Tuza [1980], Sumner [1981])

5.2.44. $(+)$ Let G be a k-chromatic graph of girth at least 5. Prove that G contains every k-vertex tree as an induced subgraph. (Gyárfás–Szemerédi–Tuza [1980])

5.3. Enumerative Aspects

Sometimes we can shed light on a hard problem by considering a more general problem. No good algorithm to test existence of a proper k-coloring is known (see Appendix B), but still we can study the number of proper k-colorings (here we fix a particular set of k colors). The chromatic number $\chi(G)$ is the minimum k such that the count is positive; knowing the count for all k would tell us the chromatic number. Birkhoff [1912] introduced this counting problem as a possible way to attack the Four Color Problem (Section 6.3).

In this section, we will discuss properties of the counting function, classes where it is easy to compute, and further related topics.

COUNTING PROPER COLORINGS

We start by defining the counting problem as a function of k.

5.3.1. Definition. Given $k \in \mathbb{N}$ and a graph G, the value $\chi(G;k)$ is the number of proper colorings $f\colon V(G) \to [k]$. The set of available colors is $[k] = \{1, \dots, k\}$; the k colors need not all be used in a coloring f. Changing the names of the colors that are used produces a different coloring.

5.3.2. Example. $\chi(\overline{K}_n;k)) = k^n$ *and* $\chi(K_n;k)) = k(k-1)\cdots(k-n+1)$.

When coloring the vertices of $\overline{K}_n$, we can use any of the k colors at each vertex no matter what colors we have used at other vertices. Each of the k^n functions from the vertex set to $[k]$ is a proper coloring, and hence $\chi(\overline{K}_n;k) = k^n$.

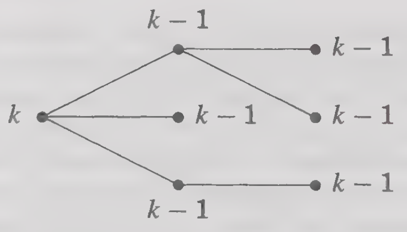
When we color the vertices of K_n, the colors chosen earlier cannot be used on the ith vertex. There remain $k-i+1$ choices for the color of the ith vertex no matter how the earlier colors were chosen. Hence $\chi(K_n;k) = k(k-1)\cdots(k-n+1)$.

We can also count this as $\binom{k}{n}n!$ by first choosing n distinct colors and then multiplying by $n!$ to count the ways to assign the chosen colors to the vertices. For example, $\chi(K_3; 3) = 6$ and $\chi(K_3; 4) = 24$.

The value of the product is 0 when $k < n$. This makes sense, since K_n has no proper k-colorings when $k < n$. ■

5.3.3. Proposition. If T is a tree with n vertices, then $\chi(T; k) = k(k-1)^{n-1}$.

Proof: Choose some vertex v of T as a root. We can color v in k ways. If we extend a proper coloring to new vertices as we grow the tree from v, at each step only the color of the parent is forbidden, and we have $k - 1$ choices for the color of the new vertex. Furthermore, deleting a leaf shows inductively that every proper k-coloring arises in this way. Hence $\chi(T; k) = k(k-1)^{n-1}$. ■



Another way to count the colorings is to observe that the color classes of each proper coloring of G partition $V(G)$ into independent sets. Grouping the colorings according to this partition leads to a formula for $\chi(G; k)$ that is a polynomial in k of degree $n(G)$. Note that this holds for the answers in Example 5.3.2 and Proposition 5.3.3. Since every graph has this property, $\chi(G; k)$ as a function of k is called the **chromatic polynomial** of G.

5.3.4. Proposition. Let $x_{(r)} = x(x-1)\cdots(x-r+1)$. If $p_r(G)$ denotes the number of partitions of $V(G)$ into r nonempty independent sets, then $\chi(G; k) = \sum_{r=1}^{n(G)} p_r(G)k_{(r)}$, which is a polynomial in k of degree $n(G)$.

Proof: When r colors are actually used in a proper coloring, the color classes partition $V(G)$ into exactly r independent sets, which can happen in $p_r(G)$ ways. When k colors are available, there are exactly $k_{(r)}$ ways to choose colors and assign them to the classes. All the proper colorings arise in this way, so the formula for $\chi(G; k)$ is correct.

Since $k_{(r)}$ is a polynomial in k and $p_r(G)$ is a constant for each r, this formula implies that $\chi(G; k)$ is a polynomial function of k. When G has n vertices, there is exactly one partition of G into n independent sets and no partition using more sets, so the leading term is k^n. ■

5.3.5. Example. Always $p_n(G) = 1$, using independent sets of size 1. Also $p_1(G) = 0$ unless G has no edges, since only for $\overline{K}_n$ is the entire vertex set an independent set.

Consider $G = C_4$. There is exactly one partition into two independent sets: opposite vertices must be in the same independent set. When $r = 3$, we put two opposite vertices together and leave the other two in sets by themselves; we can do this in two ways. Thus $p_2 = 1$, $p_3 = 2$, $p_4 = 1$.

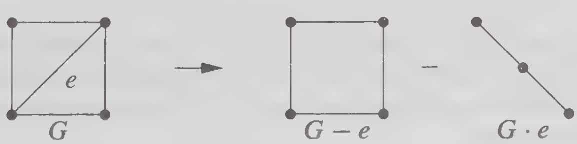
$$\begin{aligned}\chi(C_4; k) &= 1 \cdot k(k-1) + 2 \cdot k(k-1)(k-2) + 1 \cdot k(k-1)(k-2)(k-3) \\ &= k(k-1)(k^2 - 3k + 3).\end{aligned}$$ ■

Computing the chromatic polynomial in this way is not generally feasible, since there are too many partitions to consider. There is a recursive computation much like that used in Proposition 2.2.8 to count spanning trees. Again $G \cdot e$ denotes the graph obtained by contracting the edge e in G (Definition 2.2.7). Since the number of proper k-colorings is unaffected by multiple edges, **we discard multiple copies of edges that arise from the contraction**, keeping only one copy of each to form a simple graph.

5.3.6. Theorem. (Chromatic recurrence) If G is a simple graph and $e \in E(G)$, then $\chi(G; k) = \chi(G - e; k) - \chi(G \cdot e; k)$.

Proof: Every proper k-coloring of G is a proper k-coloring of $G - e$. A proper k-coloring of $G - e$ is a proper k-coloring of G if and only if it gives distinct colors to the endpoints u, v of e. Hence we can count the proper k-colorings of G by subtracting from $\chi(G - e; k)$ the number of proper k-colorings of $G - e$ that give u and v the same color.

Colorings of $G - e$ in which u and v have the same color correspond directly to proper k-colorings of $G \cdot e$, in which the color of the contracted vertex is the common color of u and v. Such a coloring properly colors all the edges of $G \cdot e$ if and only if it properly colors all the edges of G other than e. ■



5.3.7. Example. *Proper k-colorings of C_4.* Deleting an edge of C_4 produces P_4, while contracting an edge produces K_3. Since P_4 is a tree and K_3 is a complete graph, we have $\chi(P_4; k) = k(k-1)^3$ and $\chi(K_3; k) = k(k-1)(k-2)$. Using the chromatic recurrence, we obtain

$$\chi(C_4; k) = \chi(P_4; k) - \chi(K_3; k) = k(k-1)(k^2 - 3k + 3).$$ ■

Because both $G - e$ and $G \cdot e$ have fewer edges than G, we can use the chromatic recurrence inductively to compute $\chi(G; k)$. We need initial conditions for graphs with no edges, which we have already computed: $\chi(\overline{K}_n; k) = k^n$.

5.3.8. Theorem. (Whitney [1933c]) The chromatic polynomial $\chi(G;k)$ has degree $n(G)$, with integer coefficients alternating in sign and beginning $1, -e(G), \cdots$.

Proof: We use induction on $e(G)$. The claims hold trivially when $e(G) = 0$, where $\chi(\overline{K}_n;k) = k^n$. For the induction step, let G be an n-vertex graph with $e(G) \geq 1$. Each of $G-e$ and $G\cdot e$ has fewer edges than G, and $G \cdot e$ has $n-1$ vertices. By the induction hypothesis, there are nonnegative integers $\{a_i\}$ and $\{b_i\}$ such that $\chi(G-e;k) = \sum_{i=0}^{n}(-1)^i a_i k^{n-i}$ and $\chi(G\cdot e;k) = \sum_{i=0}^{n-1}(-1)^i b_i k^{n-1-i}$. By the chromatic recurrence,

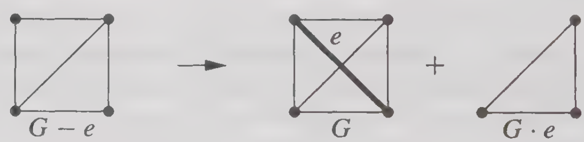
$$\begin{array}{rl}
\chi(G-e;k): & k^n - [e(G)-1]k^{n-1} + a_2k^{n-2} - \cdots + (-1)^i a_ik^{n-i} \cdots \\
-\ \chi(G\cdot e;k): & -(\qquad\qquad k^{n-1} - b_1k^{n-2} + \cdots + (-1)^{i-1} b_{i-1}k^{n-i} \cdots) \\
\hline
=\ \chi(G;k): & k^n - e(G)k^{n-1} + (a_2+b_1)k^{n-2} - \cdots + (-1)^i(a_i+b_{i-1})k^{n-i} \cdots
\end{array}$$

Hence $\chi(G;k)$ is a polynomial with leading coefficient $a_0 = 1$ and next coefficient $-(a_1+b_0) = -e(G)$, and its coefficients alternate in sign. ■

5.3.9. Example. When adding an edge yields a graph whose chromatic polynomial is easy to compute, we can use the chromatic recurrence in a different way. Instead of $\chi(G;k) = \chi(G-e;k) - \chi(G\cdot e;k)$, we can write $\chi(G-e;k) = \chi(G;k) + \chi(G\cdot e;k)$. Thus we may be able to compute $\chi(G-e;k)$ using $\chi(G;k)$.

To compute $\chi(K_n - e;k)$, for example, we let G be K_n in this alternative formula and obtain

$$\chi(K_n - e;k) = \chi(K_n;k) + \chi(K_{n-1};k) = (k-n+2)^2\prod_{i=0}^{n-3}(k-i). \qquad ■$$



We close our general discussion of $\chi(G;k)$ with an explicit formula. It has exponentially many terms, so its uses are primarily theoretical. The formula summarizes what happens if we iterate the chromatic recurrence until we dispose of all the edges.

5.3.10. Theorem. (Whitney [1932b]) Let $c(G)$ denote the number of components of a graph G. Given a set $S \subseteq E(G)$ of edges in G, let $G(S)$ denote the spanning subgraph of G with edge set S. Then the number $\chi(G;k)$ of proper k-colorings of G is given by

$$\chi(G;k) = \textstyle\sum_{S\subseteq E(G)}(-1)^{|S|}k^{c(G(S))}.$$

Proof: In applying the chromatic recurrence, contraction may produce multiple edges. We have observed that dropping these does not affect $\chi(G;k)$. We claim that deleting extra copies of edges also does not change the claimed formula.

Let e and e' be edges in G with the same endpoints. When $e' \in S$ and $e \notin S$, we have $c(G(S \cup \{e\})) = c(G(S))$, since both endpoints of e are in the same component of $G(S)$. However, $|S \cup \{e\}| = |S| + 1$. Thus the terms for S and $S \cup \{e\}$ in the sum cancel. Therefore, omitting all terms for sets of edges containing e' does not change the sum. This implies that we can keep or drop e' from the graph without changing the formula.

When computing the chromatic recurrence, we therefore obtain the same result if we do not discard multiple edges or loops and instead retain all edges for contraction or deletion. Iterating the recurrence now yields $2^{e(G)}$ terms as we dispose of all the edges; each in turn is deleted or contracted.

When all edges have been deleted or contracted, the graph that remains consists of isolated vertices. Let S be the set of edges that were contracted. The remaining vertices correspond to the components of $G(S)$; each such component becomes one vertex when the edges of S are contracted and the other edges are deleted. The $c(G(S))$ isolated vertices at the end yield a term with $k^{c(G(S))}$ colorings. Furthermore, the sign of the contribution changes for each contracted edge, so the contribution is positive if and only if $|S|$ is even.

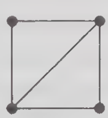
Thus the contribution when S is the set of contracted edges is $(-1)^{|S|}k^{c(G(S))}$, and this accounts for all terms in the sum. ■

5.3.11. Example. *A chromatic polynomial.* When G is a simple graph with n vertices, every spanning subgraph with 0, 1, or 2 edges has n, $n-1$, or $n-2$ components, respectively. When $|S| = 3$, the number of components is $n-2$ if and only if the three edges form a triangle; otherwise it is $n-3$.

For example, when G is a kite (four vertices, five edges) there are ten sets of three edges. For two of these, $G(S)$ consists of a triangle plus one isolated vertex. The other eight sets of three edges yield spanning subgraphs with one component. Both types of triples are counting negatively, since $|S| = 3$. All spanning subgraphs with four or five edges have only one component. Hence Theorem 5.3.10 yields

$$\chi(G;k) = k^4 - 5k^3 + 10k^2 - (2k^2 + 8k^1) + 5k - k = k^4 - 5k^3 + 8k^2 - 4k.$$

This agrees with $\chi(G;k) = k(k-1)(k-2)(k-2)$, computed by counting colorings directly or by using $\chi(G;k) = \chi(C_4;k) - \chi(P_3;k)$. ■



Whitney proved Theorem 5.3.10 using the inclusion-exclusion principle of elementary counting. Among the universe of k-colorings, the proper colorings are those not assigning the same color to the endpoints of any edge. Letting A_i be the set of k-colorings assigning the same color to the endpoints of edge e_i, we want to count the colorings that lie in none of $A_1, \dots, A_m$ (see Exercise 17).

CHORDAL GRAPHS

Counting colorings is easy for cliques and trees (and the kite) because each such graph arises from K_1 by successively adding a vertex joined to a clique. The chromatic polynomial of such a graph is a product of linear factors.

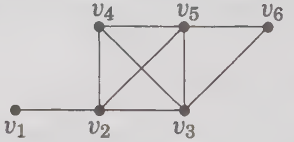
5.3.12. Definition. A vertex of G is **simplicial** if its neighborhood in G induces a clique. A **simplicial elimination ordering** is an ordering $v_n, \ldots, v_1$ for deletion of vertices so that each vertex v_i is a simplicial vertex of the remaining graph induced by $\{v_1, \ldots, v_i\}$. (These orderings are also called **perfect elimination orderings**.)

5.3.13. Example. *Chromatic polynomials from simplicial elimination orderings.* In a tree, a simplicial elimination ordering is a successive deletion of leaves. We have observed that $\chi(G; k) = k(k-1)^{n-1}$ when G is an n-vertex tree.

When $v_n, \ldots, v_1$ is a simplicial elimination ordering for G, the product rule of elementary combinatorics (Appendix A) allows us to count proper k-colorings of G. If we have colored $v_1, \ldots, v_i$, then when we add v_i there are $k - d(i)$ ways to color it, where $d(i) = |N(v_i) \cap \{v_1, \ldots, v_{i-1}\}|$. The factor $k - d(i)$ is independent of how previous color choices were made, because the neighbors of v_i that have been colored form a clique of size $d(i)$ and have distinct colors.

Deleting a simplicial vertex that starts a simplicial elimination ordering yields inductively that every proper k-coloring of G arises in this way. Thus we have expressed the chromatic polynomial as a product of linear factors.

In the graph below, $v_6, \ldots, v_1$ is a simplicial elimination ordering. When we form the graph in the order $v_1, \ldots, v_6$, the values $d(1), \ldots, d(6)$ are 0, 1, 1, 2, 3, 2, and the chromatic polynomial is $k(k-1)(k-1)(k-2)(k-3)(k-2)$. ■



5.3.14. Remark. It is important to note that some graphs without simplicial elimination orderings also have chromatic polynomials that can be expressed as a product of linear factors of the form $k - r_i$ with r_i a nonnegative integer. Exercise 19 presents an example. Thus the existence of a simplicial elimination ordering is a sufficient but not necessary condition for the chromatic polynomial to have this nice factorization property. ■

Trees, cliques, near-complete graphs ($K_n - e$), and interval graphs (Exercise 28) all have simplicial elimination orderings. When $n \geq 3$, the cycle C_n has no simplicial elimination ordering, because a cycle has no simplicial vertex to start the elimination. The existence of simplicial elimination orderings is equivalent to the absence of such cycles as induced subgraphs.

5.3.15. Definition. A **chord** of a cycle C is an edge not in C whose endpoints lie in C. A **chordless cycle** in G is a cycle of length at least 4 in G that has no chord (that is, the cycle is an induced subgraph). A graph G is **chordal** if it is simple and has no chordless cycle.

The motivation for the term "chord" is geometric. If a cycle is drawn with its vertices in order on a circle and its chords are drawn as line segments, then the chords of the cycle are chords of the circle.

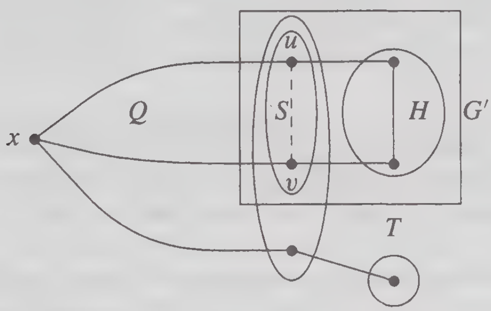
It is fairly easy to show that a graph with a simplicial elimination ordering cannot have a chordless cycle. Thus our characterization of these graphs is another TONCAS theorem. We separate the substantive part of the proof of sufficiency as a lemma that is useful on its own (see also Laskar–Shier [1983]).

5.3.16. Lemma. (Voloshin [1982], Farber–Jamison [1986]) For every vertex x in a chordal graph G, there is a simplicial vertex of G among the vertices farthest from x in G

Proof: We use induction on $n(G)$. Basis step ($n(G) = 1$): The one vertex in K_1 is simplicial.

Induction step ($n(G) \geq 2$): If x is adjacent to all other vertices, then we apply the induction hypothesis to the chordal graph $G - x$. Each simplicial vertex y of $G - x$ is also simplicial in G, since x is adjacent to all of $N(y) \cup \{y\}$.

Otherwise, let T be the set of vertices in G with maximum distance from x, and let H be a component of $G[T]$. Let S be the set of vertices in $G - T$ having neighbors in $V(H)$, and let Q be the component of $G - S$ containing x.

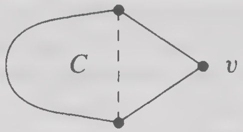


We claim that S is a clique. Each vertex of S has a neighbor in $V(H)$ and a neighbor in Q. For distinct vertices $u, v \in S$, the union of shortest u, v-paths through H and through Q is a cycle of length at least 4. Since there are no edges from $V(H)$ to $V(Q)$, this cycle has no chord other than uv. Since G has no chordless cycle, $u \leftrightarrow v$. Since $u, v \in S$ were chosen arbitrarily, S is a clique.

Now let $G' = G[S \cup V(H)]$; this omits x and thus is smaller than G. We apply the induction hypothesis to G' and a vertex $u \in S$. Since S is a clique, $S - \{u\} \subseteq N(u)$. Whether G' is a clique or not, it thus has a simplicial vertex z within $V(H)$. Since $N_G(z) \subseteq V(G')$, the vertex z is also simplicial in G, and z is a vertex with maximum distance from x, as desired. ∎

5.3.17. Theorem. (Dirac [1961]) A simple graph has a simplicial elimination ordering if and only if it is a chordal graph.

Proof: *Necessity.* Let G be a graph with a simplicial elimination ordering. Let C be a cycle in G of length at least 4. At the point when the elimination ordering first deletes a vertex of C, say v, the remaining neighbors of v form a clique. The clique includes the neighbors of v on C; the resulting edge joining them is a chord of C. Hence G has no chordless cycle.



Sufficiency. By Lemma 5.3.16, every chordal graph has a simplicial vertex. This yields a simplicial elimination ordering by induction on $n(G)$, since every induced subgraph of a chordal graph is a chordal graph. ■

Other properties of chordal graphs appear in Exercises 20–27.

A HINT OF PERFECT GRAPHS

In Proposition 5.1.16, we proved that $\chi(G) = \omega(G)$ when G is an interval graph. Furthermore, every induced subgraph of an interval graph is also an interval graph, since we can delete the interval representing v in an interval representation of G to obtain an interval representation of $G - v$. Thus $\chi(H) = \omega(H)$ holds for every induced subgraph H of an interval graph.

5.3.18. Definition. A graph G is **perfect** if $\chi(H) = \omega(H)$ for every induced subgraph $H \subseteq G$. Equivalently, $\chi(G[A]) = \omega(G[A])$ for all $A \subseteq V(G)$.

The **clique cover number** $\theta(G)$ of a graph G is the minimum number of cliques in G needed to cover $V(G)$; note that $\theta(G) = \chi(\overline{G})$.

Since cliques and independent sets exchange roles under complementation, the statement of perfection for $\overline{G}$ is "$\alpha(H) = \theta(H)$ for every induced subgraph H of G". Lovász [1972a, 1972b] proved the **Perfect Graph Theorem** (PGT): G is perfect if and only if its complement $\overline{G}$ is perfect. We prove this in Theorem 8.1.6; here we merely illustrate perfect graphs.

5.3.19. Definition. A family of graphs **G** is **hereditary** if every induced subgraph of a graph in **G** is also a graph in **G**.

5.3.20. Remark. In order to prove that every graph in a hereditary class **G** is perfect, it suffices to verify that $\chi(G) = \omega(G)$ for every $G \in$ **G**. Doing so includes the proof of equality for the induced subgraphs of G. ■

5.3.21. Example. *Bipartite graphs and their line graphs.* Bipartite graphs form a hereditary class, and $\chi(G) = \omega(G)$ for every bipartite graph; hence bipartite graphs are perfect. When H is bipartite, the statement of perfection for $\overline{H}$ is Exercise 5.1.38 and follows from $\alpha(H) = \beta'(H)$ (Corollary 3.1.24). For bipartite graphs, the nontrivial $\alpha(G) = \theta(G) = \beta'(G)$ follows at once from the trivial $\chi(G) = \omega(G)$ by the PGT.

We briefly introduced line graphs in Definition 4.2.18 to prove the edge versions of Menger's Theorem; recall that the line graph $L(G)$ has a vertex for each edge of G, with $e, f \in V(L(G))$ adjacent in $L(G)$ if they have a common endpoint in G. Line graphs of bipartite graphs form a hereditary family, since deleting a vertex in the line graph represents deleting the corresponding edge in the original graph.

Therefore, proving that $\alpha(L(G)) = \theta(L(G))$ when G is bipartite will show that complements of line graphs are perfect. A clique in $L(G)$ (when G is bipartite) consists of edges in G with a common endpoint. Thus covering the vertices of $L(G)$ with cliques corresponds to selecting vertices in G to form a vertex cover. Independent sets in $L(G)$ are matchings in G. Thus perfection for complements of line graphs of bipartite graphs amounts to the König–Egerváry Theorem ($\alpha'(G) = \beta(G)$) for matchings and vertex covers in bipartite graphs.

From this the PGT yields also $\chi(L(G)) = \omega(L(G))$. A proper coloring of $L(G)$ is a partition of $E(G)$ into matchings, and $\omega(L(G)) = \Delta(G)$ (for bipartite G). Hence $\chi(L(G)) = \omega(L(G))$ means that the edges of a bipartite graph G can be partitioned into $\Delta(G)$ matchings. In Theorem 7.1.7, we prove directly this additional result of König [1916]. ■

Since every interval graph is a chordal graph (Exercise 28), proving that all chordal graphs are perfect strengthens Proposition 5.1.16. We explore other characterizations of interval graphs and chordal graphs in Section 8.1.

5.3.22. Theorem. (Berge [1960]) Chordal graphs are perfect.

Proof: Deleting vertices cannot create chordless cycles, so the family is hereditary. By Remark 5.3.20, we need only prove $\chi(G) = \omega(G)$ when G is chordal.

In Theorem 5.3.17, we proved that G has a simplicial elimination ordering. Let $v_1, \ldots, v_n$ be the reverse of such an ordering. For each i, the neighbors of v_i among $\{v_1, \ldots, v_{i-1}\}$ form a clique.

We apply greedy coloring with this ordering. If v_i receives color k, then colors $1, \ldots, k-1$ appear on earlier neighbors of v_i. Since they form a clique, with v_i we have a clique of size k. Thus we obtain a clique whose size equals the number of colors used. ■

The argument of Theorem 5.3.22 shows that greedy coloring relative to the reverse of a simplicial elimination ordering produces an optimal coloring. This generalizes Proposition 5.1.16 about interval graphs.

We present one more fundamental class of perfect graphs; it includes all bipartite graphs.

5.3.23.* Definition. A **transitive orientation** of a graph G is an orientation D such that whenever xy and yz are edges in D, also there is an edge xz in G that is oriented from x to z in D. A simple graph G is a **comparability graph** if it has a transitive orientation.

5.3.24.* Example. If G is an X, Y-bigraph, then directing every edge from X to Y yields a transitive orientation. Thus every bipartite graph is a comparability graph. Transitive orientations arise from order relations; $x \to y$ could mean "x contains y", which is a transitive relation. ■

5.3.25.* Proposition. (Berge [1960]) Comparability graphs are perfect.

Proof: Every induced subdigraph of a transitive digraph is transitive, so the class of comparability graphs is hereditary. Thus we need only show that each comparability graph G is $\omega(G)$-colorable.

Let F be a transitive orientation of G; note that F has no cycle. As shown in proving Theorem 5.1.21, the coloring of G that assigns to each vertex v the number of vertices in the longest path of F ending at v is a proper coloring. By transitivity, the vertices of a path in F form a clique in G. Thus we have $\chi(G) \le \omega(G)$. ■

COUNTING ACYCLIC ORIENTATIONS (optional)

Surprisingly, $\chi(G; k)$ has meaning when k is a negative integer. An **acyclic orientation** of a graph is an orientation having no cycle. Setting $k = -1$ in $\chi(G; k)$ enables us to count the acyclic orientations of G.

5.3.26. Example. Since C_4 has 4 edges, it has 16 orientations. Of these, 14 are acyclic. In Example 5.3.7, we proved that $\chi(C_4; k) = k(k-1)(k^2 - 3k + 3)$. Evaluated at $k = -1$, this equals $(-1)(-2)(7) = 14$. ■

5.3.27. Theorem. (Stanley [1973]) The value of $\chi(G; k)$ at $k = -1$ is $(-1)^{n(G)}$ times the number of acyclic orientations of G.

Proof: We use induction on $e(G)$. Let $a(G)$ be the number of acyclic orientations of G. When G has no edges, $a(G) = 1$ and $\chi(G; -1) = (-1)^{n(G)}$, so the claim holds. We will prove that $a(G) = a(G - e) + a(G \cdot e)$ for $e \in E(G)$. If so, then we apply the recurrence for a, the induction hypothesis for $a(G)$ in terms of $\chi(G; k)$, and the chromatic recurrence to compute

$$a(G) = (-1)^{n(G)}\chi(G - e; -1) + (-1)^{n(G)-1}\chi(G \cdot e; -1) = (-1)^{n(G)}\chi(G; -1).$$

Now we prove the recurrence for a. Every acyclic orientation of G contains an acyclic orientation of $G - e$. An acyclic orientation D of $G - e$ may extend to 0, 1, or 2 acyclic orientations of G by orienting the edge $e = uv$. When D has no u, v-path, we can choose $v \to u$. When D has no v, u-path, we can choose $u \to v$. Since D is acyclic, D cannot have both a u, v-path and a v, u-path, so the two choices for e cannot both be forbidden.

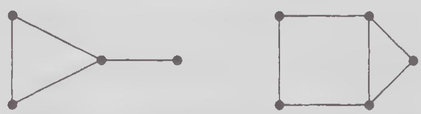
Hence every D extends in at least one way, and $a(G)$ equals $a(G-e)$ plus the number of orientations that extend in both ways. Those extending in both ways are the acyclic orientations of $G-e$ with no u,v-path *and* no v,u-path. There are exactly $a(G\cdot e)$ of these, since a u,v-path or a v,u-path in an orientation of $G-e$ becomes a cycle in $G\cdot e$. ■

The interpretation of $\chi(G;k)$ for general negative k (Exercise 32) is an instance of the phenomenon of "combinatorial reciprocity" (Stanley [1974]).

EXERCISES

Keep in mind that the notation $\chi(G;k)$ may be viewed as a polynomial or as the number of proper k-colorings of G.

5.3.1. (−) Compute the chromatic polynomials of the graphs below.



5.3.2. (−) Use the chromatic recurrence to obtain the chromatic polynomial of every tree with n vertices.

5.3.3. (−) Prove that $k^4-4k^3+3k^2$ is not a chromatic polynomial.

• • • • •

5.3.4. a) Prove that $\chi(C_n;k) = (k-1)^n + (-1)^n(k-1)$.

b) For $H = G \vee K_1$, prove that $\chi(F;k) = k\chi(G;k-1)$. From this and part (a), find the chromatic polynomial of the wheel $C_n \vee K_1$.

5.3.5. For $n \ge 1$, let $G_n = P_n \square K_2$; this is the graph with $2n$ vertices and $3n-2$ edges shown below. Prove that $\chi(G_n;k) = (k^2-3k+3)^{n-1}k(k-1)$.

5.3.6. (!) Let G be a graph with n vertices. Use Proposition 5.3.4 to give a non-inductive proof that the coefficient of k^{n-1} in $\chi(G;k)$ is $-e(G)$.

5.3.7. Prove that the chromatic polynomial of an n-vertex graph has no real root larger than $n-1$. (Hint: Use Proposition 5.3.4.)

5.3.8. (!) Prove that the number of proper k-colorings of a connected graph G is less than $k(k-1)^{n-1}$ if $k \ge 3$ and G is not a tree. What happens when $k=2$?

5.3.9. (!) Prove that $\chi(G;x+y) = \sum_{U\subseteq V(G)} \chi(G[U];x)\chi(G[\overline{U}];y)$. (Hint: Since both sides are polynomials, it suffices to prove equality when x and y are positive integers; do this by counting proper $x+y$-colorings in a different way.)

5.3.10. Let G be a connected graph with $\chi(G;k) = \sum_{i=0}^{n-1}(-1)^i a_i k^{n-i}$. For $1 \le i \le n$, prove that $a_i \ge \binom{n-1}{i}$. (Hint: Use the chromatic recurrence.)

5.3.11. (!) Prove that the sum of the coefficients of $\chi(G;k)$ is 0 unless G has no edges. (Hint: When a function is a polynomial, how can one obtain the sum of the coefficients?)

5.3.12. (+) *Coefficients of* $\chi(G;k)$.

a) Prove that the last nonzero term in the chromatic polynomial of G is the term whose exponent is the number of components of G.

b) Use part (a) to prove that if $p(k) = k^n - ak^{n-1} + \cdots \pm ck^r$ and $a > \binom{n-r+1}{2}$, then p is not a chromatic polynomial. (For example, this immediately implies that the polynomial in Exercise 5.3.3 is not a chromatic polynomial.)

5.3.13. Let G and H be graphs, possibly overlapping.

a) Prove that $\chi(G \cup H;k) = \frac{\chi(G;k)\chi(H;k)}{\chi(G\cap H;k)}$ when $G \cap H$ is a complete graph.

b) Consider two paths whose union is a cycle to show that the formula may fail when $G \cap H$ is not a complete graph.

c) Apply part (a) to conclude that the chromatic number of a graph is the maximum of the chromatic numbers of its blocks.

5.3.14. (!) Let P be the Petersen graph. By Brooks' Theorem, the Petersen graph is 3-colorable, and hence by the pigeonhole principle it has an independent set S of size 4.

a) Prove that $P - S = 3K_2$.

b) Using part (a) and symmetry, determine the number of vertex partitions of P into three independent sets.

c) In general, how can the number of partitions into the minimum number of independent sets be obtained from the chromatic polynomial of G?

5.3.15. Prove that a graph with chromatic number k has at most k^{n-k} vertex partitions into k independent sets, with equality achieved only by $K_k + (n-k)K_1$ (a k-clique plus $n-k$ isolated vertices). (Hint: Use induction on n and consider the deletion of a single vertex.) (Tomescu [1971])

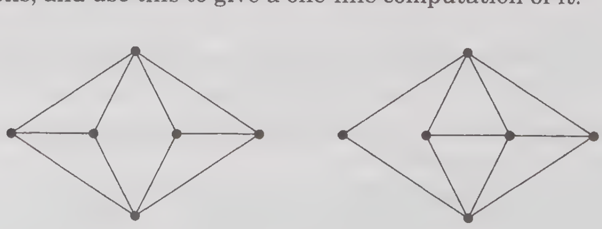
5.3.16. Let G be a simple graph with n vertices and m edges. Prove that G has at most $\frac{1}{3}\binom{m}{2}$ triangles. Conclude that the coefficient of k^{n-2} in $\chi(G;k)$ is positive, unless G has at most one edge. (Hint: Use Theorem 5.3.10.)

5.3.17. (*) Use the inclusion-exclusion principle to prove Theorem 5.3.10 directly.

5.3.18. (!) Consider the chromatic polynomials of the graphs below.

a) Without computing them, give a short proof that they are equal.

b) Express this chromatic polynomial as the sum of the chromatic polynomials of two chordal graphs, and use this to give a one-line computation of it.



5.3.19. (−) Let G be the graph obtained from K_6 by subdividing one edge. Use the chromatic recurrence to Compute $\chi(G;k)$ as a product of linear factors (factors of the form $k - c_i$). Show that G is not a chordal graph. (Read [1975], Dmitriev [1980])

5.3.20. Let G be a chordal graph. Use a simplicial elimination ordering of G to prove the following statements.

a) G has at most n maximal cliques, with equality if and only if G has no edges. (Fulkerson–Gross [1965])

b) Every maximal clique of G containing no simplicial vertex of G is a separating set.

5.3.21. The **Szekeres–Wilf number** of a graph G is $1 + \max_{H \subseteq G} \delta(H)$. Prove that a graph G is chordal if and only if in every induced subgraph the Szekeres–Wilf number equals the clique number. (Voloshin [1982])

5.3.22. Let $k_r(G)$ be the number of r-cliques in a connected chordal graph G. Prove that $\sum_{r \ge 1}(-1)^{r-1}k_r(G) = 1$. (Hint: Use induction on $n(G)$. Note that the binomial formula (Appendix A) implies that $\sum_{j \ge 0}(-1)^j \binom{m}{j} = 0$ when $m \in \mathbb{N}$.)

5.3.23. Let S be the vertex set of a cycle in a chordal graph G. Prove that G has a cycle whose vertex set consists of all but one element of S. (Comment: When G has a spanning cycle and $S \subset V(G)$, Hendry conjectured that G also has a cycle whose vertex set consists of S plus one vertex.) (Hendry [1990])

5.3.24. Let e be a edge of a cycle C in a chordal graph. Prove that e forms a triangle with a third vertex of C.

5.3.25. Let Q be a maximal clique in a chordal graph G. Prove that if $G - Q$ is connected, then Q contains a simplicial vertex. (Voloshin–Gorgos [1982])

5.3.26. Exercise 5.3.13 establishes the formula $\chi(G \cup H; k) = \frac{\chi(G;k)\chi(H;k)}{\chi(G \cap H;k)}$ when $G \cap H$ is a complete graph.

a) Prove that the formula holds when $G \cup H$ is a chordal graph regardless of whether $G \cap H$ is a complete graph.

b) Prove that if x is a vertex in a chordal graph G, then

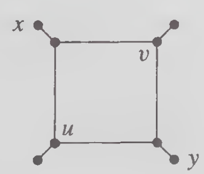
$$\chi(G; k) = \chi(G - x; k)k\frac{\chi(G[N(x)]; k-1)}{\chi(G[N(x)]; k)}.$$

(Comment: Part (b) allows the chromatic polynomial of a chordal graph to be computed via an arbitrary elimination ordering. For example, eliminating the central vertex of P_5 yields $\chi(P_5; k) = [k(k-1)]^2 k\frac{(k-1)^2}{k^2} = k(k-1)^4$.) (Voloshin [1982])

5.3.27. (+) A **minimal vertex separator** in a graph G is a set $S \subseteq V(G)$ that for some pair x, y is a minimal set whose deletion separates x and y. Every minimal separating set is a minimal vertex separator, but u, v below show that the converse need not hold.

a) Prove that if every minimal vertex separator in G is a clique, then the same property holds in every induced subgraph of G.

b) Prove that a graph G is chordal if and only if every minimal vertex separator is a clique. (Dirac [1961])



5.3.28. (!) Let G be an interval graph. Prove that G is a chordal graph and that $\overline{G}$ is a comparability graph.

5.3.29. Determine the smallest imperfect graph G such that $\chi(G) = \omega(G)$.

5.3.30. An edge in an acyclic orientation of G is **dependent** if reversing it yields a cycle.

a) Prove that every acyclic orientation of a connected n-vertex graph has at least $n-1$ independent edges.

b) Prove that if $\chi(G)$ is less than the girth of G, then G has an orientation with no dependent edges. (Hint: Use the technique in the proof of Theorem 5.1.21.)

5.3.31. (∗) The number $a(G)$ of acyclic orientations of G satisfies the recurrence $a(G) = a(G-e) + a(G \cdot e)$ (Theorem 5.3.27). The number of spanning trees of G appears to satisfy the same recurrence; does the number of acyclic orientations of G always equal the number of spanning trees? Why or why not?

5.3.32. (∗) Let D be an acyclic orientation of G, and let f be a coloring of $V(G)$ from the set $[k]$. We say that (D, f) is a **compatible pair** if $u \to v$ in D implies $f(u) \le f(v)$. Let $\eta(G;k)$ be the number of compatible pairs. Prove that $\eta(G;k) = (-1)^{n(G)}\chi(G;-k)$. (Stanley [1973])

Chapter 6

Planar Graphs

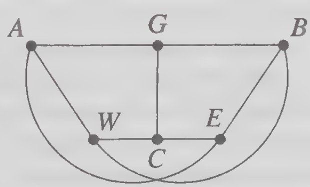
6.1. Embeddings and Euler's Formula

Topological graph theory, broadly conceived, is the study of graph layouts. Initial motivation involved the famous Four Color Problem: can the regions of every map on a globe be colored with four colors so that regions sharing a nontrivial boundary have different colors? Later motivation involves circuit layouts on silicon chips. Wire crossings cause problems in layouts, so we ask which circuits have layouts without crossings.

DRAWINGS IN THE PLANE

The following brain teaser appeared as early as Dudeney [1917].

6.1.1. Example. *Gas–water–electricity.* Three sworn enemies A, B, C live in houses in the woods. We must cut paths so that each has a path to each of three utilities, which by tradition are gas, water, and electricity. In order to avoid confrontations, we don't want any of the paths to cross. Can this be done? This asks whether $K_{3,3}$ can be drawn in the plane without edge crossings; we will give two proofs that it cannot. ■

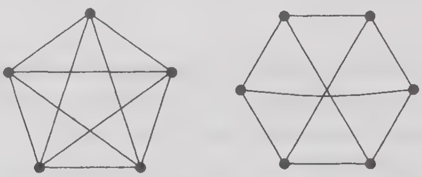


Arguments about drawings of graphs in the plane are based on the fact that every closed curve in the plane separates the plane into two regions (the inside and the outside). In elementary graph theory, we take this as an intuitive notion, but the full details in topology are quite difficult. Before discussing a way to make the arguments precise for graph theory, we show informally how this result is used to prove impossibility for planar drawings.

6.1.2. Proposition. K_5 and $K_{3,3}$ cannot be drawn without crossings.

Proof: Consider a drawing of K_5 or $K_{3,3}$ in the plane. Let C be a spanning cycle. If the drawing does not have crossing edges, then C is drawn as a closed curve. Chords of C must be drawn inside or outside this curve. Two chords conflict if their endpoints on C occur in alternating order. When two chords conflict, we can draw only one inside C and one outside C.

A 6-cycle in $K_{3,3}$ has three pairwise conflicting chords. We can put at most one inside and one outside, so it is not possible to complete the embedding. When C is a 5-cycle in K_5, at most two chords can go inside or outside. Since there are five chords, again it is not possible to complete the embeddings. Hence neither of these graphs is planar. ∎



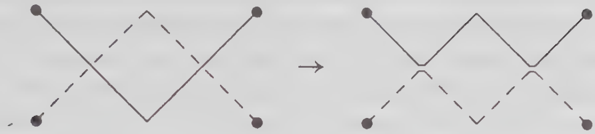
We need a precise notion of "drawing". We have used curves for edges. Using only curves formed from line segments avoids topological difficulties. These can approximate any curve well enough that the eye cannot tell the difference.

6.1.3. Definition. A **curve** is the image of a continuous map from $[0, 1]$ to $\mathbb{R}^2$. A **polygonal curve** is a curve composed of finitely many line segments. It is a **polygonal u, v-curve** when it starts at u and ends at v.

A **drawing** of a graph G is a function f defined on $V(G) \cup E(G)$ that assigns each vertex v a point $f(v)$ in the plane and assigns each edge with endpoints u, v a polygonal $f(u), f(v)$-curve. The images of vertices are distinct. A point in $f(e) \cap f(e')$ that is not a common endpoint is a **crossing**.

It is common to use the same name for a graph G and a particular drawing of G, referring to the points and curves in the drawing as the vertices and edges of G. Since the endpoint relation between the points and curves is the same as the incidence relation between the vertices and edges, the drawing can be viewed as a member of the isomorphism class containing G.

By moving edges slightly, we can ensure that no three edges have a common internal point, that an edge contains no vertex except its endpoints, and that no two edges are tangent. If two edges cross more than once, then modifying them as shown below reduces the number of crossings; thus we also require that edges cross at most once. We consider only drawings with these properties.



6.1.4. Definition. A graph is **planar** if it has a drawing without crossings. Such a drawing is a **planar embedding** of G. A **plane graph** is a particular planar embedding of a planar graph.

A curve is **closed** if its first and last points are the same. It is **simple** if it has no repeated points except possibly first=last.

A planar embedding of a graph cuts the plane into pieces. These pieces are fundamental objects of study.

6.1.5. Definition. An **open set** in the plane is a set $U \subseteq \mathbb{R}^2$ such that for every $p \in U$, all points within some small distance from p belong to U. A **region** is an open set U that contains a polygonal u, v-curve for every pair $u, v \in U$. The **faces** of a plane graph are the maximal regions of the plane that contain no point used in the embedding.

A finite plane graph G has one unbounded face (also called the **outer face**). The faces are pairwise disjoint. Points $p, q \in \mathbb{R}^2$ lying in no edge of G are in the same face if and only if there is a polygonal p, q-curve that crosses no edge.

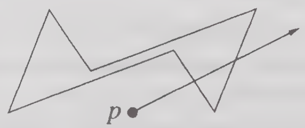
In a plane graph, every cycle is embedded as a simple closed curve. Some faces lie inside it, some outside. This again relies on the fact that a simple closed curve cuts the plane into two regions. As we have suggested, this is not too difficult for polygonal curves. We present some detail of this case in order to explain how to compute whether a point is in the inside or the outside. This proof appears in Tverberg [1980].

6.1.6.* Theorem. (Restricted Jordan Curve Theorem) A simple closed polygonal curve C consisting of finitely many segments partitions the plane into exactly two faces, each having C as boundary.

Proof: Because the list of segments is finite, nonintersecting segments cannot be arbitrarily close. Hence we can leave a face only by crossing C. As we follow C, the nearby points on our right are in a single face, and similarly for the points on the left. (There is a precise algebraic definition for "left" and "right" here.) If $x \notin C$ and $y \in C$, the segment xy first intersects C somewhere, approaching it from the right or the left. Hence every point not along C lies in the same face with at least one of the two sets we have described.

To prove that the points on the left and right lie in different faces, we consider rays in the plane. A ray emanating from a point p is "bad" if it contains an endpoint of a segment of C. Since C has finitely many segments, there are finitely many bad rays from p.

Since the list of segments is finite, each good ray from p crosses C finitely often. As the direction changes, the number of crossings changes only at a bad direction. Before and after such a direction, the parity of the number of crossings is the same. We say that p is an *even point* when every good ray from p crosses C an even number of times; otherwise p is an *odd point*.



Given points x and y in the same face of C, let P be a polygonal x, y-curve that avoids C. Since C has finitely many segments, the endpoints of segments on P can be adjusted slightly so that the rays along segments on P are good for their endpoints. A segment of P belongs to a ray from one end that contains the other; both points have good rays in the same direction. Since the segment does not intersect C, the two points have the same parity. Hence every two points in the same face have the same parity.

Because the endpoints of a short segment intersecting C exactly once have opposite parity, there are two distinct faces. The even points and odd points form the outside face and the inside face, respectively. ■

DUAL GRAPHS

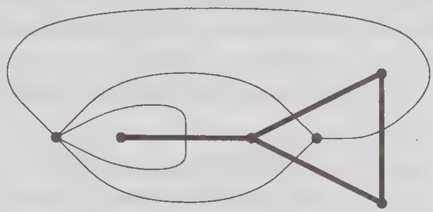
A map on the plane or the sphere can be viewed as a plane graph in which the faces are the territories, the vertices are places where boundaries meet, and the edges are the portions of the boundaries that join two vertices. We allow the full generality of loops and multiple edges. From any plane graph G, we can form a related plane graph called its "dual".

6.1.7. Definition. The **dual graph** G^* of a plane graph G is a plane graph whose vertices correspond to the faces of G. The edges of G^* correspond to the edges of G as follows: if e is an edge of G with face X on one side and face Y on the other side, then the endpoints of the dual edge $e^* \in E(G^*)$ are the vertices x, y of G^* that represent the faces X, Y of G. The order in the plane of the edges incident to $x \in V(G^*)$ is the order of the edges bounding the face X of G in a walk around its boundary.

6.1.8. Example. Every planar embedding of K_4 has four faces, and these pairwise share boundary edges. Hence the dual is another copy of K_4.

Every planar embedding of the cube Q_3 has eight vertices, 12 edges, and six faces. Opposite faces have no common boundary; the dual is a planar embedding of $K_{2,2,2}$, which has six vertices, 12 edges, and eight faces.

Taking the dual can introduce loops and multiple edges. For example, let G be the paw, drawn below in bold edges as a plane graph. Its dual graph G^* is drawn in solid edges. Since G has four vertices, four edges, and two faces, G^* has four faces, four edges, and two vertices. ■



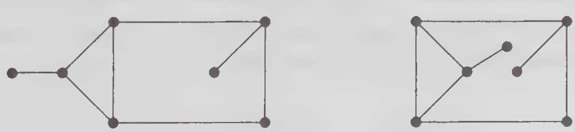
6.1.9. Remark. 1) Example 6.1.8 shows that a simple plane graph may have loops and multiple edges in its dual. A cut-edge of G becomes a loop in G^*, because the faces on both sides of it are the same. Multiple edges arise in the dual when distinct faces of G have more than one common boundary edge.

2) Some arguments require more careful geometric description of the dual. For each face X of G, we place the dual vertex x in the interior of X, so each face of G contains one vertex of G^*. For each edge e in the boundary of X, we draw a curve from x to a point on e; these do not cross. Each such curve meets another from the other side of e at the same point on e to form the edge of G^* that is dual to e. No other edges enter X. Hence G^* is a plane graph, and each edge of G^* in this layout crosses exactly one edge of G.

Such arguments lead to a proof that $(G^*)^*$ is isomorphic to G if and only if G is connected (Exercise 18). Mathematicians often use the word "dual" in a setting when performing an operation twice returns the original object. ■

6.1.10. Example. Two embeddings of a planar graph may have nonisomorphic duals. Each embedding shown below has three faces, so in each case the dual has three vertices. In the embedding on the right, the dual vertex corresponding to the outside face has degree 4. In the embedding on the left, no dual vertex has degree 4, so the duals are not isomorphic.

This does not happen with 3-connected graphs. Every 3-connected planar graph has essentially one embedding (see Exercise 8.2.45). ■



When a plane graph is connected, the boundary of each face is a closed walk. When the graph is not connected, there are faces whose boundary consists of more than one closed walk.

6.1.11. Definition. The **length** of a face in a plane graph G is the total length of the closed walk(s) in G bounding the face.

6.1.12. Example. A cut-edge belongs to the boundary of only one face, and it contributes twice to its length. Each graph in Example 6.1.10 has three faces. In the embedding on the left the lengths are 3, 6, 7; on the right they are 3, 4, 9. The sum of the lengths is 16 in each case, which is twice the number of edges. ■

6.1.13. Proposition. If $l(F_i)$ denotes the length of face F_i in a plane graph G, then $2e(G) = \sum l(F_i)$.

Proof: The face lengths are the degrees of the dual vertices. Since $e(G) = e(G^*)$, the statement $2e(G) = \sum l(F_i)$ is thus the same as the degree-sum formula $2e(G^*) = \sum d_{G^*}(x)$ for G^*. (Both sums count each edge twice.) ■

Proposition 6.1.13 illustrates that statements about a connected plane graph becomes statements about the dual graph when we interchange the roles of vertices and faces. Edges incident to a vertex become edges bounding a face, and vice versa, so the roles of face lengths and vertex degrees interchange.

We can also interpret coloring of G^* in terms of G. The edges of G^* represent shared boundaries between faces of G. Hence the chromatic number of G^* equals the number of colors needed to properly color the faces of G. Since the dual of the dual of a connected plane graph is the original graph, this means that four colors suffice to properly color the regions in every planar map if and only if every planar graph has chromatic number at most four.

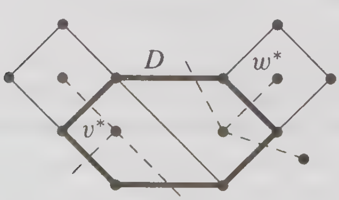
The Jordan Curve Theorem states that a simple closed curve cuts its interior from its exterior. In plane graphs, this duality between curve and cut becomes a duality between cycles and bonds.

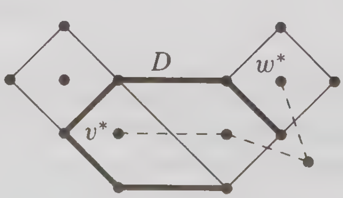
6.1.14. Theorem. Edges in a plane graph G form a cycle in G if and only if the corresponding dual edges form a bond in G^*.

Proof: Consider $D \subseteq E(G)$. Suppose first that D is the edge set of a cycle in G. The corresponding edge set $D^* \subseteq E(G^*)$ contains all dual edges joining faces inside D to faces outside D (the Jordan Curve Theorem implies that there is at least one of each). Thus D^* contains an edge cut.

If D contains a cycle and more, then D^* contains an edge cut and more. If D contains no cycle in G, then it encloses no region (see Exercise 24a). It remains possible to reach the unbounded face of G from every other without crossing D. Hence $G^* - D^*$ is connected, and D^* contains no edge cut.

Thus D^* is a minimal edge cut if and only if D is a cycle. ■

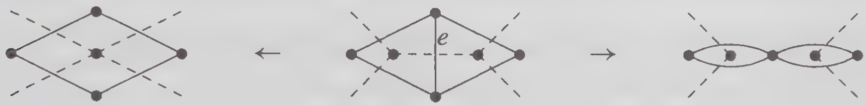




The next remark yields an inductive proof of Theorem 6.1.14 (Exercise 19).

6.1.15. Remark. Deleting a non-cut edge of G has the effect of contracting an edge in G^*, as two faces of G merge into one. Contracting a non-loop edge of G has the effect of deleting an edge in G^*. Letting G be the central solid graph below, we have $G - e$ on the left and $G \cdot e$ on the right.

Note that to maintain this duality, we keep multiple edges and loops that arise from edge contraction in plane graphs. ■



Face boundaries allow us to characterize bipartite planar graphs. The characterization can also be proved by induction (Exercise 20).

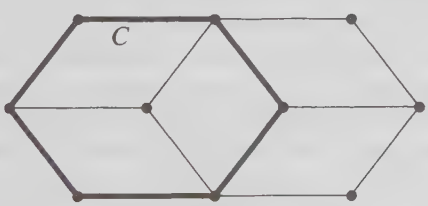
6.1.16. Theorem. The following are equivalent for a plane graph G.

A) G is bipartite.
B) Every face of G has even length.
C) The dual graph G^* is Eulerian.

Proof: A $\Rightarrow$ B. A face boundary consists of closed walks. Every odd closed walk contains an odd cycle. Therefore, in a bipartite plane graph the contributions to the length of faces are all even.

B $\Rightarrow$ A. Let C be a cycle in G. Since G has no crossings, C is laid out as a simple closed curve; let F be the region enclosed by C. Every region of G is wholly within F or wholly outside F. If we sum the face lengths for the regions inside F, we obtain an even number, since each face length is even. This sum counts each edge of C once. It also counts each edge inside F twice, since each such edge belongs twice to faces in F. Hence the parity of the length of C is the same as the parity of the full sum, which is even.

B $\Leftrightarrow$ C. The dual graph G^* is connected, and its vertex degrees are the face lengths of G. ■



Many questions we consider for general planar graphs can be answered rather easily for a special class of planar graphs.

6.1.17. Definition. A graph is **outerplanar** if it has an embedding with every vertex on the boundary of the unbounded face. An **outerplane graph** is such an embedding of an outerplanar graph.

The graph in Example 6.1.10 is outerplanar, but another embedding is needed to demonstrate this.

6.1.18. Proposition. The boundary of the outer face a 2-connected outerplane graph is a spanning cycle.

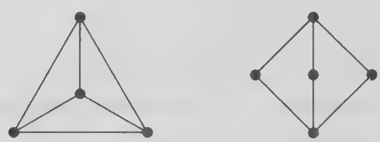
Proof: This boundary contains all the vertices. If it is not a cycle, then it passes through some vertex more than once. Such a vertex would be a cut-vertex. ■

6.1.19. Proposition. K_4 and $K_{2,3}$ are planar but not outerplanar.

Proof: The figure below shows that K_4 and $K_{2,3}$ are planar.

To show that they are not outerplanar, observe that they are 2-connected. Thus an outerplane embedding requires a spanning cycle. There is no spanning cycle in $K_{2,3}$, since it would be a cycle of length 5 in a bipartite graph.

There is a spanning cycle in K_4, but the endpoints of the remaining two edges alternate along it. Hence these chords conflict and cannot both be drawn inside. Drawing a chord outside separates a vertex from the outer face. ■



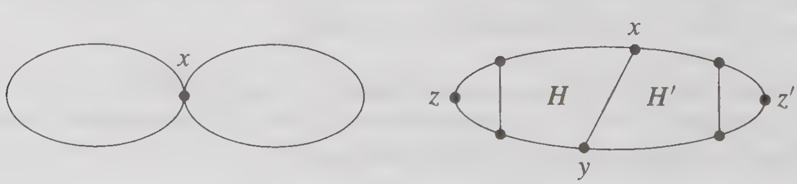
6.1.20. Proposition. Every simple outerplanar graph has a vertex of degree at most 2.

Proof: It suffices to prove the statement for connected graphs. We use induction on $n(G)$; when $n(G) \le 3$, every vertex has degree at most 2. For $n(G) \ge 4$, we prove the stronger statement that G has two nonadjacent vertices of degree at most 2.

Basis step ($n(G) = 4$): Since K_4 is not outerplanar, G has nonadjacent vertices, and two nonadjacent vertices have degree at most 2.

Induction step ($n(G) \ge 4$): If G has a cut-vertex x, then each $\{x\}$-lobe of G has a vertex of degree at most 2 other than x, and these are nonadjacent in G.

If G is 2-connected, then the outer face boundary is a cycle C. If C has no chords, then G is 2-regular. If xy is a chord of C, then the vertex sets of the two x, y-paths on C both induce outerplanar subgraphs. By the induction hypothesis, these subgraphs H, H' contain vertices z, z' of degree at most 2 that are not in $\{x, y\}$ (this includes the case where H or H' is K_3). Since no chord of C can be drawn outside C or cross xy, we have $z \not\leftrightarrow z'$. Thus z, z' is the desired pair of vertices. ■



EULER'S FORMULA

Euler's Formula ($n-e+f=2$) is the basic counting tool relating vertices, edges, and faces in planar graphs.

6.1.21. Theorem. (Euler [1758]): If a connected plane graph G has exactly n vertices, e edges, and f faces, then $n-e+f=2$.

Proof: We use induction on n. Basis step ($n=1$): G is a "bouquet" of loops, each a closed curve in the embedding. If $e=0$, then $f=1$, and the formula holds. Each added loop passes through a face and cuts it into two faces (by the Jordan Curve Theorem). This augments the edge count and the face count each by 1. Thus the formula holds when $n=1$ for any number of edges.

Induction step ($n>1$): Since G is connected, we can find an edge that is not a loop. When we contract such an edge, we obtain a plane graph G' with n' vertices, e' edges, and f' faces. The contraction does not change the number of faces (we merely shortened boundaries), but it reduces the number of edges and vertices by 1, so $n'=n-1$, $e'=e-1$, and $f'=f$. Applying the induction hypothesis yields

$$n-e+f=n'+1-(e'+1)+f'=n'-e'+f'=2. \qquad \blacksquare$$


$n=1$


$n>1$

6.1.22. Remark. 1) By Euler's Formula, all planar embeddings of a connected graph G have the same number of faces. Although the dual may depend on the embedding chosen for G, the number of vertices in the dual does not.

2) Euler's Formula as stated fails for disconnected graphs. If a plane graph G has k components, then adding $k-1$ edges to G yields a connected plane graph without changing the number of faces. Hence Euler's Formula generalizes for plane graphs with k components as $n-e+f=k+1$ (for example, consider a graph with n vertices and no edges). ■

Euler's Formula has many applications, particularly for simple plane graphs, where all faces have length at least 3.

6.1.23. Theorem. If G is a simple planar graph with at least three vertices, then $e(G) \le 3n(G)-6$. If also G is triangle-free, then $e(G) \le 2n(G)-4$.

Proof: It suffices to consider connected graphs; otherwise we could add edges. Euler's Formula will relate $n(G)$ and $e(G)$ if we can dispose of f.

Proposition 6.1.13 provides an inequality between e and f. Every face boundary in a simple graph contains at least three edges (if $n(G) \ge 3$). Letting $\{f_i\}$ be the list of face lengths, this yields $2e=\sum f_i \ge 3f$. Substituting into $n-e+f=2$ yields $e \le 3n-6$.

When G is triangle-free, the faces have length at least 4. In this case $2e = \sum f_i \geq 4f$, and we obtain $e \leq 2n - 4$. ■

6.1.24. Example. Nonplanarity of K_5 and $K_{3,3}$ follows immediately from Theorem 6.1.23. For K_5, we have $e = 10 > 9 = 3n - 6$. Since $K_{3,3}$ is triangle-free, we have $e = 9 > 8 = 2n - 4$. These graphs have too many edges to be planar. ■

6.1.25. Definition. A **maximal planar graph** is a simple planar graph that is not a spanning subgraph of another planar graph. A **triangulation** is a simple plane graph where every face boundary is a 3-cycle.

6.1.26. Proposition. For a simple n-vertex plane graph G, the following are equivalent.
A) G has $3n - 6$ edges.
B) G is a triangulation.
C) G is a maximal plane graph.

Proof: A $\Leftrightarrow$ B. For a simple n-vertex plane graph, the proof of Theorem 6.1.23 shows that having $3n - 6$ edges is equivalent to $2e = 3f$, which occurs if and only if every face is a 3-cycle.

B $\Leftrightarrow$ C. There is a face that is longer than a 3-cycle if and only if there is a way to add an edge to the drawing and obtain a larger simple plane graph. ■

6.1.27. Remark. A graph embeds in the plane if and only if it embeds on a sphere. Given an embedding on a sphere, we can puncture the sphere inside a face and project the embedding onto a plane tangent to the opposite point. This yields a planar embedding in which the punctured face on the sphere becomes the unbounded face in the plane. The process is reversible. ■

6.1.28. Application. *Regular polyhedra.* Informally, we think of a regular polyhedron as a solid whose boundary consists of regular polygons of the same length, with the same number of faces meeting at each vertex. When we expand the polyhedron out to a sphere and then lay out the drawing in the plane as in Remark 6.1.27, we obtain a regular plane graph with faces of the same length. Hence the dual also is a regular graph.

Let G be a plane graph with n vertices, e edges, and f faces. Suppose that G is regular of degree k and that all faces have length l. The degree-sum formula for G and for G^* yields $kn = 2e = lf$. By substituting for n and f in Euler's Formula, we obtain $e(\frac{2}{k} - 1 + \frac{2}{l}) = 2$. Since e and 2 are positive, the other factor must also be positive, which yields $(2/k) + (2/l) > 1$, and hence $2l + 2k > kl$. This inequality is equivalent to $(k - 2)(l - 2) < 4$.

Because the dual of a 2-regular graph is not simple, we require that $k, l \geq 3$. Now $(k - 2)(l - 2) < 4$ also requires $k, l \leq 5$. The only integer pairs satisfying these requirements for (k, l) are $(3, 3)$, $(3, 4)$, $(3, 5)$, $(4, 3)$, and $(5, 3)$.

Once we specify k and l, there is only one way to lay out the plane graph when we start with any face. Hence there are only the five Platonic solids listed below, one for each pair (k, l) that satisfying the requirements. ■

k	l	$(k-2)(l-2)$	e	n	f	name
3	3	1	6	4	4	tetrahedron
3	4	2	12	8	6	cube
4	3	2	12	6	8	octahedron
3	5	3	30	20	12	dodecahedron
5	3	3	30	12	20	icosahedron

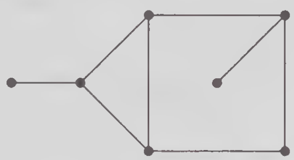
EXERCISES

6.1.1. (–) Prove or disprove:
 a) Every subgraph of a planar graph is planar.
 b) Every subgraph of a nonplanar graph is nonplanar.

6.1.2. (–) Show that the graphs formed by deleting one edge from K_5 and $K_{3,3}$ are planar.

6.1.3. (–) Determine all r, s such that $K_{r,s}$ is planar.

6.1.4. (–) Determine the number of isomorphism classes of planar graphs that can be obtained as planar duals of the graph below



6.1.5. (–) Prove that a plane graph has a cut-vertex if and only if its dual has a cut-vertex.

6.1.6. (–) Prove that a plane graph is 2-connected if and only if for every face, the bounding walk is a cycle.

6.1.7. (–) A **maximal outerplanar graph** is a simple outerplanar graph that is not a spanning subgraph of a larger simple outerplanar graph. Let G be a maximal outerplanar graph with at least three vertices. Prove that G is 2-connected.

6.1.8. (–) Prove that every simple planar graph has a vertex of degree at most 5.

6.1.9. (–) Use Theorem 6.1.23 to prove that every simple planar graph with fewer than 12 vertices has a vertex of degree at most 4.

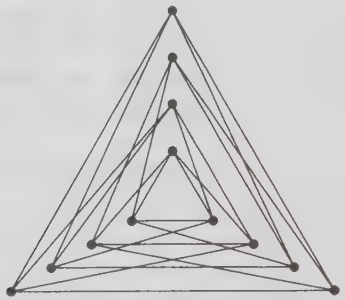
6.1.10. (–) Prove or disprove: There is no simple bipartite planar graph with minimum degree at least 4.

6.1.11. (–) Let G be a maximal planar graph. Prove that G^* is 2-edge-connected and 3-regular.

6.1.12. (–) Draw the five regular polyhedra as planar graphs. Show that the octahedron is the dual of the cube and the icosahedron is the dual of the dodecahedron.

• • • • •

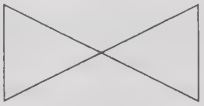
6.1.13. Find a planar embedding of the graph below.



6.1.14. Prove or disprove: For each $n \in \mathbb{N}$, there is a simple connected 4-regular planar graph with more than n vertices.

6.1.15. Construct a 3-regular planar graph of diameter 3 with 12 vertices. (Comment: T. Barcume proved that no such graph has more than 12 vertices.)

6.1.16. Let F be a figure drawn continuously in the plane without retracing any segment, ending at the start (this can be viewed as an Eulerian graph). Prove that F can be drawn without allowing the pencil point to cross what has already been drawn. For example, the figure below has two traversals; one crosses itself and the other does not.



6.1.17. Prove or disprove: If G is a 2-connected simple plane graph with minimum degree 3, then the dual graph G^* is simple.

6.1.18. Given a plane graph G, draw the dual graph G^* so that each dual edge intersects its corresponding edge in G and no other edge. Prove the following.

a) G^* is connected.

b) If G is connected, then each face of G^* contains exactly one vertex of G.

c) $(G^*)^* = G$ if and only if G is connected.

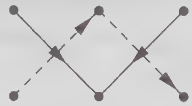
6.1.19. Let G be a plane graph. Use induction on $e(G)$ to prove Theorem 6.1.14: a set $D \subseteq E(G)$ is a cycle in G if and only if the corresponding set $D^* \subseteq E(G^*)$ is a bond in G^*. (Hint: Contract an edge of D and apply Remark 6.1.15.)

6.1.20. Prove by induction on the number of faces that a plane graph G is bipartite if and only if every face has even length.

6.1.21. (!) Prove that a set of edges in a connected plane graph G forms a spanning tree of G if and only if the duals of the remaining edges form a spanning tree of G^*.

6.1.22. The **weak dual** of a plane graph G is the graph obtained from the dual G^* by deleting the vertex for the unbounded face of G. Prove that the weak dual of an outerplane graph is a forest.

6.1.23. (!) *Directed plane graphs.* Let G be a plane graph, and let D be an orientation of G. The **dual** D^* is an orientation of G^* such that when an edge of D is traversed from tail to head, the dual edge in D^* crosses it from right to left. For example, if the solid edges below are in D, then the dashed edges are in D^*.



Prove that if D is strongly connected, then D^* has no cycle, and $\delta^-(D^*) = \delta^+(D^*) = 0$. Conclude that if D is strongly connected, then D has a face on which the edges form a clockwise cycle and another face on which the edges form a counterclockwise cycle.

6.1.24. (!) *Alternative proof of Euler's Formula.*

a) Use polygonal curves (not Euler's Formula) to prove by induction on $n(G)$ that every planar embedding of a tree G has one face.

b) Prove Euler's Formula by induction on the number of cycles.

6.1.25. (!) Prove that every n-vertex plane graph isomorphic to its dual has $2n-2$ edges. For all $n \geq 4$, construct a simple n-vertex plane graph isomorphic to its dual.

6.1.26. Determine the maximum number of edges in a simple outerplane graph with n vertices, giving three proofs.

a) By induction on n.

b) By using Euler's Formula.

c) By adding a vertex in the unbounded face and using Theorem 6.1.23.

6.1.27. Let G be a connected 3-regular plane graph in which every vertex lies on one face of length 4, one face of length 6, and one face of length 8.

a) In terms of $n(G)$, determine the number of faces of each length.

b) Use Euler's Formula and part (a) to determine the number of faces of G.

6.1.28. Let C be a closed curve bounding a convex region in the plane. Suppose that m chords of C are drawn so that no three share a point and no two share an endpoint. Let p be the number of pairs of chords that cross. In terms of m and p, compute the number of segments and the number of regions formed inside C. (Alexanderson–Wetzel [1977])

6.1.29. Prove that the complement of a simple planar graph with at least 11 vertices is nonplanar. Construct a self-complementary simple planar graph with 8 vertices.

6.1.30. (!) Let G be an n-vertex simple planar graph with girth k. Prove that G has at most $(n-2)\frac{k}{k-2}$ edges. Use this to prove that the Petersen graph is nonplanar.

6.1.31. Let G be the simple graph with vertex set $v_1, \ldots, v_n$ whose edges are $\{v_iv_j\colon |i-j| \leq 3\}$. Prove that G is a maximal planar graph.

6.1.32. Let G be a maximal planar graph. Prove that if S is a separating 3-set of G^*, then $G^* - S$ has two components. (Chappell)

6.1.33. (!) Let G be a triangulation, and let n_i be the number of vertices of degree i in G. Prove that $\sum(6-i)n_i = 12$.

6.1.34. Construct an infinite family of simple planar graphs with minimum degree 5 such that each has exactly 12 vertices of degree 5. (Hint: Modify the dodecahedron.)

6.1.35. (!) Prove that every simple planar graph with at least four vertices has at least four vertices with degree less than 6. For each even value of n with $n \geq 8$, construct an n-vertex simple planar graph G that has exactly four vertices with degree less than 6. (Grünbaum–Motzkin [1963])

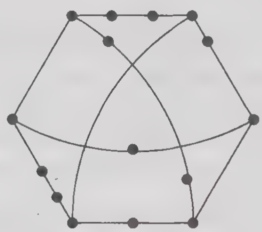
6.1.36. Let S be a set of n points in the plane such that for all $x, y \in S$, the distance in the plane between x and y is at least 1. Prove that there are at most $3n - 6$ pairs u, v in S such that the distance in the plane between u and v is exactly 1.

6.1.37. Given integers $k \geq 2$, $l \geq 1$, and kl even, construct a planar graph with exactly k faces in which every face has length l.

6.2. Characterization of Planar Graphs

Which graphs embed in the plane? We have proved that K_5 and $K_{3,3}$ do not. In fact, these are the crucial graphs and lead to a characterization of planar graphs known as Kuratowski's Theorem. Kasimir Kuratowski once asked Frank Harary about the origin of the notation for K_5 and $K_{3,3}$. Harary replied, "The K in K_5 stands for Kasimir, and the K in $K_{3,3}$ stands for Kuratowski!"

Recall that a subdivision of a graph is a graph obtained from it by replacing edges with pairwise internally-disjoint paths (Definition 5.2.19).



a subdivision of $K_{3,3}$

6.2.1. Proposition. If a graph G has a subgraph that is a subdivision of K_5 or $K_{3,3}$, then G is nonplanar.

Proof: Every subgraph of a planar graph is planar, so it suffices to show that subdivisions of K_5 and $K_{3,3}$ are nonplanar. Subdividing edges does not affect planarity; the curves in an embedding of a subdivision of G can be used to obtain an embedding of G, and vice versa. ∎

By Proposition 6.2.1, avoiding subdivisions of K_5 and $K_{3,3}$ is a necessary condition for being a planar graph. Kuratowski proved TONCAS:

6.2.2. Theorem. (Kuratowski [1930]) A graph is planar if and only if it does not contain a subdivision of K_5 or $K_{3,3}$. ∎

Kuratowski's Theorem is our goal in the first half of this section, after which we will comment on other characterizations of planar graphs.

When G is planar, we can seek a planar embedding with additional properties. Wagner [1936], Fáry [1948], and Stein [1951] showed that every finite simple planar graph has an embedding in which all edges are straight line segments; this is known as **Fáry's Theorem** (Exercise 6). For 3-connected planar graphs, we will prove the stronger property that there exists an embedding in which every face is a convex polygon.

PREPARATION FOR KURATOWSKI'S THEOREM

We introduce short names for subgraphs that demonstrate nonplanarity.

6.2.3. Definition. A **Kuratowski subgraph** of G is a subgraph of G that is a subdivision of K_5 or $K_{3,3}$. A **minimal nonplanar graph** is a nonplanar graph such that every proper subgraph is planar.

We will prove that a minimal nonplanar graph with no Kuratowski subgraph must be 3-connected. Showing that every 3-connected graph with no Kuratowski subgraph is planar then completes the proof of Kuratowski's Theorem.

6.2.4. Lemma. If F is the edge set of a face in a planar embedding of G, then G has an embedding with F being the edge set of the unbounded face.

Proof: Project the embedding onto the sphere, where the edge sets of regions remain the same and all regions are bounded, and then return to the plane by projecting from inside the face bounded by F. ■

6.2.5. Lemma. Every minimal nonplanar graph is 2-connected.

Proof: Let G be a minimal nonplanar graph. If G is disconnected, then we embed one component of G inside one face of an embedding of the rest.

If G has a cut-vertex v, let $G_1, \ldots, G_k$ be the $\{v\}$-lobes of G. By the minimality of G, each G_i is planar. By Lemma 6.2.4, we can embed each G_i with v on the outside face. We squeeze each embedding to fit in an angle smaller than $360/k$ degrees at v, after which we combine the embeddings at v to obtain an embedding of G. ■

6.2.6. Lemma. Let $S = \{x, y\}$ be a separating 2-set of G. If G is nonplanar, then adding the edge xy to some S-lobe of G yields a nonplanar graph.

Proof: Let $G_1, \ldots, G_k$ be the S-lobes of G, and let $H_i = G_i \cup xy$. If H_i is planar, then by Lemma 6.2.4 it has an embedding with xy on the outside face. For each $i > 1$, this allows H_i to be attached to an embedding of $\bigcup_{j=1}^{i-1} H_j$ by embedding H_i in a face that has xy on its boundary. Afterwards, deleting the edge xy if it is not in G yields a planar embedding of G. ■

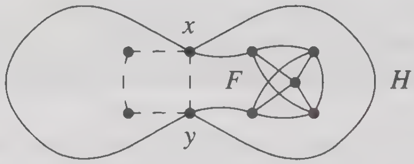
The next lemma allows us to restrict our attention to 3-connected graphs in order to prove Kuratowski's Theorem. The hypothesized graph doesn't exist, but if it did, it would be 3-connected.

6.2.7. Lemma. If G is a graph with fewest edges among all nonplanar graphs without Kuratowski subgraphs, then G is 3-connected.

Proof: Deleting an edge of G cannot create a Kuratowski subgraph in G. The hypothesis thus guarantees that deleting one edge produces a planar subgraph, and hence G is a minimal nonplanar graph. By Lemma 6.2.5, G is 2-connected.

Suppose that G has a separating 2-set $S = \{x, y\}$. Since G is nonplanar, the union of xy with some S-lobe is nonplanar (Lemma 6.2.6); let H be such a graph. Since H has fewer edges than G, the minimality of G forces H to have a Kuratowski subgraph F. All of F appears in G except possibly the edge xy.

Since S is a minimal vertex cut, both x and y have neighbors in every S-lobe. Thus we can replace xy in F with an x, y-path through another S-lobe to obtain a Kuratowski subgraph of G. This contradicts the hypothesis that G has no Kuratowski subgraph, so G has no separating 2-set. ■



CONVEX EMBEDDINGS

To complete the proof of Kuratowski's Theorem, it suffices to prove that 3-connected graphs without Kuratowski subgraphs are planar. We will use induction. In order to facilitate the proof of the induction step, it is helpful to prove a stronger statement.

6.2.8. Definition. A **convex embedding** of a graph is a planar embedding in which each face boundary is a convex polygon.

Tutte [1960, 1963] proved that every 3-connected planar graph has a convex embedding. This is best possible in terms of connectivity, since for $n \geq 4$ the 2-connected planar graph $K_{2,n}$ has no convex embedding. We follow Thomassen's approach to proving Kuratowski's Theorem by proving Tutte's stronger conclusion for 3-connected graphs without Kuratowski subgraphs. (Another proof of Tutte's result is based on ear decompositions—Kelmans [2000].)

We prove this theorem of Tutte by induction on $n(G)$. The paradigm for proving conditional statements by induction (Remark 1.3.25) tells us what lemmas we need. Our hypotheses are "3-connected" and "no Kuratowski subgraph"; our conclusion is "convex embedding". For a graph G satisfying the hypotheses, we need to find a smaller graph G' that satisfies *both* hypotheses in order to apply the induction hypothesis.

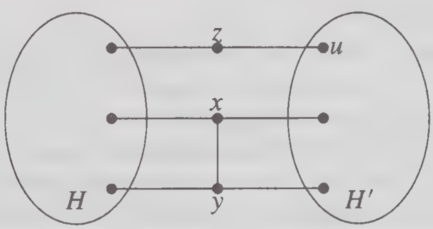
The first lemma allows us to obtain a smaller 3-connected graph G' by contracting some edge in G. The second shows that G' will also satisfy the hypothesis of having no Kuratowski subgraph. The proof will then be completed by obtaining a convex embedding of G from a convex embedding of G'.

6.2.9. Lemma. (Thomassen [1980]) Every 3-connected graph G with at least five vertices has an edge e such that $G \cdot e$ is 3-connected.

Proof: We use contradiction and extremality. Consider an edge e with endpoints x, y. If $G \cdot e$ is not 3-connected, then it has a separating 2-set S. Since G is 3-connected, S must include the vertex obtained by shrinking e. Let z denote the other vertex of S and call it the *mate* of the adjacent pair x, y. Note that $\{x, y, z\}$ is a separating 3-set in G.

Suppose that G has no edge whose contraction yields a 3-connected graph, so every adjacent pair has a mate. Among all the edges of G, choose $e = xy$ and their mate z so that the resulting disconnected graph $G - \{x, y, z\}$ has a component H with the largest order. Let H' be another component of $G - \{x, y, z\}$ (see the figure below). Since $\{x, y, z\}$ is a minimal separating set, each of x, y, z has a neighbor in each of H, H'. Let u be a neighbor of z in H', and let v be the mate of u, z.

By the definition of "mate", $G - \{z, u, v\}$ is disconnected. However, the subgraph of G induced by $V(H) \cup \{x, y\}$ is connected. Deleting v from this subgraph, if it occurs there, cannot disconnect it, since then $G - \{z, v\}$ would be disconnected. Therefore, $G_{V(H)\cup\{x,y\}} - v$ is contained in a component of $G - \{z, u, v\}$ that has more vertices than H, which contradicts the choice of x, y, z. ■



Next we need to show that edge contraction preserves the absence of Kuratowski subgraphs. We introduce a convenient term: the **branch vertices** in a subdivision H' of H are the vertices of degree at least 3 in H'.

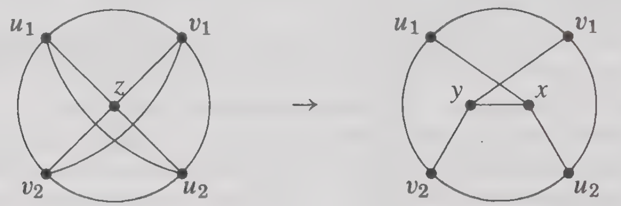
6.2.10. Lemma. If G has no Kuratowski subgraph, then also $G \cdot e$ has no Kuratowski subgraph.

Proof: We prove the contrapositive: If $G \cdot e$ contains a Kuratowski subgraph, then so does G. Let z be the vertex of $G \cdot e$ obtained by contracting $e = xy$. If z is not in H, then H itself is a Kuratowski subgraph of G. If $z \in V(H)$ but z is not a branch vertex of H, then we obtain a Kuratowski subgraph of G from H by replacing z with x or y or with the edge xy.

Similarly, if z is a branch vertex in H and at most one edge incident to z in

H is incident to x in G, then expanding z into xy lengthens that path, and y is the corresponding branch vertex for a Kuratowski subgraph in G.

In the remaining case (shown below), H is a subdivision of K_5 and z is a branch vertex, and the four edges incident to z in H consist of two incident to x and two incident to y in G. In this case, let u_1, u_2 be the branch vertices of H that are at the other ends of the paths leaving z on edges incident to x in G, and let v_1, v_2 be the branch vertices of H that are at the other ends of the paths leaving z on edges incident to y in G. By deleting the u_1, u_2-path and v_1, v_2-path from H, we obtain a subdivision of $K_{3,3}$ in G, in which y, u_1, u_2 are the branch vertices for one partite set and x, v_1, v_2 are the branch vertices of the other. ■



K_5 subdivision in G' $K_{3,3}$ subdivision in G

Now we can prove Tutte's Theorem.

6.2.11. Theorem. (Tutte [1960, 1963]) If G is a 3-connected graph with no subdivision of K_5 or $K_{3,3}$, then G has a convex embedding in the plane with no three vertices on a line.

Proof: (Thomassen [1980, 1981]) We use induction on $n(G)$.

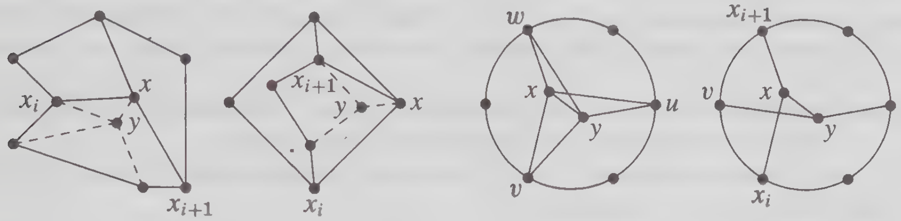
Basis step: $n(G) \leq 4$. The only 3-connected graph with at most four vertices is K_4, which has such an embedding.

Induction step: $n(G) \geq 5$. Let e be an edge such that $G \cdot e$ is 3-connected, as guaranteed by Lemma 6.2.9. Let z be the vertex obtained by contracting e. By Lemma 6.2.10, $G \cdot e$ has no Kuratowski subgraph. By the induction hypothesis, we obtain a convex embedding of $H = G \cdot e$ with no three vertices on a line.

In this embedding, the subgraph obtained by deleting the edges incident to z has a face containing z (perhaps unbounded). Since $H - z$ is 2-connected, the boundary of this face is a cycle C. All neighbors of z lie on C; they may be neighbors in G of x or y or both, where x and y are the original endpoints of e.

The convex embedding of H includes straight segments from z to all its neighbors. Let $x_1, \ldots, x_k$ be the neighbors of x in cyclic order on C. If all neighbors of y lie in the portion of C from x_i to x_{i+1}, then we obtain a convex embedding of G by putting x at z in H and putting y at a point close to z in the wedge formed by xx_i and xx_{i+1}, as shown in the diagrams for Case 0 below.

If this does not occur, then either 1) y shares three neighbors u, v, w with x, or 2) y has neighbors u, v that alternate on C with neighbors x_i, x_{i+1} of x. In Case 1, C together with xy and the edges from $\{x, y\}$ to $\{u, v, x\}$ form a subdivision of K_5. In Case 2, C together with the paths uyv, x_ixx_{i+1}, and xy form a subdivision of $K_{3,3}$. Since we are considering only graphs without Kuratowski subgraphs, in fact Case 0 must occur. ■



Case 0 Case 1 Case 2

Together, Lemma 6.2.7 and Theorem 6.2.11 imply Kuratowski's Theorem (Theorem 6.2.2). Fáry's Theorem can be obtained separately: if a graph has a planar embedding, then it has a straight-line planar embedding (Exercise 6).

For applications in computer science, we want more—a straight-line planar embedding in which the vertices are located at the integer points in a relatively small grid. Schnyder [1992] proved that every n-vertex planar graph has a straight-line embedding in which the vertices are located at integer points in the grid $[n-1] \times [n-1]$.

Many other characterizations of planar graphs have been proved; some are mentioned in the exercises. We describe two additional characterizations.

6.2.12.* Definition. A graph H is a **minor** of a graph G if a copy of H can be obtained from G by deleting and/or contracting edges of G.

For example, K_5 is a minor of the Petersen graph, although the Petersen graph does not contain a subdivision of K_5.

6.2.13.* Remark. Deletions and contractions can be performed in any order, as long as we keep track of which edge is which. Thus the minors of G can be described as "contractions of subgraphs of G".

If G contains a subdivision of H, say H', then H also is a minor of G, obtained by deleting the edges of G not in H' and then contracting edges incident to vertices of degree 2. If H has maximum degree at most 3, then H is a minor of G if and only if G contains a subdivision of H (Exercise 11).

Wagner [1937] proved that a graph G is planar if and only if neither K_5 nor $K_{3,3}$ is a minor of G. Exercise 12 obtains this from Kuratowski's Theorem. ■

6.2.14.* Remark. Some characterizations are more closely related to actual embeddings. For example, when a 3-connected graph is drawn in the plane, deleting the vertex set of a facial cycle leaves a connected subgraph.

We say that a cycle in a graph is **nonseparating** if its vertex set is not a separating set. Kelmans [1980, 1981b] proved that a subdivision of a 3-connected graph is planar if and only if every edge e lies in exactly two nonseparating cycles. Kelmans [1993] surveys related material. ■

PLANARITY TESTING (optional)

Dirac and Schuster [1954] gave the first short proof of Kuratowski's Theorem. Appearing in Harary [1969, 109–112], Bondy–Murty [1976, p153–156], and Chartrand–Lesniak [1986, p96–98], it uses special subgraphs of a graph.

6.2.15. Definition. When H is a subgraph of G, an H**-fragment** of G is either

1) an edge not in H whose endpoints are in H, or

2) a component of $G - V(H)$ together with the edges (and vertices of attachment) that connect it to H.

Together with the subgraph H itself, the H-fragments form a decomposition of G. The H-fragments are the "pieces" that must be added to an embedding of H to obtain an embedding of G. Historically, the term "H-bridge" was used; we use "H-fragment" to avoid confusion with other uses of "bridge".

An H-fragment differs from a $V(H)$-lobe because the H-fragment omits the edges of H. Also, an H-fragment may be a single edge not in H but joining vertices of H, since H need not be an induced subgraph.

For the 3-connected case of Kuratowski's Theorem, Dirac and Schuster considered a minimal nonplanar 3-connected graph G with no Kuratowski subgraph. Deleting an edge e yields a planar 2-connected graph. After choosing a cycle C through the endpoints of e, we can add e to the embedding unless there is a C-fragment embedded inside C and another embedded outside C that "conflict" with e. As in the proof of Theorem 6.2.11, this produces a Kuratowski subgraph of G. Tutte used the idea of conflicting C-fragments to obtain another characterization of planar graphs.

6.2.16. Definition. Let C be a cycle in a graph G. Two C-fragments A, B **conflict** if they have three common vertices of attachment to C or if there are four vertices v_1, v_2, v_3, v_4 in cyclic order on C such that v_1, v_3 are vertices of attachment of A and v_2, v_4 are vertices of attachment of B. The **conflict graph** of C is a graph whose vertices are the C-fragments of G, with conflicting C-fragments adjacent.

Tutte [1958] proved that G is planar if and only if the conflict graph of each cycle in G is bipartite (Exercise 13). We used this idea in our first proof that K_5 and $K_{3,3}$ are nonplanar (Proposition 6.1.2); the conflict graph of a spanning cycle in $K_{3,3}$ is C_3, and the conflict graph of a spanning cycle in K_5 is C_5.

Nonplanar 3-connected graphs have Kuratowski subgraphs of a special type. Kelmans [1984a] conjectured this extension of Kuratowski's Theorem, and it was proved independently by Kelmans [1983, 1984b] and Thomassen [1984]: Every 3-connected nonplanar graph with at least six vertices contains a cycle with three pairwise crossing chords.

Characterizations of planarity lead us to ask whether we can test quickly whether a graph is planar. There are linear-time algorithms due to Hopcroft and Tarjan [1974] and to Booth and Luecker [1976], but these are very complicated (Gould [1988, p177–185] discusses the ideas used in the Hopcroft–Tarjan algorithm). A simpler earlier algorithm is not linear but runs in polynomial time. Due to Demoucron, Malgrange, and Pertuiset [1964], it uses H-fragments.

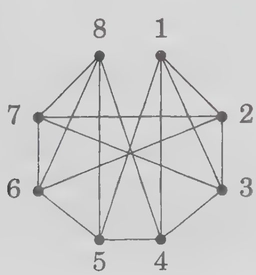
The idea is that if a planar embedding of H can be extended to a planar embedding of G, then in that extension every H-fragment of G appears inside a single face of H. We build increasingly larger plane subgraphs H of G that can be extended to an embedding of G if G is planar. We try to enlarge H by making small decisions that won't lead to trouble.

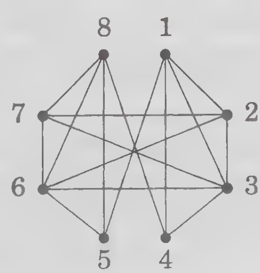
To enlarge H, we choose a face F that can accept an H-fragment B; the boundary of F must contain all vertices of attachment of B. Although we do not know the best way to embed B in F, a single path in B between vertices of attachment by itself has only one way to be added across F, so we add such a path. The details of choosing F and B appear below. Like the other algorithms mentioned, this algorithm produces an embedding if G is planar.

6.2.17. Algorithm. (Planarity Testing)
Input: A 2-connected graph. (Since G is planar if and only if each block of G is planar, and Algorithm 4.1.23 computes blocks, we may assume that G is a block with at least three vertices.)
Idea: Successively add paths from current fragments. Maintain the vertex sets forming face boundaries of the subgraph already embedded.
Initialization: G_0 is an arbitary cycle in G embedded in the plane, with two face boundaries consisting of its vertices.
Iteration: Having determined G_i, find G_{i+1} as follows.

1. Determine all G_i-fragments of the input block G.
2. For each G_i-fragment B, determine all faces of G_i that contain all vertices of attachment of B; call this set $F(B)$.
3. If $F(B)$ is empty for some B, return NONPLANAR. If $|F(B)| = 1$ for some B, select such a B. If $|F(B)| > 1$ for every B, select any B.
4. Choose a path P between two vertices of attachment of the selected B. Embed P across a face in $F(B)$. Call the resulting graph G_{i+1} and update the list of face boundaries.
5. If $G_{i+1} = G$, return PLANAR. Otherwise, augment i and return to Step 1.

6.2.18. Example. Consider the two graphs below (from Bondy–Murty [1976, p165–166]). Algorithm 6.2.17 produces a planar embedding of the graph on the left, but it terminates in Step 3 for the graph on the right. The cycle 12348765 has three pairwise crossing chords: 14, 27, 36. ■





6.2.19. Theorem. (Demoucron–Malgrange–Pertuiset [1964]) Algorithm 6.2.17 produces a planar embedding if G is planar.

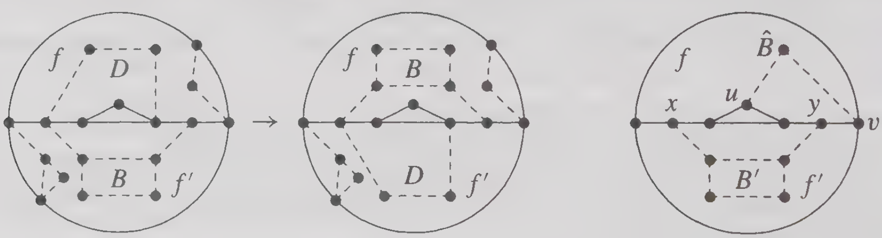
Proof: We may assume that G is 2-connected. A cycle appears as a simple closed curve in every planar embedding. Since we can reflect the plane, every embedding of a cycle in a planar graph G extends to an embedding of G.

Hence G_0 extends to a planar embedding of G if G is planar. It suffices to show that if the plane graph G_i is extendable to a planar embedding of G and the algorithm produces a plane graph G_{i+1} from G_i, then G_{i+1} also is extendable to a planar embedding of G. Note that every G_i-fragment has at least two vertices of attachment, since G is 2-connected,

If some G_i-fragment B has $|F(B)| = 1$, then there is only one face of G_i that can contain P in an extension of G_i to a planar embedding of G. The algorithm puts P in that face to obtain G_{i+1}, so in this case G_{i+1} is extendable.

Problems can arise only if $|F(B)| > 1$ for all B and we select the wrong face in which to embed a path P from the selected fragment. Suppose that (1) we embed P in face $f \in F(B)$, and (2) G_i can be extended to a planar embedding $\hat{G}$ of G in which P is inside face $f' \in F(B)$. We modify $\hat{G}$ to show that G_i can be extended to another embedding G' of G in which P is inside f. This shows that our choice causes no problem, and the constructed G_{i+1} is extendable.

Let C be the set of vertices in the boundaries of both f and f'; this includes the vertices of attachment of B. We draw G' by switching between f and f' all G_i-fragments that $\hat{G}$ places in f or f' and whose vertices of attachment lie in C. We show this on the left below, where edges of G not present in G_i are dashed.



$\hat{G}$ G'

The change switches B and produces the desired embedding G' unless some unswitched G_i-fragment $\hat{B}$ conflicts with a switched fragment. Since the switch is symmetric in f and f' and changes only their interiors, we may assume that $\hat{B}$ appears in f in $\hat{G}$. "Conflict" means that $\hat{G}$ has some B' in f', which we are trying to move to f, such that $\hat{B}$ and B' are adjacent in the conflict graph of f.

Let $\hat{A}$, A' denote the vertex sets where $\hat{B}$, B' attach to the boundary of f. Since $\hat{B}$ and B' conflict, $\hat{A}$, A' have three common vertices or four alternating vertices on the boundary of f. Since $A' \subseteq C$ but $\hat{A} \not\subseteq C$, the first possibility implies the second. Let x, u, y, v be the alternation, with $x, y \in A' \subseteq C$ and $u, v \in \hat{A}$. We may assume that $u \notin C$, as shown on the right above; if there is no such alternation, then $\hat{B}$, B' do not conflict or $\hat{B}$ can switch to f'.

Since $u \notin C$ and y is between u and v on f, no other face contains both u and v. Thus $\hat{B}$ fails to have its vertices of attachment contained in at least two faces, contradicting the hypothesis that $\left|F(\hat{B})\right| > 1$. ■

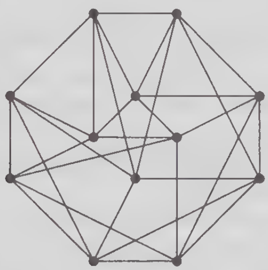
We can begin by checking that G has at most $3n - 6$ edges, maintain appropriate lists for the face boundaries, and perform the other operations via searches of linear size. Thus this algorithm runs in quadratic time. The proof of Kuratowski's Theorem by Klotz [1989] also gives a quadratic algorithm to test planarity, and it finds a Kuratowski subgraph when G is not planar.

EXERCISES

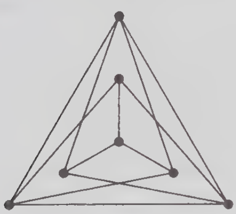
6.2.1. (−) Prove that the complement of the 3-dimensional cube Q_3 is nonplanar.

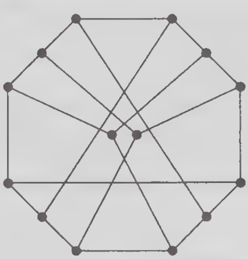
6.2.2. (−) Give three proofs that the Petersen graph is nonplanar.
a) Using Kuratowski's Theorem.
b) Using Euler's Formula and the fact that the Petersen graph has girth 5.
c) Using the planarity-testing algorithm of Demoucron–Malgrange–Pertuiset.

6.2.3. (−) Find a convex embedding in the plane for the graph below.



6.2.4. (−) For each graph below, prove nonplanarity or provide a convex embedding.





• • • • •

6.2.5. Determine the minimum number of edges that must be deleted from the Petersen graph to obtain a planar subgraph.

6.2.6. (!) *Fáry's Theorem.* Let R be a region in the plane bounded by a simple polygon with at most five sides (**simple polygon** means the edges are line segments that do not cross). Prove there is a point x inside R that "sees" all of R, meaning that the segment from x to any point of R does not cross the boundary of R. Use this to prove inductively that every simple planar graph has a straight-line embedding.

6.2.7. (!) Use Kuratowski's Theorem to prove that G is outerplanar if and only if it has no subgraph that is a subdivision of K_4 or $K_{2,3}$. (Hint: To apply Kuratowski's Theorem, find an appropriate modification of G. This is much easier than trying to mimic a proof of Kuratowski's Theorem.)

6.2.8. (!) Prove that every 3-connected graph with at least six vertices that contains a subdivision of K_5 also contains a subdivision of $K_{3,3}$. (Wagner [1937])

6.2.9. (+) For $n \geq 5$, prove that the maximum number of edges in a simple planar n-vertex graph not having two disjoint cycles is $2n - 1$. (Comment: Compare with Exercise 5.2.28.) (Markus [1999])

6.2.10. (!) Let $f(n)$ be the maximum number of edges in a simple n-vertex graph containing no $K_{3,3}$-subdivision.

a) Given that $n - 2$ is divisible by 3, construct a graph to show that $f(n) \geq 3n - 5$.

b) Prove that $f(n) = 3n - 5$ when $n - 2$ is divisible by 3 and that otherwise $f(n) = 3n - 6$. (Hint: Use induction on n, invoking Exercise 6.2.8 for the 3-connected case.) (Thomassen [1984])

(Comment: Mader [1998] proved the more difficult result that $3n - 6$ is the maximum number of edges in an n-vertex simple graph with no K_5-subdivision.)

6.2.11. (!) Let H be a graph with maximum degree at most 3. Prove that a graph G contains a subdivision of H if and only if G contains a subgraph contractible to H.

6.2.12. (!) Wagner [1937] proved that the following condition is necessary and sufficient for a graph G to be planar: neither K_5 nor $K_{3,3}$ can be obtained from G by performing deletions and contractions of edges.

a) Show that deletion and contraction of edges preserve planarity. Conclude from this that Wagner's condition is necessary.

b) Use Kuratowski's Theorem to prove that Wagner's condition is sufficient.

6.2.13. Prove that a graph G is planar if and only if for every cycle C in G, the conflict graph for C is bipartite. (Tutte [1958])

6.2.14. Let x and y be vertices of a planar graph G. Prove that G has a planar embedding with x and y on the same face unless $G - x - y$ has a cycle C with x and y in conflicting C-fragments in G. (Hint: Use Kuratowski's Theorem. Comment: Tutte proved this without Kuratowski's Theorem and used it to prove Kuratowski's Theorem.)

6.2.15. Let G be a 3-connected simple plane graph containing a cycle C. Prove that C is the boundary of a face in G if and only if G has exactly one C-fragment. (Comment: Tutte [1963] proved this to obtain Whitney's [1933b] result that 3-connected planar graphs have essentially only one planar embedding. See also Kelmans [1981a])

6.2.16. (+) Let G be an outerplanar graph with n vertices, and let P be a set of n points in the plane, no three of which lie on a line. The *extreme points* of P induce a convex polygon that contains the other points in its interior.

a) Let p_1, p_2 be consecutive extreme points of P. Prove that there is a point $p \in P - \{p_1, p_2\}$ such that 1) no point of P is inside $p_1 p_2 p$, and 2) some line l through p separates p_1 from p_2, meets P only at p, and has exactly $i - 2$ points of P on the side of l containing p_2.

b) Prove that G has a straight-line embedding with its vertices mapped onto P. (Hint: Use part (a) to prove the stronger statement that if v_1, v_2 are two consecutive vertices of the unbounded face of a maximal outerplanar graph G, and p_1, p_2 are consecutive vertices of the convex hull of P, then G can be straight-line embedded on P such that $f(v_1) = p_1$ and $f(v_2) = p_2$.) (Gritzmann–Mohar–Pach–Pollack [1989])

6.3. Parameters of Planarity

Every property and parameter we have studied for general graphs can be studied for planar graphs. The problem of greatest historical interest is the maximum chromatic number of planar graphs. We will also study parameters that measure how far a graph is from being a planar graph.

COLORING OF PLANAR GRAPHS

Because every simple n-vertex planar graph has at most $3n-6$ edges, such a graph has a vertex of degree at most 5. This yields an inductive proof that planar graphs are 6-colorable (see Exercise 2). Heawood improved the bound.

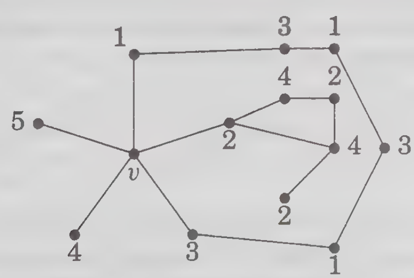
6.3.1. Theorem. (Five Color Theorem—Heawood [1890]) Every planar graph is 5-colorable.

Proof: We use induction on $n(G)$.

Basis step: $n(G) \le 5$. All such graphs are 5-colorable.

Induction step: $n(G) > 5$. The edge bound (Theorem 6.1.23) implies that G has a vertex v of degree at most 5. By the induction hypothesis, $G - v$ is 5-colorable. Let $f\colon V(G-v) \to [5]$ be a proper 5-coloring of $G-v$. If G is not 5-colorable, then f assigns each color to some neighbor of v, and hence $d(v) = 5$. Let v_1, v_2, v_3, v_4, v_5 be the neighbors of v in clockwise order around v. Name the colors so that $f(v_i) = i$.

Let $G_{i,j}$ denote the subgraph of $G - v$ induced by the vertices of colors i and j. Switching the two colors on any component of $G_{i,j}$ yields another proper 5-coloring of $G - v$. If the component of $G_{i,j}$ containing v_i does not contain v_j, then we can switch the colors on it to remove color i from $N(v)$. Now giving color i to v produces a proper 5-coloring of G. Thus G is 5-colorable unless, for each choice of i and j, the component of $G_{i,j}$ containing v_i also contains v_j. Let $P_{i,j}$ be a path in $G_{i,j}$ from v_i to v_j, illustrated below for $(i, j) = (1, 3)$.



Consider the cycle C completed with $P_{1,3}$ by v; this separates v_2 from v_4.

By the Jordan Curve Theorem, the path $P_{2,4}$ must cross C. Since G is planar, paths can cross only at shared vertices. The vertices of $P_{1,3}$ all have color 1 or 3, and the vertices of $P_{2,4}$ all have color 2 or 4, so they have no common vertex.

By this contradiction, G is 5-colorable. ∎

Every planar graph is 5-colorable, but are five colors ever needed? The history of this infamous question is discussed in Aigner [1984, 1987], Ore [1967a], Saaty–Kainen [1977, 1986], Appel–Haken [1989], and Fritsch–Fritsch [1998]. The earliest known posing of the Four Color Problem is in a letter of October 23, 1852, from Augustus de Morgan to Sir William Hamilton. The question was asked by de Morgan's student Frederick Guthrie, who later attributed it to his brother Francis Guthrie. It was phrased in terms of map coloring.

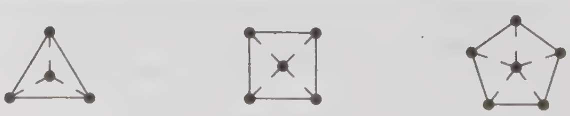
The problem's ease of statement and geometric subtleties invite fallacious proofs; some were published and remained unexposed for years. It does not suffice to forbid five pairwise-adjacent regions, since there are 5-chromatic graphs not containing K_5 (recall Mycielski's construction, for example).

Cayley announced the problem to the London Mathematical Society in 1878, and Kempe [1879] published a "solution". In 1890, Heawood published a refutation. Nevertheless, Kempe's idea of alternating paths, used by Heawood to prove the Five Color Theorem, led eventually to a proof by Appel and Haken [1976, 1977, 1986] (working with Koch). A path on which the colors alternate between two specified colors is a **Kempe chain**.

In proving the Five Color Theorem inductively, we argued that a minimal counterexample contains a vertex of degree at most 5 and that a planar graph with such a vertex cannot be a minimal counterexample. This suggests an approach to the Four Color Problem; we seek an unavoidable set of graphs that can't be present! We need only consider triangulations, since every simple planar graph is contained in a triangulation.

6.3.2. Definition. A **configuration** in a planar triangulation is a separating cycle C (the **ring**) together with the portion of the graph inside C. For the Four Color Problem, a set of configurations is **unavoidable** if a minimal counterexample must contain a member of it. A configuration is **reducible** if a planar graph containing it cannot be a minimal counterexample.

6.3.3. Example. *An unavoidable set.* We have remarked that $\delta(G) \leq 5$ for every simple planar graph. In a triangulation, every vertex has degree at least 3. Thus the set of three configurations below is unavoidable.



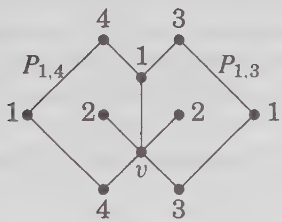
The edges from the ring to the interior are drawn with dashes because a configuration (in a triangulation) is completely determined if we state the degrees of the vertices adjacent to the ring and delete the ring (Exercise 7). Thus these configurations are written as "• 3", "• 4", and "• 5", respectively. ∎

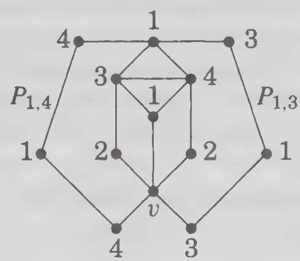
When we say that a configuration cannot be in a minimal counterexample, we mean that if it appears in a triangulation G, then it can be replaced to obtain a triangulation G' with fewer vertices such that every 4-coloring of G' can be manipulated to obtain a 4-coloring of G.

6.3.4. Remark. *Kempe's proof.* Let us try to prove the Four Color Theorem by induction using the unavoidable set $\{\bullet 3, \bullet 4, \bullet 5\}$. The approach is similar to Theorem 6.3.1. We can extend a 4-coloring of $G - v$ to complete a 4-coloring of G unless all four colors appear on $N(v)$. Thus "$\bullet$ 3" is reducible. If $d(v) = 4$, then the Kempe-chain argument works as in Theorem 6.3.1, and "$\bullet$ 4" is reducible.

Now consider "$\bullet$ 5". When $d(v) = 5$, the restriction to triangulations implies that the repeated color on $N(v)$ in the proper 4-coloring of $G - v$ appears on nonconsecutive neighbors of v. Let v_1, v_2, v_3, v_4, v_5 again be the neighbors of v in clockwise order. In the 4-coloring f of $G - v$, we may assume by symmetry that $f(v_5) = 2$ and that $f(v_i) = i$ for $1 \le i \le 4$.

Define $G_{i,j}$ and $P_{i,j}$ as in Theorem 6.3.1. We can eliminate color 1 from $N(v)$ unless the chains $P_{1,3}$ and $P_{1,4}$ exist from v_1 to v_3 and v_4, respectively, as shown on the left below. The component H of $G_{2,4}$ containing v_2 is separated from v_4 and v_5 by the cycle completed by v with $P_{1,3}$. Also, the component H' of $G_{2,3}$ containing v_5 is separated from v_2 and v_3 by the cycle completed by v with $P_{1,4}$. We can eliminate color 2 from $N(v)$ by switching colors 2 and 4 in H and colors 2 and 3 in H'. Right? This was the final case in Kempe's proof.

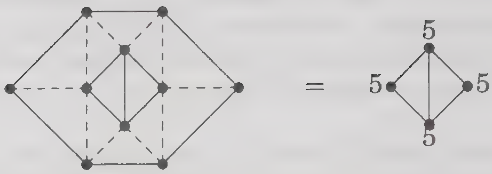




The problem is that $P_{1,3}$ and $P_{1,4}$ can intertwine, intersecting at a vertex with color 1 as shown on the right above. We can make the switch in H or in H', but making them both creates a pair of adjacent vertices with color 2. ■

Because of this difficulty, we have not shown that "$\bullet$ 5" is reducible, and we must consider larger configurations. Heesch [1969] contributed the idea of seeking configurations with small ring size instead of few vertices inside. It is not hard to show that every configuration having ring size 3 or 4 is reducible (Exercise 9). This is equivalent to showing that no minimal 5-chromatic triangulation has a separating cycle of length at most 4.

6.3.5.* Example. Birkhoff [1913] pushed the idea farther. He proved that every configuration with ring size 5 that has more than one vertex inside is reducible. He also proved that the configuration with ring size 6 below, called the **Birkhoff diamond**, is reducible.



Proving that the Birkhoff diamond is reducible takes a full page of detailed analysis. One approach is to try to show that all proper 4-colorings of the ring extend to the interior. Although some cases can be combined, and some do extend, in some cases it is necessary to use Kempe chains to show that the coloring can be changed into one that extends. ■

The intricate analysis of this first nontrivial example suggests that we have barely begun. The detail remaining is enormous. From 1913 to 1950, additional reducible configurations were found, enough to prove that all planar graphs with at most 36 vertices are 4-colorable. This was slow progress. In the 1960s, Heesch focused attention on the size of the ring, gave heuristics for finding reducible configurations, and developed methods for generating unavoidable sets.

The first proof used configurations with ring size up to 14. A ring of size 13 has 66430 distinguishable 4-colorings. Reducibility requires showing that each leads to a 4-coloring of the full graph. Kempe-chain arguments and partial collapsing of the configuration may be needed, so reducibility proofs are not easy.

Appel and Haken, working with Koch, improved upon the heuristics of Heesch and others to restrict computer searches to "promising" configurations. Using 1000 hours of computer time on three computers in 1976, they found an unavoidable set of 1936 reducible configurations, all with ring size at most 14.

6.3.6. Theorem. (Four Color Theorem—Appel–Haken–Koch [1977]) Every planar graph is 4-colorable. ■

By 1983, refinements led to an unavoidable set of 1258 reducible configurations. The proof was revisited by Robertson, Sanders, Seymour, and Thomas [1996], using the same approach. They reduced the rules used for producing unavoidable sets to a set of 32 rules. Their simplifications yielded an unavoidable set of 633 reducible configurations. They made their computer code available on the Internet; in 1997, it would prove the Four Color Theorem on a desktop workstation in about three hours.

6.3.7.* Remark. *Discharging.* To generate unavoidable sets, we replace the problem case (vertex of degree 5) by larger configurations involving a vertex of degree 5; this can be viewed as a more detailed case analysis for the hard case. Systematic rules are needed to maintain a reasonably small exhaustive set.

In a triangulation, $\sum d(v) = 2e(G) = 6n - 12$. We rewrite this as $12 = \sum(6-d(v))$ and think of $6-d(v)$ as a **charge** on vertex v. Because 12 is positive, some vertices must have positive charge (degree 5). The rules for replacing bad cases involve moving the charge around; they are called **discharging rules**. Since positive charge must remain somewhere, we obtain new unavoidable sets. The next proposition describes the effect of the simplest discharging rule. ■

6.3.8.* Proposition. Every planar triangulation with minimum degree 5 contains a configuration in the set below.

5 •—• 5   5 •—• 6

Proof: Start with charge defined by $6 - d(v)$. The first discharging rule takes the charge from each vertex of positive charge (degree 5) and distributes that charge equally among its neighbors.

A vertex of degree 5 or 6 now having positive charge must have a neighbor of degree 5. A vertex of degree 7 now having positive charge must have at least six neighbors of degree 5. Since G is a triangulation, this requires adjacent vertices of degree 5. No vertex of degree 8 or more can acquire positive charge from this discharging rule.

The total charge in the graph remains 12, so some vertex v has positive charge. For each case of $d(v)$, one of the specified configurations occurs. ■

Discharging methods are now being applied to attack other problems using computer-assisted analysis by cases.

The proof of the Four Color Theorem met with considerable uproar. Some objected in principle to the use of a computer. Others complained that the proof was too long to be verified. Others worried about computer error. A few errors were found in the original algorithms, but these were fixed (Appel–Haken [1986]). Those who have checked calculations by hand recognize that the probability of human error in a mathematical proof is much higher than the probability of computer error when the algorithm has been proved correct.

CROSSING NUMBER

In the remainder of this section, we consider parameters that measure a graph's deviation from planarity. One natural parameter is the number of planar graphs needed to form the graph; Exercises 16–20 consider this.

6.3.9. Definition. The **thickness** of a graph G is the minimum number of planar graphs in a decomposition of G into planar graphs.

6.3.10. Proposition. A simple graph G with n vertices and m edges has thickness at least $m/(3n - 6)$. If G has no triangles, then it has thickness at least $m/(2n - 4)$.

Proof: By Theorem 6.1.23, the denominator is the maximum size of each planar subgraph. The pigeonhole principle then yields the inequality. ■

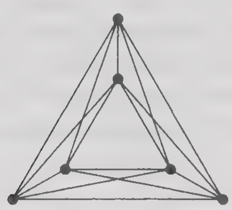
Sometimes we simply must draw a graph in the plane, even if it is not a planar graph. For example, a circuit laid out on a chip corresponds to a drawing of a graph. Since wire crossings lessen performance and cause potential problems, we try to minimize the number of crossings. We discuss the resulting parameter in the remainder of this subsection.

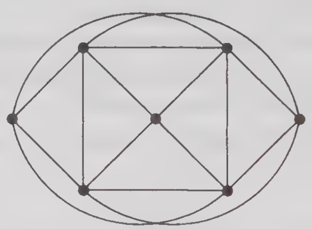
6.3.11. Definition. The **crossing number** $\nu(G)$ of a graph G is the minimum number of crossings in a drawing of G in the plane.

6.3.12. Example. $\nu(K_6) = 3$ *and* $\nu(K_{3,2,2}) = 2$. We can determine the crossing number of some small graphs by considering maximal planar subgraphs. Consider a drawing of G in the plane. If H is a maximal plane subgraph of this drawing, then every edge of G not in H crosses some edge of H, so the drawing has at least $e(G) - e(H)$ crossings. If G has n vertices, then $e(H) \le 3n - 6$. If also G has no triangles, then $e(H) \le 2n - 4$.

Since K_6 has 15 edges, and planar 6-vertex graphs have at most 12 edges, we have $\nu(K_6) \ge 3$. The drawing on the left below proves equality.

Since $K_{3,2,2}$ has 16 edges, and planar graphs with seven vertices have at most 15 edges, $\nu(K_{3,2,2}) \ge 1$. The best drawing we find has two crossings, as shown on the right below. To improve the lower bound, observe that $K_{3,2,2}$ contains $K_{3,4}$. Because $K_{3,4}$ is triangle-free, its planar subgraphs have at most $2 \cdot 7 - 4 = 10$ edges, and hence $\nu(K_{3,4}) \ge 2$. Every drawing of $K_{3,2,2}$ contains a drawing of $K_{3,4}$, so $\nu(K_{3,2,2}) \ge \nu(K_{3,4}) \ge 2$. ■





6.3.13. Proposition. Let G be an n-vertex graph with m edges. If k is the maximum number of edges in a planar subgraph of G, then $\nu(G) \ge m - k$. Furthermore, $\nu(G) \ge \frac{m^2}{2k} - \frac{m}{2}$.

Proof: Given a drawing of G in the plane, let H be a maximal subgraph of G whose edges do not cross in this drawing. Every edge not in H crosses at least one edge in H; otherwise, it could be added to H. Since H has at most k edges, we have at least $m - k$ crossings between edges of H and edges of $G - E(H)$.

After discarding $E(H)$, we have at least $m - k$ edges remaining. The same argument yields at least $(m - k) - k$ crossings in the drawing of the remaining graph. Iterating the argument yields at least $\sum_{i=1}^{t}(m - ik)$ crossings, where $t = \lfloor m/k \rfloor$. The value of the sum is $mt - kt(t + 1)/2$.

We now write $m = tk + r$, where $0 \le r \le k - 1$. We substitute $t = (m - r)/k$ in the value of the sum and simplify to obtain $\nu(G) \ge \frac{m^2}{2k} - \frac{m}{2} + \frac{r(k-r)}{2k}$. ■

The first bound $m - k$ in Proposition 6.3.13 is useful when G has few edges: the crossing number of a simple graph G is at least $e(G) - 3n + 6$, and when G is bipartite it is at least $e(G) - 2n + 4$. Iterating the argument improves the bound when $e(G)$ is larger, but for dense graphs this lower bound is weak.

Consider K_n, for example. Lacking an exact answer, we hope at least to determine the leading term in a polynomial expression for $\nu(K_n)$. To indicate a polynomial of degree k in n with leading term an^k, we often write $an^k + O(n^{k-1})$. This is consistent with the definition of "Big Oh" notation in Definition 3.2.3.

Proposition 6.3.13 yields $\nu(K_n) \geq \frac{1}{24}n^3 + O(n^2)$, but actually $\nu(K_n)$ grows like a polynomial of degree 4. The crossing number cannot exceed $\binom{n}{4}$, since we can place the vertices on the circumference of a circle and draw chords. For K_n, each set of four vertices contributes exactly one crossing. Actually, this is the worst possible straight-line drawing of K_n, since in every straight-line drawing, each set of four vertices contributes at most one crossing, depending on whether one vertex is inside the triangle formed by the other three. How many crossings can be saved by a better drawing?

6.3.14. Theorem. (R. Guy [1972]) $\frac{1}{80}n^4 + O(n^3) \leq \nu(K_n) \leq \frac{1}{64}n^4 + O(n^3)$.

Proof: A counting argument yields a recursive lower bound. A drawing of K_n with fewest crossings contains n drawings of K_{n-1}, each obtained by deleting one vertex. Each subdrawing has at least $\nu(K_{n-1}$ crossings. The total count is at least $n\nu(K_{n-1})$, but each crossing in the full drawing has been counted $(n-4)$ times. We conclude that $(n-4)\nu(K_n) \geq n\nu(K_{n-1})$.

From this inequality, we prove by induction on n that $\nu(K_n) \geq \frac{1}{5}\binom{n}{4}$ when $n \geq 5$. Basis step: $n = 5$. The crossing number of K_5 is 1. Induction step: $n > 5$. Using the induction hypothesis, we compute

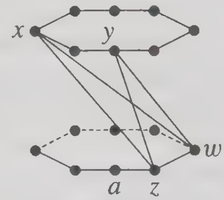
$$\nu(K_n) \geq \frac{n}{n-4}\nu(K_{n-1}) \geq \frac{n}{n-4}\frac{1}{5}\frac{(n-1)(n-2)(n-3)(n-4)}{24} = \frac{1}{5}\binom{n}{4}.$$

The denominator of the quartic term in the lower bound can be improved from 120 to 80 by considering copies of $K_{6,n-6}$, which has crossing number $6\left\lfloor\frac{n-6}{2}\right\rfloor\left\lfloor\frac{n-7}{2}\right\rfloor$ (Exercise 26b).

A better drawing lowers the upper bound from $\binom{n}{4}$ to $\frac{1}{64}n^4 + O(n^3)$. Consider $n = 2k$. Drawing K_n in the plane is equivalent to drawing it on a sphere or on the surface of a can. Place k vertices on the top rim of the can and k vertices on the bottom rim, drawing chords on the top and bottom for those k-cliques.

The edges from top to bottom fall into k natural classes. The "class number" is the circular separation between the top and bottom endpoints, ranging from $\left\lceil\frac{-k+1}{2}\right\rceil$ to $\left\lceil\frac{k-1}{2}\right\rceil$. We draw these edges to wind around the can as little as possible in passing from top to bottom, so edges in the same class don't cross. We now twist the can to make the class displacements run from 1 to k. This makes them easier to count but doesn't change the pairs of edges that cross.

Crossings on the side of the can involve two vertices on the top and two on the bottom. For top vertices x, y and bottom vertices z, w, where xz has smaller positive displacement than xw, we have a crossing for x, y, z, w if and only if the displacements to y, z, w are distinct positive values in increasing order. (For example, this holds for x, y, z, w in the illustration, but not for x, y, z, a; the edge ya winds around the can.) Hence there are $k\binom{k}{3}$ crossings on the side of the twisted can, and $\nu(K_n) \le 2\binom{k}{4} + k\binom{k}{3} = \frac{1}{64}n^4 + O(n^3)$. ■



6.3.15. Example. $\nu(K_{m,n})$. The most naive drawing puts the vertices of one partite set on one side of a channel and the vertices of the other partite set on the other side, with all edges drawn straight across. This has $\binom{n}{2}\binom{m}{2}$ crossings, but it is easy to reduce this by a factor of 4. Place the vertices of $K_{m,n}$ along two perpendicular axes. Put $\lceil n/2 \rceil$ vertices along the positive y-axis and $\lfloor n/2 \rfloor$ along the negative y-axis; similarly split the m vertices along the positive and negative x-axis. Adding up the four types of crossings generated when we join every vertex on the x-axis to every vertex on the y-axis yields $\nu(K_{m,n}) \le \left\lfloor \frac{m}{2} \right\rfloor \left\lfloor \frac{m-1}{2} \right\rfloor \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor$ (Zarankiewicz [1954]).

This bound is conjectured to be optimal (Guy [1969] tells the history). Kleitman [1970] proved it for $\min\{n, m\} \le 6$. Aided by a computer search, Woodall [1993] extended this so that the smallest unknown cases are $K_{7,11}$ and $K_{9,9}$. From Kleitman's result, Guy [1970] proved that $\nu(K_{m,n}) \ge \frac{m(m-1)}{5} \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor$, which is not far from the upper bound (Exercise 26). ■

Another general lower bound for crossing number, conjectured in Erdős-Guy [1973], has an appealing geometric application. Our proof is inductive, generalizing the lower bound argument in Theorem 6.3.14. There is an elegant probabilistic proof in Exercise 8.5.11and stronger results in Pach–Tóth [1997].

6.3.16.* Theorem. (Ajtai-Chvátal-Newborn-Szemerédi [1982], Leighton [1983]) Let G be a simple graph. If $e(G) \ge 4n(G)$, then $\nu(G) \ge \frac{1}{64}e(G)^3/n(G)^2$.

Proof: Let $m = e(G)$ and $n = n(G)$. We use induction on n.

Basis step: $m \le 5n$ (this includes all simple graphs with at most 11 vertices). Note that $(\alpha - 3) \ge \frac{1}{64}\alpha^3$ when $4 \le \alpha \le 5$. Letting $m = \alpha n$ for $4 \le \alpha \le 5$, we obtain $\nu(G) \ge m - 3n \ge \frac{1}{64}m^3/n^2$, as desired.

Induction step: $n > 11$. Given an optimal drawing of G, each crossing appears in $n-4$ of the drawings obtained by deleting a single vertex. By the induction hypothesis, $\nu(G - v) \ge \frac{1}{64}\frac{(m-d(v))^3}{(n-1)^2}$. Thus $(n-4)\nu(G) \ge \sum_{v \in V(G)} \frac{1}{64}\frac{(m-d(v))^3}{(n-1)^2}$.

By convexity, the lower bound is always at least what results when the vertex degrees are all replaced by the average degree. In other words, $\sum(m - d(v))^3 \ge n(m - 2m/n)^3$. Also $(n-1)^2(n-4) \le (n-2)^3$. Thus

$$\nu(G) \ge \frac{1}{64} n \frac{(n-2)^3 m^3}{n^3(n-1)^2(n-4)} \ge \frac{1}{64}\frac{m^3}{n^2}.$$ ■

6.3.17.* Example. *Achieving the bound.* The order of magnitude in Theorem 6.3.16 is best possible. Consider $G = \frac{n}{2m} K_{2m/n}$, where $2m$ is a multiple of n. The total number of vertices is n, and the total number of edges is asymptotic to $\frac{n^2}{2m}\frac{1}{2}(\frac{2m}{n})^2 = m$. Since $\nu(K_r) \le \frac{1}{64}r^4$, we have $\nu(G) \le \frac{n^2}{2m}\frac{1}{64}(\frac{2m}{n})^4 = \frac{1}{8}\frac{m^3}{n^2}$. This is within a constant factor of the lower bound from Theorem 6.3.16. ■

We apply Theorem 6.3.16 to a problem in combinatorial geometry. Erdős [1946] asked how many unit distances can occur among a set of n points in the plane. If the points occur in a unit grid, then the graph of unit distances is the cartesian product of two paths, and this produces about $n - O(\sqrt{n})$ edges. By taking all the points of a refined grid that lie within an appropriate distance from the origin, Erdős obtained about $n^{1+c/\log\log n}$ unit distances. This growth rate is superlinear, but it is slower than $n^{1+\epsilon}$ for each positive ϵ.

Erdős also proved an upper bound of $O(n^{3/2})$. Since two circles of radius 1 intersect in at most two points, the graph G of unit distances cannot contain $K_{2,3}$. Thus each pair of points has at most two common neighbors. Since each vertex v is a common neighbor for its $\binom{d(v)}{2}$ pairs of neighbors, $\sum\binom{d(v)}{2} \le 2\binom{n}{2}$. Since $2e(G)/n$ is the average vertex degree, convexity yields $\sum\binom{d(v)}{2} \ge n\binom{2e(G)/n}{2}$. Together, these inequalities yield the desired bound (Exercise 5.2.25 considers the edge-maximization problem in general when a biclique is forbidden).

Using number-theoretic arguments about incidences between lines and points in a point set, Spencer–Szemerédi–Trotter [1984] improved the upper bound to $O(n^{4/3})$. Székely applied Theorem 6.3.16 to give an elegant and short graph-theoretic proof of this bound.

6.3.18.* Theorem. (Spencer–Szemerédi–Trotter [1984]) There are at most $4n^{4/3}$ pairs of points at distance 1 among a set of n points in the plane.

Proof: (Székely [1997]) By moving points or pairs of points without reducing the number of pairs at distance 1, we can ensure that each point is involved in such a pair and that no two points have distance 1 only from each other. If any point now is involved in only one unit distance pair, we can rotate it around its mate until it is distance 1 from another point. This reduces the problem to the case that every point is involved in at least two such pairs.

Let P be an optimal n-point configuration, with q unit distance pairs. We obtain a graph from P, not by using the unit distance pairs as edges, but rather by drawing a unit circle around each point. If a point in P is at distance 1 from k other points in P, then these points partition the circle into k arcs. Altogether we obtain $2q$ arcs. These are the edges of a loopless graph G.

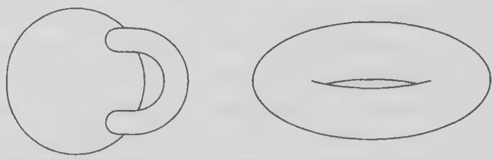
Since two points can appear on two (but not three) unit circles, G may have edges of multiplicity 2 but no larger multiplicity. We delete one copy of each duplicated edge to obtain a simple graph G' with at least q edges. We may assume that $q \ge 4n$; otherwise the bound already holds.

Because these arcs lie on n circles, they cannot produce many crossings; each pair of circles crosses at most twice. Thus our layout of G' has at most $2\binom{n}{2}$ crossings. By Theorem 6.3.16, G' has at least $\frac{1}{64}q^3/n^2$ crossings. Together, these inequalities yield $q \le 4n^{4/3}$. ■

SURFACES OF HIGHER GENUS (optional)

Instead of minimizing crossings in the plane, we could change the surface to avoid crossings. This is the effect of building overpasses and cloverleafs instead of installing traffic lights. The surface of the earth is a sphere, and for this discussion it is convenient to consider drawings on the sphere instead of in the plane. As observed in Remark 6.1.27, these settings are equivalent.

To avoid creating boundaries in the surface, we add an overpass by cutting two holes in the sphere and joining the edges of the holes by a tube. By stretching the tube and squeezing the rest of the sphere, we obtain a doughnut.



6.3.19. Definition. A **handle** is a tube joining two holes cut in a surface. The **torus** is the surface obtained by adding one handle to a sphere.

The torus is topologically the same as the sphere with one handle, in the sense that one surface can be continuously transformed into the other.†

A large graph may have many crossings and need more handles. For any graph, adding enough handles to a drawing on the sphere will eliminate all crossings and produce an embedding. When we add some number of handles, it doesn't matter how we do it, because a fundamental result of topology says that two surfaces obtained by adding the same number of handles to a sphere can be continuously deformed into each other.

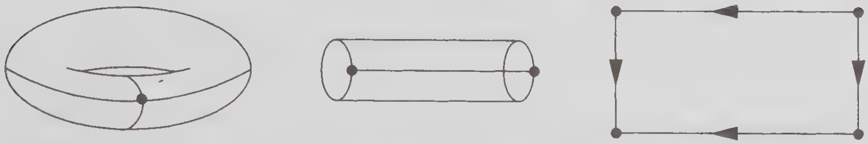
6.3.20. Definition. The **genus** of a surface obtained by adding handles to a sphere is the number of handles added; we use S_γ for the surface of genus γ. The **genus** of a graph G is the minimum γ such that G embeds on S_γ. The graphs embeddable on the surfaces of genus 0, 1, 2 are the **planar**, **toroidal**, and **double-toroidal** graphs, respectively (the surface with two handles is the **double-torus**).

The theory of planar graphs extends in some ways to graphs embeddable on higher surfaces; we discuss this only briefly, for cultural interest. Drawings of large graphs on surfaces of large genus are hard to follow, even on the **pretzel** (S_3). Locally, the surface looks like a plane sheet of paper. To draw the graph we want to lay the entire surface flat; to do this we must cut the surface. If we keep track of how the edges should be pasted back together to get the surface, we can describe the surface on a flat piece of paper. Consider first the torus.

---

†This is the source of the joke that a topologist is a person who can't tell the difference between a doughnut and a coffee cup.

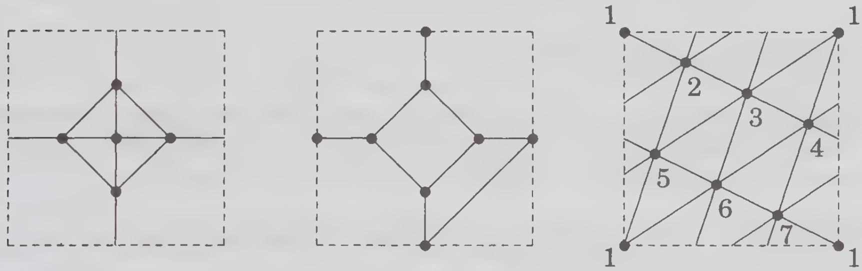
6.3.21. Example. *Combinatorial description of the torus.*



Cutting the closed tube once turns it into a cylinder, and then slitting the length of the cylinder allows us to lay it flat as a rectangle. Labeling the edges of the rectangle indicates how to paste it back together. The two sides of a cut labeled with the same letter are "identified".

Keeping track of the identifications is important because edges of an embedding on a surface may cross such a cut. When the edge reaches one border of the rectangle, it is reaching one side of the imagined cut. When it crosses the cut, it emerges from the identical point on the other copy of this border. The four "corners" of the rectangle correspond to the single point on the surface through which both cuts pass.

These ideas lead to nice toroidal embeddings of K_5, $K_{3,3}$, and K_7. ■



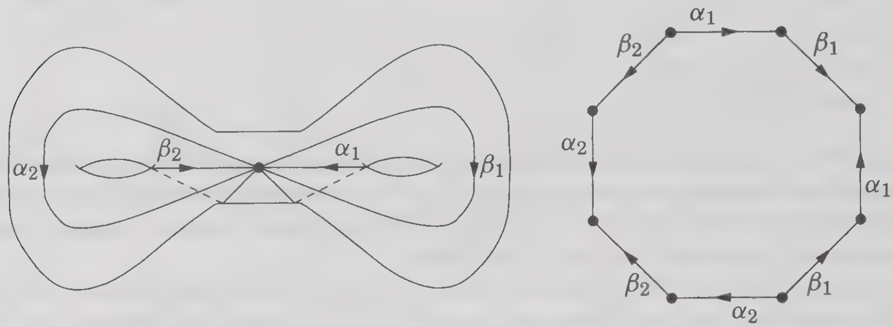
For surfaces of higher genus, there is some flexibility in making the cuts, but each way takes two cuts per handle before we can lay the surface flat. The usual representation comes from expressing the handles as "lobes" of the surface, with the cuts having a common point on the hub.

6.3.22. Example. *Laying the double torus flat.* Below is a polygonal representation for the double torus. Making the cuts is equivalent to adding loops at a single vertex until we have a one-face embedding of a bouquet of loops. In general, we make 2γ cuts through a single point to lay S_γ flat.

Keeping track of the borders from each cut leads to representing S_γ by a 4γ-gon in which a clockwise traversal of the boundary can be described by reading out the cuts as we traverse them. We record a cut using the notation of inverses when we traverse it in the opposite order.

Since we are following the boundary of a single face, with our left hand always on the wall, each edge will be followed once forward and once backward. For the example here, the traversal is $\alpha_1\beta_1\alpha_1^{-1}\beta_1^{-1}\alpha_2\beta_2\alpha_2^{-1}\beta_2^{-1}$.

Each surface S_γ has a layout of the form $\alpha_1\beta_1\alpha_1^{-1}\beta_1^{-1}\cdots\alpha_\gamma\beta_\gamma\alpha_\gamma^{-1}\beta_\gamma^{-1}$. Other layouts result from other ways of making the cuts – different ways of embedding a bouquet of 2γ loops. For example, the double torus can also be represented by an octagon with boundary $\alpha\beta\gamma\delta\alpha^{-1}\beta^{-1}\gamma^{-1}\delta^{-1}$. ■



6.3.23. Remark. *Euler's Formula for S_γ.* A **2-cell** is a region such that every closed curve in the interior can be continuously contracted to a point. A **2-cell embedding** is an embedding where every region is a 2-cell. Euler's Formula generalizes for 2-cell embeddings of connected graphs on S_γ (Exercise 35) as

$$n - e + f = 2 - 2\gamma.$$

For example, our embedding of K_7 on the torus ($\gamma = 1$) has 7 vertices, 21 edges, 14 faces, and $7 - 21 + 14 = 0$. The proof of Euler's Formula for S_γ is like the proof in the plane, except that the basis case of 1-vertex graphs needs more care. It requires showing that it takes 2γ cuts to lay the surface flat (that is, to obtain a 2-cell embedding of a graph with one vertex and one face). ■

6.3.24. Lemma. Every simple n-vertex graph embedded on S_γ has at most $3(n - 2 + 2\gamma)$ edges.

Proof: Exercise 35. ■

Note that K_7 satisfies Lemma 6.3.24 with equality on the torus ($\gamma = 1$), as every face in the toroidal embedding of K_7 is a 3-gon. Hence K_7 is a maximal toroidal graph. Rewriting $e \le 3(n-2+2\gamma)$ yields a lower bound on the number of handles we must add to obtain a surface on which G is embeddable; thus $\gamma(G) \ge 1 + (e - 3n)/6$.

Lemma 6.3.24 leads to an analogue of the Four Color Theorem for S_γ.

6.3.25. Theorem. (Heawood's Formula—Heawood [1890]) If G is embeddable on S_γ with $\gamma > 0$, then $\chi(G) \le \left\lfloor (7 + \sqrt{1 + 48\gamma})/2 \right\rfloor$.

Proof: Let $c = (7 + \sqrt{1 + 48\gamma})/2$. It suffices to prove that every simple graph embeddable on S_γ has a vertex of degree at most $c - 1$; the bound on $\chi(G)$ then follows by induction on $n(G)$. Since $\chi(G) \le c$ for all graphs with at most c vertices, so need only consider $n(G) > c$.

We use Lemma 6.3.24 to show that the average (and hence minimum) degree is at most $c - 1$. The second inequality below follows from $\gamma > 0$ and $n > c$.

Since c satisfies $c^2 - 7c + (12 - 12\gamma) = 0$, we have $c - 1 = 6 - (12 - 12\gamma)/c$, so the average degree satisfies the desired bound.

$$\frac{2e}{n} \le \frac{6(n-2+2\gamma)}{n} \le 6 - \frac{12-12\gamma}{c} = c - 1. \qquad \blacksquare$$

The key inequality here fails when $\gamma = 0$. Thus the argument is invalid for planar graphs, even though the formula reduces to $\chi(G) \le 4$ when $\gamma = 0$. Proving that the Heawood bound is sharp involves embedding K_n on S_γ with $\gamma = \lceil (n-3)(n-4)/12 \rceil$. The proof breaks into cases by the congruence class of n modulo 12 (K_7 is the first example in the easy class). Completed in Ringel–Youngs [1968], it comprises the book *Map Color Theorem* (Ringel [1974]).

Having considered the coloring problem on S_γ, one naturally wonders which graphs embed on S_γ. Planar graphs have many characterizations, beginning with Kuratowski's Theorem (Theorem 6.2.2) and Wagner's Theorem (Exercise 6.2.12). On any surface, embeddability is preserved by deleting or contracting an edge. Thus every surface has a list of "minor-minimal" obstructions to embeddability. Wagner's Theorem states that the list for the plane is $\{K_{3,3}, K_5\}$; every nonplanar graph has one of these as a minor.

More than 800 minimal forbidden minors are known for the torus. For each surface, the list is finite; this follows from the much more general statement below (the *subdivision* relation in Kuratowski's Theorem leads to infinite lists).

6.3.26. Theorem. (The Graph Minor Theorem—Robertson–Seymour [1985])
In any infinite list of graphs, some graph is a minor of another. ■

This is perhaps the most difficult theorem known in graph theory. The complete proof takes well over 500 pages (without computer assistance) in a series of 20 papers stretching beyond the year 2000. It has many ramifications about structure of graphs and complexity of computation. The techniques involved in the proof have spawned new areas of graph theory. Some aspects of these techniques and their relation to the proof of the Graph Minor Theorem are presented in the final chapter of the text by Diestel [1997].

EXERCISES

6.3.1. (−) State a polynomial-time algorithm that takes an arbitrary planar graph as input and produces a proper 5-coloring of the graph.

6.3.2. (−) A graph G is **k-degenerate** if every subgraph of G has a vertex of degree at most k. Prove that every k-degenerate graph is $k+1$-colorable.

6.3.3. (−) Use the Four Color Theorem to prove that every outerplanar graph is 3-colorable.

6.3.4. (−) Determine the crossing numbers of $K_{2,2,2,2}$, $K_{4,4}$, and the Petersen graph.

• • • • •

6.3.5. Prove that every planar graph decomposes into two bipartite graphs. (Hedetniemi [1969], Mabry [1995])

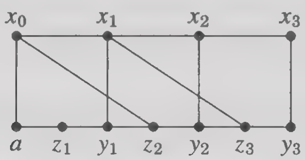
6.3.6. Without using the Four Color Theorem, prove that every planar graph with at most 12 vertices is 4-colorable. Use this to prove that every planar graph with at most 32 edges is 4-colorable.

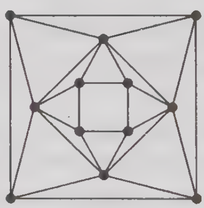
6.3.7. (!) Let H be a configuration in a planar triangulation (Definition 6.3.2). Let H' be obtained by labeling the neighbors of the ring vertices with their degrees and then deleting the ring vertices. Prove that H can be retrieved from H'.

6.3.8. Create a configuration with ring size 5 in a planar triangulation such that every internal vertex has degree at least five.

6.3.9. (+) Prove that every planar configuration having ring size at most four is reducible. (Hint: The ring is a separating cycle C. Prove that if smaller triangulations are 4-colorable, then the C-lobes of G have 4-colorings that agree on C.) (Birkhoff [1913])

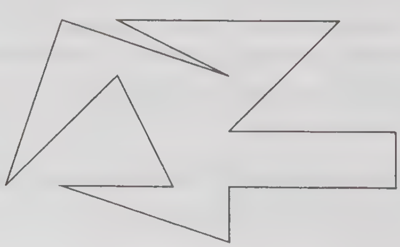
6.3.10. Grötzsch's Theorem [1959] (see Steinberg [1993], Thomassen [1994a]) states that a triangle-free planar graph G is 3-colorable. Hence $\alpha(G) \geq n(G)/3$. Tovey–Steinberg [1993] proved that $\alpha(G) > n(G)/3$ always. Prove that this is best possible by considering the family of graphs G_k defined as follows: G_1 is the 5-cycle, with vertices a, x_0, x_1, y_1, z_1 in order. For $k > 1$, G_k is obtained from G_{k-1} by adding the three vertices x_k, y_k, z_k and the five edges $x_{k-1}x_k, x_k y_k, y_k z_k, z_k y_{k-1}, z_k x_{k-2}$. The graph G_3 is shown on the left below. (Fraughnaugh [1985])





6.3.11. Define a sequence of plane graphs as follows. Let G_1 be C_4. For $n > 1$, obtain G_n from G_{n-1} by adding a new 4-cycle surrounding G_{n-1}, making each vertex of the new cycle also adjacent to two consecutive vertices of the previous outside face. The graph G_3 is shown on the right above. Prove that if n is even, then every proper 4-coloring of G_n uses each color on exactly n vertices. (Albertson)

6.3.12. (!) Without using the Four Color Theorem, prove that every outerplanar graph is 3-colorable. Apply this to prove the Art Gallery Theorem: If an art gallery is laid out as a simple polygon with n sides, then it is possible to place $\lfloor n/3 \rfloor$ guards such that every point of the interior is visible to some guard. Construct a polygon that requires $\lfloor n/3 \rfloor$ guards. (Chvátal [1975], Fisk [1978])



6.3.13. An *art gallery with walls* is a polygon plus some nonintersecting chords called "walls" that join vertices. Each interior wall has a tiny opening called a "doorway". A guard in a doorway can see everything in the two neighboring rooms, but a guard not in a doorway cannot see past a wall. Determine the minimum number t such that for every walled art gallery with n vertices, it is possible to place t guards so that every interior point is visible to some guard. (Hutchinson [1995], Kündgen [1999])

6.3.14. (+) Prove that a maximal planar graph is 3-colorable if and only if it is Eulerian. (Hint: For sufficiency, use induction on $n(G)$. Choose an appropriate pair or triple of adjacent vertices to replace with appropriate edges.) (Heawood [1898])

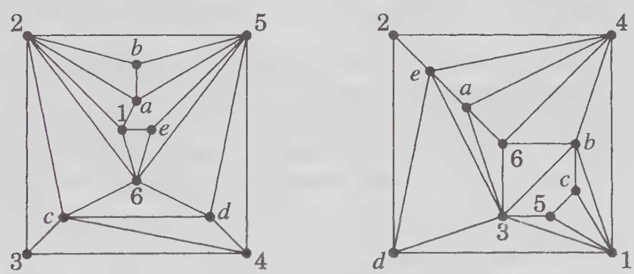
6.3.15. (!) Prove that the vertices of an outerplanar graph can be partitioned into two sets so that the subgraph induced by each set is a disjoint union of paths. (Hint: Define the partition using the parity of the distance from a fixed vertex.) (Akiyama–Era–Gervacio [1989], Goddard [1991])

6.3.16. (−) Prove that the 4-dimensional cube Q_4 is nonplanar. Decompose it into two isomorphic planar graphs; hence Q_4 has thickness 2.

6.3.17. Prove that K_n has thickness at least $\left\lfloor \frac{n+7}{6} \right\rfloor$. (Hint: $\left\lceil \frac{x}{r} \right\rceil = \left\lfloor \frac{x+r-1}{r} \right\rfloor$.) Show that equality holds for K_8 by finding a self-complementary planar graph with 8 vertices. (Comment: The thickness equals $\left\lfloor \frac{n+7}{6} \right\rfloor$ except that K_9 and K_{10} have thickness 3; Beineke–Harary [1965] for $n \not\equiv 4 \bmod 6$, and Alekseev–Gončakov [1976] for $n \equiv 4 \bmod 6$.)

6.3.18. Decompose K_9 into three pairwise-isomorphic planar graphs.

6.3.19. Prove that if G has thickness 2, then $\chi(G) \le 12$. Use the two graphs below to show that $\chi(G)$ may be as large as 9 when G has thickness 2. (Sulanke)



6.3.20. (!) When r is even and s is greater than $(r-2)^2/2$, prove that the thickness of $K_{r,s}$ is $r/2$. (Beineke–Harary–Moon [1964])

6.3.21. Determine $\nu(K_{1,2,2,2})$ and use it to compute $\nu(K_{2,2,2,2})$.

6.3.22. Prove that $K_{3,2,2}$ has no planar subgraph with 15 edges. Use this to give another proof that $\nu(K_{3,2,2}) \ge 2$.

6.3.23. Let M_n be the graph obtained from the cycle C_n by adding chords between vertices that are opposite (if n is even) or nearly opposite (if n is odd). The graph M_n is 3-regular if n is even, 4-regular if n is odd. Determine $\nu(M_n)$. (Guy–Harary [1967])

6.3.24. The graph P_n^k has vertex set $[n]$ and edge set $\{ij\colon |i-j| \le k\}$. Prove that P_n^3 is a maximal planar graph. Use a planar embedding of P_n^3 to prove that $\nu(P_n^4) = n-4$. (Harary–Kainen [1993])

6.3.25. For every positive integer k, construct a graph that embeds on the torus but requires at least k crossings when drawn in the plane. (Hint: A single easily described toroidal family suffices; use Proposition 6.3.13.)

6.3.26. (!) Use Kleitman's computation that $\nu(K_{6,n}) = 6\left\lfloor\frac{n-6}{2}\right\rfloor\left\lfloor\frac{n-7}{2}\right\rfloor$ to give counting arguments for the following lower bounds.

a) $\nu(K_{m,n}) \ge m\frac{m-1}{5}\left\lfloor\frac{n}{2}\right\rfloor\left\lfloor\frac{n-1}{2}\right\rfloor$. (Guy [1970])

b) $\nu(K_p) \ge \frac{1}{80}p^4 + O(p^3)$.

6.3.27. (!) It is conjectured that $\nu(K_{m,n}) = \left\lfloor\frac{m}{2}\right\rfloor\left\lfloor\frac{m-1}{2}\right\rfloor\left\lfloor\frac{n}{2}\right\rfloor\left\lfloor\frac{n-1}{2}\right\rfloor$. Suppose that this conjecture holds for $K_{m,n}$ and that m is odd. Prove that the conjecture then holds also for $K_{m+1,n}$. (Kleitman [1970])

6.3.28. (!) Suppose that m and n are odd. Prove that in all drawings of $K_{m,n}$, the parity of the number of pairs of edges that cross is the same. (We consider only drawings where edges cross at most once and edges sharing an endpoint do not cross.) Conclude that $\nu(K_{m,m})$ is odd when $m-3$ and $n-3$ are divisible by 3 and even otherwise.

6.3.29. Suppose that n is odd. Prove that in all drawings of K_n, the parity of the number of pairs of edges that cross is the same. Conclude that $\nu(K_n)$ is even when n is congruent to 1 or 3 modulo 8 and is odd when n is congruent to 5 or 7 modulo 8.

6.3.30. (!) It is known that $\nu(C_m \square C_n) = (m-2)n$ if $m \le \min\{5, n\}$. Also $\nu(K_4 \square C_n) = 3n$.

a) Find drawings in the plane to establish the upper bounds.

b) Prove that $\nu(C_3 \square C_3) \ge 2$. (Hint: Find three subdivisions of $K_{3,3}$ that together use each edge exactly twice.)

6.3.31. Let $f(n) = \nu(K_{n,n,n})$.

a) Show that $3\nu(K_{n,n}) \le f(n) \le 3\binom{n}{2}^2$.

b) Show that $\nu(K_{3,2,2}) = 2$ and $\nu(K_{3,3,1}) = 3$. Show that $5 \le \nu(K_{3,3,2}) \le 7$ and $9 \le \nu(K_{3,3,3}) \le 15$.

c) Exercise 6.3.26a shows that the lower bound in part (a) is at least $(3/20)n^4 + O(n^3)$. Improve it by using a recurrence to show that $f(n) \ge n^3(n-1)/6$.

d) The upper bound in part (a) is $\frac{3}{4}n^4 + O(n^3)$. Improve it to $f(n) \le \frac{9}{16}n^4 + O(n^3)$. (Hint: One construction embeds the graph on a tetrahedron and generalizes to a construction for $K_{l,m,n}$; another uses K_n and generalizes to a construction for $K_{n,\dots,n}$.)

6.3.32. (∗) Construct an embedding of a 3-regular nonbipartite simple graph on the torus so that every face has even length.

6.3.33. (∗) Suppose that n is at least 9 and is not a prime or twice a prime. Construct a 6-regular toroidal graph with n vertices.

6.3.34. (∗) An embedding of a graph on a surface is **regular** if its faces all have the same length. Construct regular embeddings of $K_{4,4}$, $K_{3,6}$, and $K_{3,3}$ on the torus.

6.3.35. (∗) Prove Euler's Formula for genus γ: For every 2-cell embedding of a graph on the surface S_γ, the numbers of vertices, edges, and faces satisfy $n - e + f = 2 - 2\gamma$. Conclude that an n-vertex graph embeddable on S_γ has at most $3(n-2+2\gamma)$ edges.

6.3.36. (∗) Use Euler's Formula for S_γ to prove that $\gamma(K_{3,3,n}) \ge n-2$, and determine the value exactly for $n \le 3$.

6.3.37. (∗) For every positive integer k, use Euler's Formula for higher surfaces to prove that there exists a planar graph G such that $\gamma(G \square K_2) \ge k$.

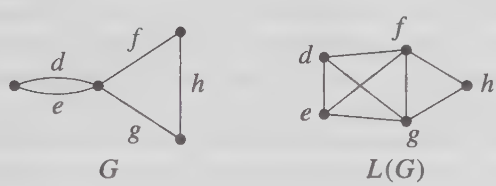
Chapter 7

Edges and Cycles

7.1. Line Graphs and Edge-coloring

Many questions about vertices have natural analogues for edges. Independent sets have no adjacent vertices; matchings have no "adjacent" edges. Vertex colorings partition vertices into independent sets; we can instead partition edges into matchings. These pairs of problems are related via line graphs (Definition 4.2.18). Here we repeat the definition, emphasizing our return to the context in which a graph may have multiple edges. We use "line graph" and $L(G)$ instead of "edge graph" because $E(G)$ already denotes the edge set.

7.1.1. Definition. The **line graph** of G, written $L(G)$, is the simple graph whose vertices are the edges of G, with $ef \in E(L(G))$ when e and f have a common endpoint in G.



Some questions about edges in a graph G can be phrased as questions about vertices in $L(G)$. When extended to all simple graphs, the vertex question may be more difficult. If we can solve it, then we can answer the original question about edges in G by applying the vertex result to $L(G)$.

In Chapter 1, we studied Eulerian circuits. An Eulerian circuit in G yields a spanning cycle in the line graph $L(G)$. (Exercise 7.2.10 shows that the converse need not hold!) In Section 7.2, we study spanning cycles for graphs in general. As discussed in Appendix B, this problem is computationally difficult.

In Chapter 3, we studied matchings. A matching in G becomes an independent set in $L(G)$. Thus $\alpha'(G) = \alpha(L(G))$, and the study of α' for graphs is the study of α for line graphs. Computing α is harder for general graphs than for line graphs. Section 3.1 considers this for bipartite graphs, and we describe the general case briefly in Appendix B.

In Chapter 4, we studied connectivity. Menger's Theorem gave a min-max relation for connectivity and internally disjoint paths in all graphs. By applying this theorem to an appropriate line graph, we proved the analogous min-max relation for edge-connectivity and edge-disjoint paths in all graphs.

In Chapter 5, we studied vertex coloring. Coloring edges so that each color class is a matching amounts to proper vertex coloring of the line graph. Thus edge-coloring is a special case of vertex coloring and therefore potentially easier. We discuss edge-coloring in this section. Our main result, when stated in terms of vertex coloring of line graphs, is an algorithm to compute $\chi(H)$ within 1 when H is the line graph of a simple graph.

Thus line graphs suggest the problems of edge-coloring and spanning cycles that are discussed in this chapter. We first study these separately. In Section 7.3, we study their connections to each other and to planar graphs.

In applying algorithms for line graphs, we may need to know whether G is a line graph. There are good algorithms to check this; they use characterizations of line graphs, which we postpone to the end of this section.

EDGE-COLORINGS

In Example 1.1.11 that introduced vertex coloring, we needed to schedule Senate committees. Edge-coloring problems arise when the objects being scheduled are pairs of underlying elements.

7.1.2. Example. *Edge-coloring of K_{2n}.* In a league with $2n$ teams, we want to schedule games so that each pair of teams plays a game, but each team plays at most once a week. Since each team must play $2n-1$ others, the season lasts at least $2n-1$ weeks. The games of each week must form a matching. We can schedule the season in $2n-1$ weeks if and only if we can partition $E(K_{2n})$ into $2n-1$ matchings. Since K_{2n} is $2n-1$-regular, these must be perfect matchings.

The figure below describes the solution. Put one vertex in the center. Arrange the other $2n-1$ vertices cyclically, viewed as congruence classes modulo $2n-1$. As in Theorem 2.2.16, the *difference* between two congruence classes is 1 if they are consecutive, 2 if there is one class between them, and so on up to difference $n-1$. There are $2n-1$ edges with each difference i, for $1 \le i \le n-1$.

Each matching consists of one edge from each difference class plus one edge involving the center vertex. We show one such matching in bold. Rotating the picture (to obtain the solid matching) yields n new edges; again they are one of each length plus one to the center. The $2n-1$ rotations of the figure yield the desired matchings, since these matchings take distinct edges from each difference class and distinct edges involving the center vertex. ■

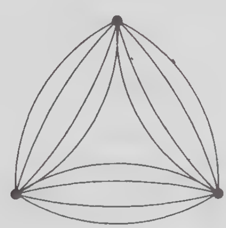
7.1.3. Definition. A k**-edge-coloring** of G is a labeling $f: E(G) \to S$, where $|S| = k$ (often we use $S = [k]$). The labels are **colors**; the edges of one color form a **color class**. A k-edge-coloring is **proper** if incident edges have different labels; that is, if each color class is a matching. A graph is k**-edge-colorable** if it has a proper k-edge-coloring. The **edge-chromatic number** $\chi'(G)$ of a loopless graph G is the least k such that G is k-edge-colorable.

Chromatic index is another name for $\chi'(G)$. Since edges sharing a vertex need different colors, $\chi'(G) \geq \Delta(G)$. Vizing [1964] and Gupta [1966] independently proved that $\Delta(G)+1$ colors suffice when G is simple; this is our main objective. A clique in $L(G)$ is a set of pairwise-intersecting edges of G. When G is simple, such edges form a star or a triangle in G (Exercise 9). For the hereditary class of line graphs of simple graphs, Vizing's Theorem thus states that $\chi(H) \leq \omega(H)+1$; thus line graphs are "almost" perfect.

In contrast to $\chi(G)$ in Chapter 5, multiple edges greatly affect $\chi'(G)$. A graph with a loop has no proper edge-coloring; the adjective "loopless" excludes loops but allows multiple edges.

7.1.4. Definition. In a graph G with multiple edges, we say that a vertex pair x, y is an edge of **multiplicity** m if there are m edges with endpoints x, y. We write $\mu(xy)$ for the multiplicity of the pair, and we write $\mu(G)$ for the maximum of the edge multiplicities in G.

7.1.5. Example. *The "Fat Triangle".* For loopless graphs with multiple edges, $\chi'(G)$ may exceed $\Delta(G)+1$. Shannon [1949] proved that the maximum of $\chi'(G)$ in terms of $\Delta(G)$ alone is $3\Delta(G)/2$ (see Theorem 7.1.13). Vizing and Gupta proved that $\chi'(G) \leq \Delta(G)+\mu(G)$, where $\mu(G)$ is the maximum edge multiplicity. The graph below achieves both bounds. The edges are pairwise intersecting and hence require distinct colors. Thus $\chi'(G) = 3\Delta(G)/2 = \Delta(G)+\mu(G)$. ■



7.1.6. Remark. We have observed that always $\chi'(G) \geq \Delta(G)$. The upper bound $\chi'(G) \leq 2\Delta(G)-1$ also follows easily. Color the edges in some order, always assigning the current edge the least-indexed color different from those already appearing on edges incident to it. Since no edge is incident to more than $2(\Delta(G)-1)$ other edges, this never uses more than $2\Delta(G)-1$ colors. The procedure is precisely greedy coloring for vertices of $L(G)$.

$$\chi'(G) = \chi(L(G)) \le \Delta(L(G)) + 1 \le 2\Delta(G) - 1. \qquad \blacksquare$$

For bipartite graphs, the results of Chapter 3 improve the upper bound of Remark 7.1.6, achieving the trivial lower bound even when multiple edges are allowed. Furthermore, there is a good algorithm to produce a proper $\Delta(G)$-edge-coloring in a bipartite graph G.

7.1.7. Theorem. (König [1916]) If G is bipartite, then $\chi'(G) = \Delta(G)$.

Proof: Corollary 3.1.13 states that every regular bipartite graph H has a 1-factor. By induction on $\Delta(H)$, this yields a proper $\Delta(H)$-edge-coloring. It therefore suffices to show that for every bipartite graph G with maximum degree k, there is a k-regular bipartite graph H containing G.

To construct such a graph, first add vertices to the smaller partite set of G, if necessary, to equalize the sizes. If the resulting G' is not regular, then each partite set has a vertex with degree less than k. Add an edge with these two vertices as endpoints. Continue adding such edges until the graph becomes k-regular; the resulting graph is H. ■

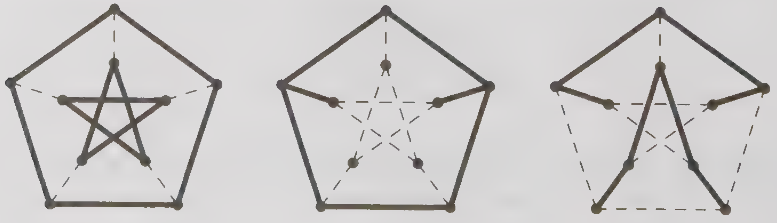
For a regular graph G, proper edge-coloring with $\Delta(G)$ colors is equivalent to decomposition into 1-factors.

7.1.8. Definition. A decomposition of a regular graph G into 1-factors is a **1-factorization** of G. A graph with a 1-factorization is **1-factorable**.

An odd cycle is not 1-factorable; $\chi'(C_{2m+1}) = 3 > \Delta(C_{2m+1})$. The Petersen graph also requires an extra color, but only one extra color.

7.1.9. Example. *The Petersen graph is 4-edge-chromatic* (Petersen [1898]). The Petersen graph is 3-regular; 3-edge-colorability requires a 1-factorization. Deleting a perfect matching leaves a 2-factor; all components are cycles. The 1-factorization can be completed only if these are all even cycles.

Thus it suffices to show that every 2-factor is isomorphic to $2C_5$. Consider the drawing consisting of two 5-cycles and a matching (the **cross edges**) between them. We consider cases by the number of cross edges used.



Every cycle uses an even number of cross edges, so a 2-factor H has an even number m of cross edges. If $m = 0$ (left figure), then $H = 2C_5$.

If $m = 2$ (central figure), then the two cross edges have nonadjacent endpoints on the inner cycle or the outer cycle. On the cycle where their endpoints are nonadjacent, the remaining three vertices force all five edges of that cycle into H, which violates the 2-factor requirement.

If $m = 4$ (right figure), then the cycle edges forced into H by the unused cross edges form a $2P_5$ whose only completion to a 2-factor in H is $2C_5$.

Note that since C_5 is 3-edge-colorable, the graph is 4-edge-colorable. ■

Now we consider all simple graphs. We make $\Delta(G) + 1$ colors available and build a proper edge-coloring, incorporating edges one by one until we have a proper $\Delta(G) + 1$-edge-coloring of G. The algorithm runs surprisingly quickly.

7.1.10. Theorem. (Vizing [1964, 1965], Gupta [1966]) If G is a simple graph, then $\chi'(G) \leq \Delta(G) + 1$.

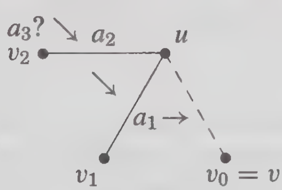
Proof: Let f be a proper $\Delta(G) + 1$-edge-coloring of a subgraph G' of G. If $G' \neq G$, then some edge uv is uncolored by f. After possibly recoloring some edges, we extend the coloring to include uv; call this an *augmentation*. After $e(G)$ augmentations, we obtain a proper $\Delta(G) + 1$-edge-coloring of G.

Since the number of colors exceeds $\Delta(G)$, every vertex has some color *not* appearing on its incident edges. Let a_0 be a color missing at u. We generate a list of neighbors of u and a corresponding list of colors. Begin with $v_0 = v$.

Let a_1 be a color missing at v_0. We may assume that a_1 appears at u on some edge uv_1; otherwise, we would use a_1 on uv_0.

Let a_2 be a color missing at v_1. We may assume that a_2 appears at u on some edge uv_2; otherwise, we would replace color a_1 with a_2 on uv_1 and then use a_1 on uv_0 to augment the coloring.

Having selected uv_{i-1} with color a_{i-1}, let a_i be a color missing at v_{i-1}. If a_i is missing at u, then we use a_i on uv_{i-1} and shift color a_j from uv_j to uv_{j-1} for $1 \leq j \leq i-1$ to complete the augmentation. We call this *downshifting from i*. If a_i appears at u (on some edge uv_i), then the process continues.



Since we have only $\Delta(G)+1$ colors to choose from, the list of selected colors eventually repeats (or we complete the augmentation by downshifting). Let l be the smallest index such that a color missing at v_l is in the list $a_1, \ldots, a_l$; let this color be a_k. Instead of extending the list, we use this repetition to perform the augmentation in one of several ways.

The color a_k missing at v_l is also missing at v_{k-1} and appears on uv_k. If a_0 does not appear at v_l, then we downshift from v_l and use color a_0 on uv_l to complete the augmentation. Hence we may assume that a_0 appears at v_l.

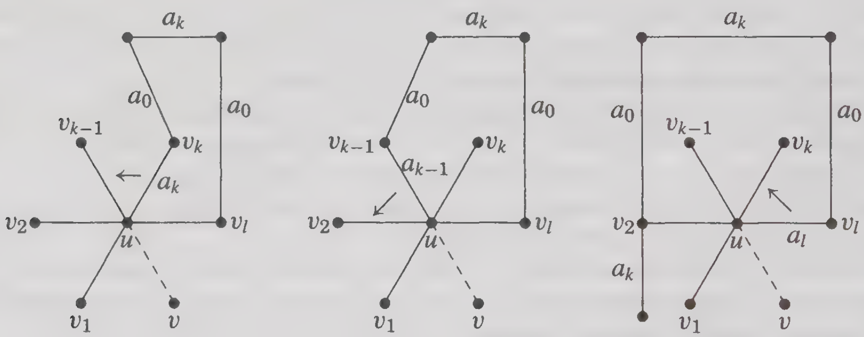
Let P be the maximal alternating path of edges colored a_0 and a_k that begins at v_l along color a_0. There is only one such path, because each vertex has at most one incident edge in each color (we ignore edges not yet colored). To complete the augmentation, we will interchange colors a_0 and a_k on P and downshift from an appropriate neighbor of u, depending on where P goes.

If P reaches v_k, then it arrives at v_k along an edge with color a_0, follows $v_k u$ in color a_k, and stops at u, which lacks color a_0. In this case, we downshift from v_k and switch colors on P (left picture below).

If P reaches v_{k-1}, then it reaches v_{k-1} on color a_0 and stops there, because a_k does not appear at v_{k-1}. In this case, we downshift from v_{k-1}, give color a_0 to uv_{k-1}, and switch colors on P (middle picture).

If P does not reach v_k or v_{k-1}, then it ends at some vertex outside $\{u, v_l, v_k, v_{k-1}\}$. In this case, we downshift from v_l, give color a_0 to uv_l, and switch colors on P (rightmost picture).

In each case, the changes described yield a proper $\Delta(G)+1$-edge-coloring of $G' + uv$, so we have completed the desired augmentation. ■



P reaches v_k $\quad$ P reaches v_{k-1} $\quad$ otherwise

For simple graphs, we now have only two possibilities for χ'.

7.1.11. Definition. A simple graph G is **Class 1** if $\chi'(G) = \Delta(G)$. It is **Class 2** if $\chi'(G) = \Delta(G) + 1$.

Determining whether a graph is Class 1 or Class 2 is generally hard (Holyer [1981]; see Appendix B). Thus we seek conditions that forbid or guarantee $\Delta(G)$-edge-colorability. Examples of such conditions include Exercises 24–27.

7.1.12.* Remark. There is an obvious necessary condition for a graph to be Class 1 that is conjectured to be sufficient when $\Delta(G) > \frac{3}{10}n(G)$. Part (a) of Exercise 27 observes that a subgraph of G with odd order is an obstruction to $\Delta(G)$-edge-colorability if it has too many edges. A subgraph H of a simple graph G is an **overfull subgraph** if $n(H)$ is odd and $2e(H)/(n(H)-1) > \Delta(G)$.

The **Overfull Conjecture** (Chetwynd–Hilton [1986]—see also Hilton [1989]) states that if $\Delta(G) > n(G)/3$, then a simple graph G is Class 1 if and only if G has no overfull subgraph. The Petersen graph with a vertex deleted shows that the condition is not sufficient when $\Delta(G) = n(G)/3$ (Exercise 28).

The Overfull Conjecture implies the **1-factorization Conjecture**: If $r \geq m$ (or $r \geq m - 1$ if m is even), then every r-regular simple graph of order $2m$ is Class 1. This also is sharp (Exercise 29).

The conclusions of the two conjectures hold when $\Delta(G)$ is large enough (Chetwynd–Hilton [1989], Niessen–Volkmann [1990], Perkovic–Reed [1997], Plantholt [2001]. ■

When G has multiple edges, $\chi'(G) \leq \lfloor 3\Delta(G)/2 \rfloor$ (Shannon [1949]) and $\chi'(G) \leq \Delta(G) + \mu(G)$ (Vizing [1964, 1965], Gupta [1966]) These bounds follow (Exercise 35) from that of Andersen [1977] and Goldberg [1977, 1984]:

$$\chi'(G) \leq \max\{\Delta(G), \max_{\mathbf{P}} \left\lfloor \tfrac{1}{2}(d(x) + \mu(xy) + \mu(yz) + d(z)) \right\rfloor\}$$

where $\mathbf{P} = \{x, y, z \in V(G) \colon y \in N(x) \cap N(z)\}$. Proving this bound uses the methods of Theorem 7.1.10 plus counting arguments. To illustrate the use of counting arguments, we prove Shannon's Theorem from that of Vizing and Gupta.

7.1.13.* Theorem. (Shannon [1949]) If G is a graph, then $\chi'(G) \leq \frac{3}{2}\Delta(G)$.

Proof: Let $k = \chi'(G)$, and assume $k \geq (3/2)\Delta(G)$. Let G' be a minimal subgraph of G with $\chi'(G') = k$. Since $k \leq \Delta(G') + \mu(G')$ (Vizing–Gupta), we obtain $\mu(G') \geq \Delta(G)/2$. Let e with endpoints x, y be an edge with multiplicity $\mu(G')$.

Let f be a proper $k - 1$-edge-coloring of $G' - e$. In $G' - e$, both x and y have degree at most $\Delta(G) - 1$, so under f at least $(k - 1) - (\Delta(G) - 1)$ colors are missing at x, and similarly at y. No color is missing at both, since G' is not $k - 1$-edge-colorable. Accounting for the $\mu(G') - 1$ colors used on edges with endpoints x, y yields

$$2(k - \Delta(G)) + (\Delta(G)/2) - 1 \leq 2(k - \Delta(G)) + \mu(G') - 1 \leq k - 1,$$

and hence $k \leq (3/2)\Delta(G)$. ■

Finally, there is a general conjecture analogous to the Overfull Conjecture.

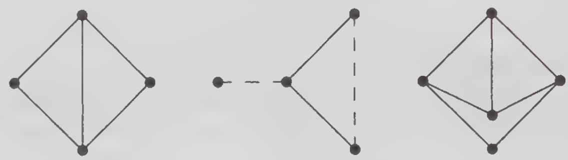
7.1.14.* Conjecture. (Goldberg [1973, 1984], Seymour [1979a])
If $\chi'(G) \geq \Delta(G) + 2$, then $\chi'(G) = \max_{H \subseteq G} \left\lceil \frac{e(H)}{\lfloor n(H)/2 \rfloor} \right\rceil$. ■

CHARACTERIZATION OF LINE GRAPHS (optional)

Characterizations of line graphs can lead to good algorithms to test whether a graph G is a line graph and, if so, to obtain H such that $L(H) = G$.

7.1.15. Example. To illustrate the ideas, we prove that the rightmost graph below is not the line graph of a simple graph. The kite G (two triangles with a common edge) is the line graph of the paw H (a claw plus an edge). By case analysis, we find that H is the only simple graph whose line graph is G, and the edges becoming the vertices of degree 2 in G must be the dashed edges.

The rightmost graph adds a vertex to G having only the vertices of degree 2 as neighbors. The result is not a line graph, because there is no way to add an edge to H that shares an endpoint with each dashed edge without sharing an endpoint with a solid edge. ■



$G = L(H)$ $\qquad$ H $\qquad$ not a line graph

Our first characterization encodes the process of taking the line graph. If $G = L(H)$ and H is simple, then each $v \in V(H)$ with $d(v) \geq 2$ generates a clique $Q(v)$ in G corresponding to edges incident to v. These cliques partition $E(G)$. Furthermore, each vertex $e \in V(G)$ belongs only to the cliques generated by the two endpoints of $e \in E(H)$.

For example, when G is the kite, we can partition $E(G)$ into three cliques (a triangle plus two edges), each vertex covered at most twice. These three cliques correspond to the vertices of degree at least 2 in the paw. The rightmost graph above does not have such a partition.

7.1.16. Theorem. (Krausz [1943]) For a simple graph G, there is a solution to $L(H) = G$ if and only if G decomposes into complete subgraphs, with each vertex of G appearing in at most two in the list.

Proof: We argued above that the condition is necessary. Note that when $G = L(H)$, the vertices of G that belong to only one of the cliques we have defined are those corresponding to edges of H that are incident to leaves.

For sufficiency, let $S_1, \dots, S_k$ be the vertex sets of the specified complete subgraphs. We construct H such that $G = L(H)$. Isolated vertices of G become isolated edges of H, so we may assume that $\delta(G) \geq 1$. Let $v_1, \dots, v_l$ be the vertices of G (if any) that appear in exactly one of $S_i, \dots, S_1n$. Give H one vertex for each set in the list $\mathbf{A} = S_1, \dots, S_k, \{v_1\}, \dots, \{v_l\}\}$, and let vertices of H be adjacent if the corresponding sets intersect.

Each vertex of G appears in exactly two sets in $\mathbf{A}$, and no two vertices appear in the same two sets. Hence H is a simple graph with one edge for each vertex of G. If vertices are adjacent in G, then they appear together in some S_i, and the corresponding edges of H share the vertex for S_i. Hence $G = L(H)$. ■

Krausz's characterization does not directly yield an efficient test for line graphs, because there are too many possible decompositions to test. The next characterization tests substructures of fixed size and therefore yields a good algorithm. We say that each triangle T in G is odd or even as defined below.

T is **odd** if $|N(v) \cap V(T)|$ is odd for *some* $v \in V(G)$.
T is **even** if $|N(v) \cap V(T)|$ is even for *every* $v \in V(G)$.

An induced kite is a **double triangle**; it consists of two triangles sharing an edge, and the two vertices not in that edge are nonadjacent.

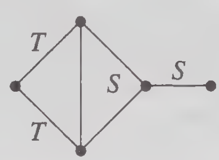
7.1.17. Theorem. (van Rooij and Wilf [1965]) For a simple graph G, there is a solution to $L(H) = G$ if and only if G is claw-free and no double triangle of G has two odd triangles.

Proof: *Necessity.* Suppose that $G = L(H)$. A vertex e in G with neighbors x, y, z corresponds to an edge e in H incident to edges x, y, z. Since e has only two endpoints in H, two of x, y, z are incident at one of them and hence are adjacent in G. This forbids the claw as an induced subgraph of G.

For the other condition, we saw in Example 7.1.15 that the vertices of a double triangle in G must correspond to the edges of a paw in H. In particular, the vertices of one of these triangles in G correspond to the edges of a triangle in H. This triangle must be even, because every edge in H incident to exactly one vertex of a triangle shares an endpoint with exactly two of its edges. Hence for each double triangle in G, at least one of its triangles is even.

Sufficiency. Suppose that G satisfies the specified conditions. We may assume that G is connected; otherwise, we apply the construction to each component. The case where G is claw-free and has a double triangle with both triangles even is very special; there are only three such graphs (Exercise 38). Here we consider only the general case, in which every double triangle of G has exactly one odd triangle.

By Theorem 7.1.16, it suffices to decompose G into complete subgraphs, using each vertex in at most two of them. Let $S_1, \dots, S_k$ be the maximal complete subgraphs of G that are not even triangles, and let $T_1, \dots, T_l$ be the edges that belong to one even triangle and no odd triangle. We claim that together these form the desired decomposition **B**.



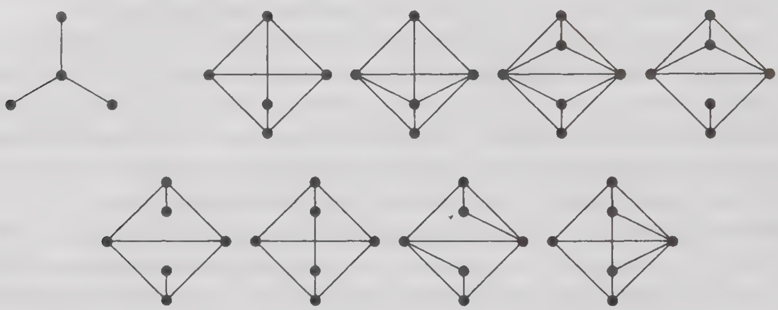
Every edge appears in a maximal complete subgraph, but every triangle in a complete subgraph with more than three vertices is odd. Hence each edge T_j in the list is not in any S_i. Also S_i and $S_{i'}$ share no edge, because G has no double triangles with both triangles odd. Hence the subgraphs in **B** are pairwise edge-disjoint.

If $e \in E(G)$, then e is in some S_i unless the only maximal clique containing e is an even triangle. In this case e is a T_j, since we have forbidden double triangles with both triangles even. Hence **B** is a decomposition.

It remains to show that each $v \in G$ appears in at most two of these subgraphs. Suppose that v belongs to $A, B, C \in \mathbf{B}$. Edge-disjointness implies that v has neighbors x, y, z with each belonging to only one of $\{A, B, C\}$. Since G has no induced claw, we may assume that $x \leftrightarrow y$. By edge-disjointness, the triangle vxy cannot belong to a member of **B**. Hence it must be an even triangle. Therefore, z must have exactly one other edge to vxy, say $z \leftrightarrow x$ and $z \not\leftrightarrow y$. But now the same argument shows zvx is an even triangle, and we have a double triangle with both triangles even. ∎

Theorem 7.1.17 is close to a forbidden subgraph characterization.

7.1.18. Theorem. (Beineke [1968]) A simple graph G is the line graph of some simple graph if and only if G does not have any of the nine graphs below as an induced subgraph.



Proof: By Theorem 7.1.17, it suffices to show that the eight graphs listed other than $K_{1,3}$ are the vertex-minimal claw-free graphs containing a double triangle with both triangles odd. Each such graph has a double triangle and one or two additional vertices that make the triangles odd by having one or three neighbors in the triangles. The details of showing that this is the full list are requested in Exercise 40. ∎

The characterizations in Theorems 7.1.17–7.1.18 yield algorithms to test whether G is a line graph that run in time polynomial in $n(G)$. In fact, there is such an algorithm that runs in linear time (Lehot [1974]) and produces a graph H such that $G = L(H)$ when G is a line graph. This graph H is unique if G has no component that is a triangle (Exercise 39).

EXERCISES

7.1.1. (−) For each graph G below, compute $\chi'(G)$ and draw $L(G)$.

7.1.2. (−) Give an explicit edge-coloring to prove that $\chi'(Q_k) = \Delta(Q_k)$

7.1.3. (−) Determine the edge-chromatic number of $C_n \square K_2$.

7.1.4. (−) Obtain an inequality for $\chi'(G)$ in terms of $e(G)$ and $\alpha'(G)$.

7.1.5. (−) Prove that the Petersen graph is the complement of $L(K_5)$.

7.1.6. (−) Determine the number of triangles in the line graph of the Petersen graph.

7.1.7. (−) Determine whether $\overline{P}_5$ is a line graph. If so, find H such that $L(H) = \overline{P}_5$.

7.1.8. (−) Prove that $L(K_{m,n}) \cong K_m \square K_n$.

• • • • •

7.1.9. Let G be a simple graph. Prove that vertices form a clique in $L(G)$ if and only if the corresponding edges in G have one common endpoint or form a triangle. (Comment: Thus $\omega(L(G)) = \Delta(G)$ unless $\Delta(G) = 2$ and some component of G is a triangle.)

7.1.10. Let G be a simple graph without isolated vertices. Prove that if $L(G)$ is connected and regular, then either G is regular or G is a bipartite graph in which vertices of the same partite set have the same degree. (Ray-Chaudhuri [1967])

7.1.11. (!) Let G be a simple graph.
a) Prove that the number of edges in $L(G)$ is $\sum_{v \in V(G)} \binom{d(v)}{2}$.
b) Prove that G is isomorphic to $L(G)$ if and only if G is 2-regular.

7.1.12. Let G be a connected simple graph. Use part (a) of Exercise 7.1.11 to determine when $e(L(G)) < e(G)$.

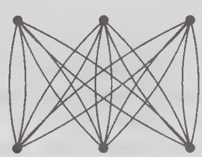
7.1.13. (+) Prove that the graph below is the only simple graph whose line graph is isomorphic to its complement. (Albertson)

7.1.14. (!) Let G be a k-edge-connected simple graph. Prove that $L(G)$ is k-connected and is $2k-2$-edge-connected. (Hint: For a minimum edge cut $[S, \overline{S}]$ in $L(G)$, describe what the cut corresponds to in G and count its edges in terms of the vertices of G.)

7.1.15. (!) Use Tutte's 1-factor Theorem to prove that every connected line graph of even order has a perfect matching. Conclude from this that the edges of a simple connected graph of even size can be partitioned into paths of length 2. (Comment: Exercise 3.3.22 shows that every connected claw-free graph has a perfect matching , but that stronger result is more difficult than this.) (Chartrand–Polimeni–Stewart [1973])

7.1.16. (∗) Let G be a simple graph. Prove that $\gamma(L(G)) \geq \gamma(G)$, where $\gamma(G)$ denotes the genus of G (Definition 6.3.20). (D. Greenwell)

7.1.17. Compute the number of proper 6-edge-colorings of the graph below.



7.1.18. (!) Give an explicit edge-coloring to prove that $\chi'(K_{r,s}) = \Delta(K_{r,s})$.

7.1.19. (!) Prove that for every simple bipartite graph G, there is a $\Delta(G)$-regular simple bipartite graph H that contains G.

7.1.20. (!) Let D be a digraph (loops allowed) such that $d^+(v) \le d$ and $d^-(v) \le d$ for all $v \in V(D)$. Prove that $E(D)$ can be colored using at most d colors so that the edges entering each vertex have distinct colors and the edges exiting each vertex have distinct colors. (Hint: Transform the digraph into another object where a known result applies.)

7.1.21. *Algorithmic proof of Theorem 7.1.7.* Let G be a bipartite graph with maximum degree k. Let f be a proper k-edge-coloring of a subgraph H of G. Let uv be an edge not in H. By using a path alternating in two colors, show that f can be altered and then extended to a proper k-edge-coloring of $H + uv$. Conclude that $\chi'(G) = \Delta(G)$.

7.1.22. Use Brooks' Theorem to an appropriate graph to prove that if G is a simple graph with $\Delta(G) = 3$, then G is 4-edge-colorable. (Comment: The result is a special case of Vizing's Theorem; do not use Vizing's Theorem to prove this.)

7.1.23. (+) Let $K(p,q)$ be the complete p-partite graph with q vertices in each partite set. Let $G[H]$ denote the composition operation, in which each vertex of G expands into a copy of H. Note that $K(p,q) = K(p,d)[\overline{K}_{q/d}]$ when d divides q.

a) Show that if G has a decomposition into copies of F, then $G[\overline{K}_m]$ has a decomposition into copies of $F[\overline{K}_m]$. Show also that the relation "G decomposes into spanning copies of F" is transitive.

b) Cliques of even order decompose into 1-factors. Cliques of odd order decompose into spanning cycles. Use these statements and part (a) to prove that $K(p,q)$ decomposes into 1-factors when pq is even. (Hartman [1997])

7.1.24. (!) Let G and H be nontrivial simple graphs. Use Vizing's Theorem to prove that $\chi'(H) = \Delta(H)$ implies $\chi'(G \,\square\, H) = \Delta(G \,\square\, H)$.

7.1.25. *Kotzig's Theorem for cartesian products of simple graphs.*

a) Use Vizing's Theorem to prove that $\chi'(G \,\square\, K_2) = \Delta(G \,\square\, K_2)$.

b) Let G_1, G_2 be edge-disjoint graphs with vertex set V, and let H_1, H_2 be edge-disjoint graphs with vertex set W. Prove that $(G_1 \cup G_2) \square (H_1 \cup H_2) = (G_1 \square H_2) \cup (G_2 \square H_1)$.

c) Use parts (a) and (b) to prove that $\chi'(G \,\square\, H) = \Delta(G \,\square\, H)$ if both G and H have 1-factors. (Comment: As a result, the product of the Petersen graph with itself is Class 1, which does not follow from Exercise 7.1.24. Here neither factor need be Class 1; there G need not have a 1-factor.) (Kotzig [1979], J. George [1991])

7.1.26. (!) Let G be a regular graph with a cut-vertex. Prove that $\chi'(G) > \Delta(G)$.

7.1.27. *Density conditions for $\chi'(G) > \Delta(G)$.*

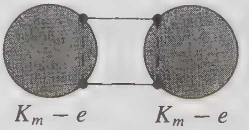
a) Prove that if $n(G) = 2m + 1$ and $e(G) > m \cdot \Delta(G)$, then $\chi'(G) > \Delta(G)$.

b) Prove that if G is obtained from a k-regular graph with $2m+1$ vertices by deleting fewer than $k/2$ edges, then $\chi'(G) > \Delta(G)$.

c) Prove that if G is obtained by subdividing an edge of a regular graph with $2m$ vertices and degree at least 2, then $\chi'(G) > \Delta(G)$.

7.1.28. (∗−) Prove that the Petersen graph has no overfull subgraph.

7.1.29. Let G be the $m-1$-regular connected graph formed from $2K_m$ by deleting an edge from each component and adding two edges between the components to restore regularity. Prove that G is not 1-factorable if m is odd and greater than 3. (Comment: This shows that the 1-factorization Conjecture (Remark 7.1.12) is sharp.)



7.1.30. (∗!) *Overfull Conjecture ⇒ 1-factorization Conjecture* (Remark 7.1.12).

a) Prove that in a regular graph of even order, an induced subgraph is overfull if and only if the subgraph induced by the other vertices is overfull.

b) Let G be an k-regular graph of order $2m$ having an overfull subgraph. Prove that $k < m$ if m is odd and that $k < m - 1$ if m is even.

7.1.31. Given an edge-coloring of a graph G, let $c(v)$ denote the number of distinct colors appearing on edges incident to v. Among all k-edge-colorings of G, a coloring is **optimal** if it maximizes $\sum_{v \in V(G)} c(v)$.

a) Prove that if no component is an odd cycle, then G has a 2-edge-coloring where both colors appear at each vertex of degree at least 2. (Hint: Use Eulerian circuits.)

b) Let f be an optimal k-edge-coloring of G in which color a appears at least twice at $u \in V(G)$ and color b does not appear at u. Let H be the subgraph of G consisting of edges colored a or b. Prove that the component of H containing u is an odd cycle.

c) Let G be a bipartite graph. Conclude from part (b) that G is $\Delta(G)$-edge-colorable. (Comment: These ideas also lead to a proof of Vizing's Theorem.) (Fournier [1973])

7.1.32. Let G be a bipartite graph with minimum degree k. Prove that G has a k-edge-coloring in which at each vertex v, each color appears $\lceil d(v)/k \rceil$ or $\lfloor d(v)/k \rfloor$ times. (Hint: Use a graph transformation.) (Gupta [1966])

7.1.33. Use Vizing's Theorem to prove that every simple graph with maximum degree Δ has an "equitable" $\Delta + 1$-edge-coloring: a proper edge-coloring with each color used $\lceil e(G)/(\Delta + 1) \rceil$ or $\lfloor e(G)/(\Delta + 1) \rfloor$ times. (de Werra [1971], McDiarmid [1972])

7.1.34. Use Petersen's Theorem (every $2k$-regular graph has a 2-factor—Theorem 3.3.9) to prove that $\chi'(G) \le 3 \lceil \Delta(G)/2 \rceil$ when G is a loopless graph.

7.1.35. *Bounds on* $\chi'(G)$. Let $\mathbf{P} = \{x, y, z \in V(G)\colon y \in N(x) \cap N(z)$. Prove that the last bound below (Andersen [1977], Goldberg [1977, 1984]) implies the earlier bounds.

$\chi'(G) \le \lfloor 3\Delta(G)/2 \rfloor$. (Shannon [1949])

$\chi'(G) \le \Delta(G) + \mu(G)$. (Vizing [1964, 1965], Gupta [1966])

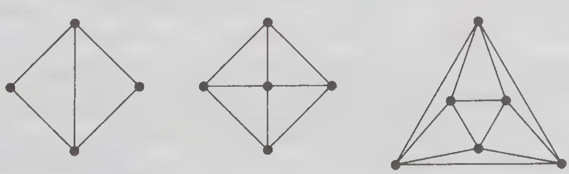
$\chi'(G) \le \max\{\Delta(G), \max_{\mathbf{P}} \left\lfloor \frac{1}{2}(d(x) + d(y) + d(z)) \right\rfloor\}$. (Ore [1967a])

$\chi'(G) \le \max\{\Delta(G), \max_{\mathbf{P}} \left\lfloor \frac{1}{2}(d(x) + \mu(xy) + \mu(yz) + d(z)) \right\rfloor\}$.

7.1.36. (+) For $n \ne 8$, prove that $L(K_n)$ is the only $2n - 4$-regular simple graph of order $\binom{n}{2}$ in which nonadjacent vertices have four common neighbors and adjacent vertices have $n - 2$ common neighbors. (Comment: When $n = 8$, three exceptional graphs satisfy the conditions.) (Chang [1959], Hoffman [1960])

7.1.37. (+) For n, m not both equalling 4, prove that $L(K_{m,n})$ is the only $(n+m-2)$-regular simple graph of order mn in which nonadjacent vertices have two common neighbors, $n\binom{m}{2}$ pairs of adjacent vertices have $m - 2$ common neighbors, and $m\binom{n}{2}$ pairs of adjacent vertices have $n - 2$ common neighbors. (Comment: When $n = m = 4$, there one exceptional graph—Shrikande [1959].) (Moon [1963], Hoffman [1964])

7.1.38. (∗) Let G be a connected, simple, claw-free graph having a double triangle H with each triangle even. Prove that G is one of the three graphs below, and conclude that G is a line graph. (Comment: This completes the proof of Theorem 7.1.17.)



7.1.39. (*) A **Krausz decomposition** of a simple graph H is a partition of $E(H)$ into cliques such that each vertex of H appears in at most two of the cliques.

a) Prove that for a connected simple graph H, two Krausz decompositions of H that have a common clique are identical.

b) Find distinct Krausz decompositions for the graphs in Exercise 7.1.38.

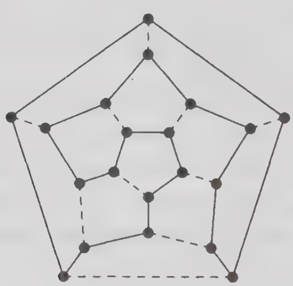
c) Prove that no other connected simple graph except K_3 has two distinct Krausz decompositions (use Exercise 7.1.38 and the proof of Theorem 7.1.17).

d) Conclude that $K_{1,3}$, K_3 is the only pair of nonisomorphic connected simple graphs with isomorphic line graphs. (Whitney [1932a])

7.1.40. (*) Complete the proof of Theorem 7.1.18 by proving that a simple graph with no induced claw has a double triangle with both triangles odd if and only if it contains an induced subgraph among the other eight graphs listed in the theorem statement.

7.2. Hamiltonian Cycles

Studied first by Kirkman [1856], Hamiltonian cycles are named for Sir William Hamilton, who described a game on the graph of the dodecahedron in which one player specifies a 5-vertex path and the other must extend it to a spanning cycle. The game was marketed as the "Traveller's Dodecahedron", a wooden version in which the vertices were named for 20 important cities.



7.2.1. Definition. A **Hamiltonian graph** is a graph with a spanning cycle, also called a **Hamiltonian cycle**.

Until the 1970s, interest in Hamiltonian cycles centered on their relationship to the Four Color Problem (Section 7.3). Later study was stimulated by practical applications and by the issue of complexity (Appendix B).

No easily testable characterization is known for Hamiltonian graphs; we will study necessary conditions and sufficient conditions. Loops and multiple edges are irrelevant; a graph is Hamiltonian if and only if the simple graph obtained by keeping one copy of each non-loop edge is Hamiltonian. Therefore, **in this section we restrict our attention to simple graphs**; this is relevant when discussing conditions involving vertex degrees.

For further material on Hamiltonian cycles, see Chvátal [1985a].

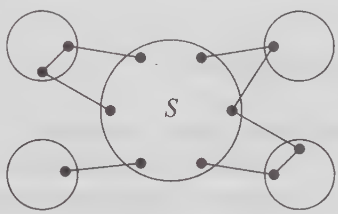
NECESSARY CONDITIONS

Every Hamiltonian graph is 2-connected, because deleting a vertex leaves a subgraph with a spanning path. Bipartite graphs suggest a way to strengthen this necessary condition.

7.2.2. Example. *Bipartite graphs.* A spanning cycle in a bipartite graph visits the two partite sets alternately, so there can be no such cycle unless the partite sets have the same size. Hence $K_{m,n}$ is Hamiltonian only if $m = n$. Alternatively, we can argue that the cycle returns to different vertices of one partite set after each visit to the other partite set. ■

7.2.3. Proposition. If G has a Hamiltonian cycle, then for each nonempty set $S \subseteq V$, the graph $G - S$ has at most $|S|$ components.

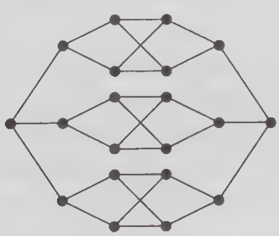
Proof: When leaving a component of $G - S$, a Hamiltonian cycle can go only to S, and the arrivals in S must use distinct vertices of S. Hence S must have at least as many vertices as $G - S$ has components. ■

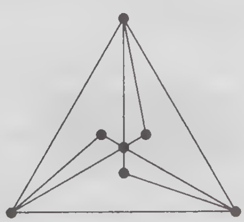


7.2.4. Definition. Let $c(H)$ denote the number of components of a graph H.

Thus the necessary condition is that $c(G - S) \leq |S|$ for all $\emptyset \neq S \subseteq V$. This condition guarantees that G is 2-connected (deleting one vertex leaves at most one component), but it does not guarantee a Hamiltonian cycle.

7.2.5. Example. The graph on the left below is bipartite with partite sets of equal size. However, it fails the necessary condition of Proposition 7.2.3. Hence it is not Hamiltonian.





The graph on the right shows that the necessary condition is not sufficient. This graph satisfies the condition but has no spanning cycle. All edges incident to vertices of degree 2 must be used, but in this graph that requires three edges incident to the central vertex.

The Petersen graph is another non-Hamiltonian graph satisfying the condition. We proved in Example 7.1.9 that $2C_5$ is the only 2-factor of the Petersen graph, so it has no spanning cycle. ■

7.2.6.* Remark. Strengthening a necessary condition may yield a sufficient condition. Perhaps requiring $|S| \geq 2c(G-S)$ for every cutset S would guarantee a spanning cycle. A graph G is **t-tough** if $|S| \geq tc(G-S)$ for every cutset $S \subset V$. The **toughness** of G is the maximum t such that G is t-tough. For example, the toughness of the Petersen graph is 4/3 (Exercise 23).

By Proposition 7.2.3, spanning cycles require toughness at least 1. Chvátal [1974] conjectured that a sufficiently large toughness is sufficient. No value of toughness larger than 1 is necessary, since C_n itself is only 1-tough. For some years it was thought that toughness 2 would be sufficient. Enomoto–Jackson–Katerinis–Saito [1985] constructed non-Hamiltonian graphs with toughness $2-\epsilon$ for each $\epsilon > 0$. Finally, Bauer–Broersma–Veldman [2000] constructed non-Hamiltonian graphs with toughness approaching 9/4. Chvátal's conjecture that some value of toughness suffices remains open. ■

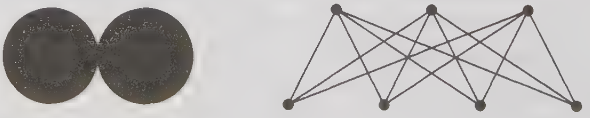
SUFFICIENT CONDITIONS

The number of edges needed to force an n-vertex graph to be Hamiltonian is quite large (Exercises 26–27). Under conditions that "spread out" the edges, we can reduce the number of edges while still guaranteeing Hamiltonian cycles. The simplest such condition is a lower bound on the minimum degree; $\delta(G) \geq n(G)/2$ suffices. We first note that no smaller minimum degree is sufficient.

7.2.7. Example. The graph consisting of cliques of orders $\lfloor (n+1)/2 \rfloor$ and $\lceil (n+1)/2 \rceil$ sharing a vertex has minimum degree $\lfloor (n-1)/2 \rfloor$ but is not Hamiltonian (not even 2-connected).

For odd order, another non-Hamiltonian graph with this minimum degree is the biclique with partite sets of sizes $(n-1)/2$ and $(n+1)/2$.

Proving that $\delta(G) \geq n(G)/2$ forces a spanning cycle thus shows that $\lfloor (n-1)/2 \rfloor$ is the largest value of the minimum degree among non-Hamiltonian graphs with n vertices. ■



7.2.8. Theorem. (Dirac [1952b]). If G is a simple graph with at least three vertices and $\delta(G) \geq n(G)/2$, then G is Hamiltonian.

Proof: The condition $n(G) \geq 3$ is annoying but must be included, since K_2 is not Hamiltonian but satisfies $\delta(K_2) = n(K_2)/2$.

The proof uses contradiction and extremality. If there is a non-Hamiltonian graph satisfying the hypotheses, then adding edges cannot reduce the minimum degree. Thus we may restrict our attention to maximal non-Hamiltonian graphs with minimum degree at least $n/2$, where "maximal" means that adding any edge joining nonadjacent vertices creates a spanning cycle.

When $u \not\leftrightarrow v$ in G, the maximality of G implies that G has a spanning path $v_1, \dots, v_n$ from $u = v_1$ to $v = v_n$, because every spanning cycle in $G + uv$ contains the new edge uv. To prove the theorem, it suffices to make a small change in this cycle to avoid using the edge uv; this will build a spanning cycle in G.

If a neighbor of u directly follows a neighbor of v on the path, such as $u \leftrightarrow v_{i+1}$ and $v \leftrightarrow v_i$, then $(u, v_{i+1}, v_{i+2}, \dots, v, v_i, v_{i-1}, \dots, v_2)$ is a spanning cycle.

To prove that such a cycle exists, we show that there is a common index in the sets S and T defined by $S = \{i\colon u \leftrightarrow v_{i+1}\}$ and $T = \{i\colon v \leftrightarrow v_i\}$. Summing the sizes of these sets yields

$$|S \cup T| + |S \cap T| = |S| + |T| = d(u) + d(v) \geq n.$$

Neither S nor T contains the index n. Thus $|S \cup T| < n$, and hence $|S \cap T| \geq 1$. We have established a contradiction by finding a spanning cycle in G; hence there is no (maximal) non-Hamiltonian graph satisfying the hypotheses. ■

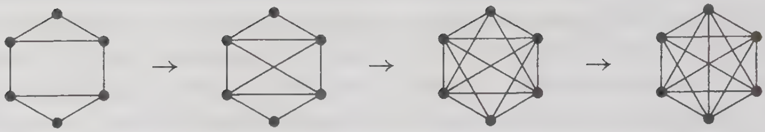
Ore observed that this argument uses $\delta(G) \geq n(G)/2$ only to show that $d(u) + d(v) \geq n$. Therefore, we can weaken the requirement of minimum degree $n/2$ to require only that $d(u) + d(v) \geq n$ whenever $u \not\leftrightarrow v$. We also did not need that G was a maximal non-Hamiltonian graph, only that $G + uv$ was Hamiltonian and thereby provided a spanning u, v-path.

7.2.9. Lemma. (Ore [1960]) Let G be a simple graph. If u, v are distinct nonadjacent vertices of G with $d(u) + d(v) \geq n(G)$, then G is Hamiltonian if and only if $G + uv$ is Hamiltonian.

Proof: One direction is trivial, and the proof of the other direction is the same as for Theorem 7.2.8. ■

Bondy and Chvátal [1976] phrased the essence of Ore's argument in a much more general form that yields sufficient conditions for cycles of length l and other subgraphs. Here we discuss only the application to spanning cycles. Using Lemma 7.2.9 to add edges, we can test whether G is Hamiltonian by testing whether the larger graph is Hamiltonian.

7.2.10. Definition. The **(Hamiltonian) closure** of a graph G, denoted $C(G)$, is the graph with vertex set $V(G)$ obtained from G by iteratively adding edges joining pairs of nonadjacent vertices whose degree sum is at least n, until no such pair remains.



The graph above begins with vertices of degree 2, but its closure is K_6. Ore's Lemma yields the following theorem.

7.2.11. Theorem. (Bondy–Chvátal [1976]) A simple n-vertex graph is Hamiltonian if and only if its closure is Hamiltonian. ■

Fortunately, the closure does not depend on the order in which we choose to add edges when more than one is available.

7.2.12. Lemma. The closure of G is well-defined.

Proof: Let $e_1, \dots, e_r$ and $f_1, \dots, f_s$ be sequences of edges added in forming $C(G)$, the first yielding G_1 and the second G_2. If in either sequence nonadjacent vertices u and v acquire degree summing to at least $n(G)$, then the edge uv must be added before the sequence ends.

Thus f_1, being initially addable to G, must belong to G_1. Similarly, if $f_1, \dots, f_{i-1} \in E(G_1)$, then f_i becomes addable to G_1 and therefore belongs to G_1. Hence neither sequence contains a first edge omitted by the other sequence, and we have $G_1 \subseteq G_2$ and $G_2 \subseteq G_1$. ■

We now have a necessary and sufficient condition to test for Hamiltonian cycles in simple graphs. It doesn't help much, because it requires us to test whether another graph is Hamiltonian! Nevertheless, it does furnish a method for proving sufficient conditions. A condition that forces $C(G)$ to be Hamiltonian also forces a Hamiltonian cycle in G.

For example, the condition may imply $C(G) = K_n$. Chvátal used this method to prove the best possible degree sequence condition for Hamiltonian cycles. Some vertex degrees can be small if others are large enough.

7.2.13. Theorem. (Chvátal [1972]) Let G be a simple graph with vertex degrees $d_1 \le \cdots \le d_n$, where $n \ge 3$. If $i < n/2$ implies that $d_i > i$ or $d_{n-i} \ge n-i$ (**Chvátal's condition**), then G is Hamiltonian.

Proof: Adding edges to form the closure reduces no entry in the degree sequence. Also, G is Hamiltonian if and only if $C(G)$ is Hamiltonian. Thus it suffices to consider the case where $C(G) = G$, which we describe by saying that G is *closed*. In this case, we prove that Chvátal's condition implies that $G = K_n$.

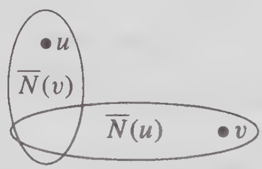
We prove the contrapositive; if G is a closed n-vertex graph that is not a complete graph, then we construct a value of i less than $n/2$ for which Chvátal's condition is violated. Violation means that at least i vertices have degree at most i and at least $n-i$ vertices have degree less than $n-i$.

With $G \neq K_n$, we choose among the pairs of nonadjacent vertices a pair u, v with maximum degree sum. Because G is closed, $u \not\leftrightarrow v$ implies that $d(u) + d(v) < n$. We choose the labels on u, v so that $d(u) \le d(v)$. Since $d(u)+d(v) < n$, we thus have $d(u) < n/2$. Let $i = d(u)$.

We need to find i vertices with degree at most i. Because we chose a nonadjacent pair with maximum degree sum, every vertex of $V - \{v\}$ that is not adjacent to v has degree at most $d(u)$, which equals i. There are $n - 1 - d(v)$ such vertices, and $d(u) + d(v) \le n - 1$ yields $n - 1 - d(v) \ge i$.

We also need $n - i$ vertices with degree less than $n - i$. Every vertex of $V - \{u\}$ that is not adjacent to u has degree at most $d(v)$, and we have $d(v) < n - d(u) = n - i$. There are $n - 1 - d(u)$ such vertices. Since $d(u) \le d(v)$, we can also add u itself to the set of vertices with degree at most $d(v)$. We thus obtain $n - i$ vertices with degree less than $n - i$.

We have proved that $d_i \le i$ and $d_{n-i} < n - i$ for this specially chosen i, which contradicts the hypothesis. ■

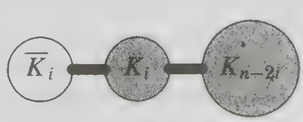


7.2.14. Example. *Non-Hamiltonian graphs with "large" vertex degrees.* Theorem 7.2.13 characterizes the degree sequences of simple graphs that force Hamiltonian cycles. If the degree sequence fails Chvátal's condition at i, then the largest we can make the terms in $d_1, \ldots, d_n$ is

$$\begin{aligned} d_j &= i && \text{for } j \le i,\\ d_j &= n - i - 1 && \text{for } i + 1 \le j \le n - i,\\ d_j &= n - 1 && \text{for } j > n - i. \end{aligned}$$

Let G be a simple graph realizing this degree sequence (if it exists). The i vertices of degree $n - 1$ are adjacent to all others (the central clique in the figure). This already gives i neighbors to the i vertices of degree i, so they form an independent set and have no additional neighbors. With degree $n - i - 1$, each of the remaining $n - 2i$ vertices must be adjacent to all vertices except itself and the independent set. Thus these vertices form a clique. The only possible realization is $(\overline{K}_i + K_{n-2i}) \vee K_i$, shown below.

This graph is not Hamiltonian, because deleting the i vertices of degree $n - 1$ leaves a subgraph with $i + 1$ components. If a simple graph H is non-Hamiltonian and has vertex degrees $d_1' \le \cdots \le d_n'$, then Chvátal's result implies that for some i the graph $(\overline{K}_i + K_{n-2i}) \vee K_i$ with vertex degrees $d_1 \le \cdots \le d_n$ satisfies $d_j \ge d_j'$ for all i. ■



7.2.15. Definition. A **Hamiltonian path** is a spanning path.

Every graph with a spanning cycle has a spanning path, but P_n shows that the converse is not true. We could make arguments like those above to prove sufficient conditions for Hamiltonian paths, but it is easier to use our previous work and prove the new theorem by invoking a theorem about cycles. To do this, we use a standard transformation.

7.2.16. Remark. A graph G has a spanning path if and only if the graph $G \vee K_1$ has a spanning cycle. ■

Remark 7.2.16 applies in several of the exercises. Here we use it to derive the analogue for paths of Chvátal's condition for spanning cycles.

7.2.17. Theorem. Let G be a simple graph with vertex degrees $d_1 \leq \cdots \leq d_n$. If $i < (n+1)/2$ implies ($d_i \geq i$ or $d_{n+1-i} \geq n-i$), then G has a spanning path.

Proof: Let $G' = G \vee K_1$, let $n' = n+1$, and let $d'_1, \ldots, d'_{n'}$ be the degree sequence of G'. Since a spanning cycle in $G \vee K_1$ becomes a spanning path in G when the extra vertex is deleted, it suffices to show that G' satisfies Chvátal's sufficient condition for Hamiltonian cycles.

Since the new vertex is adjacent to all of $V(G)$, we have $d'_{n'} = n$ and $d'_j = d_j + 1$ for $j < n'$. For $i < n'/2 = (n+1)/2$, the hypothesis on G yields

$$d'_i = d_i + 1 \geq i + 1 > i \quad \text{or} \quad d'_{n'-i} = d_{n+1-i} + 1 \geq n - i + 1 = n' - i.$$

This is precisely Chvátal's sufficient condition, so G' has a spanning cycle, and deleting the extra vertex leaves a spanning path in G. ■

7.2.18.* Remark. The degree requirements can be weakened under conditions such as regularity or high toughness. Every regular simple graph G with vertex degrees at least $n(G)/3$ is Hamiltonian (Jackson [1980]). Only the Petersen graph prevents lowering the threshold to $(n(G)-1)/3$ (Zhu–Liu–Yu [1985], partly simplified in Bondy–Kouider [1988]; see also Exercise 13).

It may be possible to lower the degree condition further when connectivity is high. For example, Tutte [1971] conjectured that every 3-connected 3-regular bipartite graph is Hamiltonian. Horton [1982] found a counterexample with 96 vertices, and the smallest known counterexample has 50 vertices (Georges [1989]), but stronger conditions of this sort may suffice. ■

Our last sufficient condition for Hamiltonian cycles involves connectivity and independence, not degrees. The proof yields a good algorithm that constructs a Hamiltonian cycle or shows that the hypothesis is false.

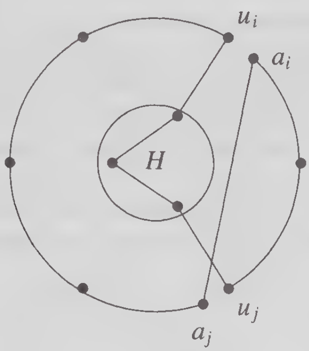
7.2.19. Theorem. (Chvátal–Erdős [1972]) If $\kappa(G) \geq \alpha(G)$, then G has a Hamiltonian cycle (unless $G = K_2$).

Proof: With $G \neq K_2$, the conditions require $\kappa(G) > 1$. Suppose that $\kappa(G) \geq \alpha(G)$. Let $k = \kappa(G)$, and let C be a longest cycle in G. Since $\delta(G) \geq \kappa(G)$, and every graph with $\delta(G) \geq 2$ has a cycle of length at least $\delta(G) + 1$ (Proposition 1.2.28), C has at least $k + 1$ vertices.

Let H be a component of $G - V(C)$. The cycle C has at least k vertices with edges to H; otherwise, deleting the vertices of C with edges to H contradicts $\kappa(G) = k$. Let $u_1, \ldots, u_k$ be k vertices of C with edges to H, in clockwise order.

For $i = 1, \ldots, k$, let a_i be the vertex immediately following u_i on C. If any two of these vertices are adjacent, say $a_i \leftrightarrow a_j$, then we construct a longer cycle by using $a_i a_j$, the portions of C from a_i to u_j and a_j to u_i, and a u_i, u_j-path through H (see illustration).

If a_i has a neighbor in H, then we can detour to H between u_i and a_i on C. Thus we also conclude that no a_i has a neighbor in H. Hence $\{a_1, \ldots, a_k\}$ plus a vertex of H forms an independent set of size $k + 1$. This contradiction implies that C is a Hamiltonian cycle. ■



7.2.20.* Remark. Most sufficient conditions for Hamiltonian cycles generalize to conditions for long cycles. The **circumference** of a graph is the length of its longest cycle. A weaker form of a sufficient condition for spanning cycles may force a long cycle. Dirac [1952b] proved the first such result: a 2-connected graph with minimum degree k has circumference at least $\min\{n, 2k\}$. Proposition 1.2.28 only guarantees a cycle of length at least $k + 1$. Most long-cycle results are more more difficult than the corresponding sufficient conditions for Hamiltonian cycles (see Lemma 8.4.36–Theorem 8.4.37). ■

CYCLES IN DIRECTED GRAPHS (optional)

The theory of cycles in digraphs is similar to that of cycles in graphs. For a digraph G, let $\delta^-(G) = \min d^-(v)$ and $\delta^+(G) = \min d^+(v)$. The arguments of Chapter 1 using maximal paths guarantee paths of length k and cycles of length $k+1$, where $k = \max\{\delta^-(G), \delta^+(G)\}$.

Every complete graph is Hamiltonian, but orientations of complete graphs are more complicated. The necessary condition of 2-connectedness becomes a necessary condition of strong connectedness for spanning cycles in digraphs. For tournaments, this necessary condition is also sufficient (Exercise 45).

For arbitrary digraphs, we prove an analogue of Dirac's theorem (Theorem 7.2.8). Indeed, it yields Dirac's theorem as a special case (Exercise 49). Meyniel [1973] subtantially strengthened the theorem by weakening the hypothesis (Theorem 8.4.42).

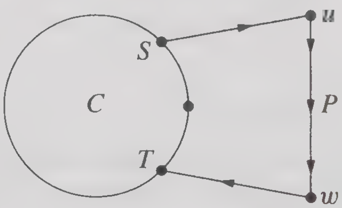
7.2.21. Definition. A digraph is **strict** if it has no loops and has at most one copy of each ordered pair as an edge.

7.2.22. Theorem. (Ghouilà-Houri [1960]) If D is a strict digraph, and $\min\{\delta^+(D), \delta^-(D)\} \geq n(D)/2$, then D is Hamiltonian.

Proof: Again we use contradiction and extremality. In an n-vertex counterexample D, let C be a longest cycle, with length l. As we have observed, $l > \max\{\delta^+, \delta^-\} \geq n/2$. Let P be a longest path in $D - V(C)$, beginning at u, ending at w, and having length $m \geq 0$. Now $l > n/2$ and $n \geq l + m + 1$ imply $m < n/2$.

Let S be the set of predecessors of u on C, and let T be the set of successors of w on C. By the maximality of P, every predecessor of u and successor of w lies in $V(C) \cup V(P)$. Thus S and T each have size at least $\min\{\delta^+, \delta^-\} - m$, which is at least $\geq n/2 - m$ and hence is positive. Thus S and T are nonempty.

The maximality of C guarantees that the distance along C from a vertex $u' \in S$ to a vertex $w' \in T$ must exceed $m + 1$. Otherwise, traveling along P instead of C from u' to w' yields a longer cycle. Hence we may assume that every vertex of S is followed on C by more than m vertices not in T.



If the distance between successive vertices of S along C is always at most $m + 1$, then there is no legal place to put a vertex of T. Since both S and T are nonempty, we may thus assume there is a vertex of S followed on C by at least $m + 1$ vertices not in S. These are forbidden from T, as is the immediate successor on C of all the other vertices of S.

Thus at least $|S| - 1 + m + 1 \geq n/2$ vertices of C are not in T. Together withthe vertices that are in T, this yields $|V(C)| \geq n - m$, which contradicts $l \leq n - m - 1$. The contradiction implies that C must be a spanning cycle. ∎

EXERCISES

7.2.1. (−) For which values of r is $K_{r,r}$ Hamiltonian?

7.2.2. (−) Is the Grötzsch graph (Example 5.2.2) Hamiltonian?

7.2.3. (−) For $n > 1$, prove that $K_{n,n}$ has $(n-1)!n!/2$ Hamiltonian cycles.

7.2.4. (−) Prove that G has a Hamiltonian path only if for every $S \subseteq V(G)$, the number of components of $G - S$ is at most $|S| + 1$.

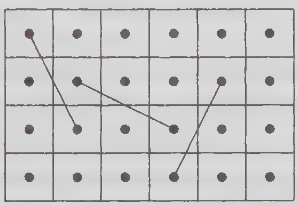
• • • • •

7.2.5. Prove that every 5-vertex path in the dodecahedron lies in a Hamiltonian cycle.

7.2.6. (!) Let G be a Hamiltonian bipartite graph, and choose $x, y \in V(G)$. Prove that $G - x - y$ has a perfect matching if and only if x and y are on opposite sides of the bipartition of G. Apply this to prove that deleting two unit squares from an 8 by 8 chessboard leaves a board that can be partitioned into 1 by 2 rectangles if and only if the two missing squares have opposite colors.

7.2.7. A mouse eats its way through a $3 \times 3 \times 3$ cube of cheese by eating all the $1 \times 1 \times 1$ subcubes. If it starts at a corner subcube and always moves on to an adjacent subcube (sharing a face of area 1), can it do this and eat the center subcube last? Give a method or prove impossible. (Ignore gravity.)

7.2.8. (!) On a chessboard, a **knight** can move from one square to another that differs by 1 in one coordinate and by 2 in the other coordinate, as shown below. Prove that no $4 \times n$ chessboard has a **knight's tour**: a traversal by knight's moves that visits each square once and returns to the start. (Hint: Find an appropriate set of vertices in the corresponding graph to violate the necessary condition.)



7.2.9. Construct an infinite family of non-Hamiltonian graphs satisfying the necessary condition of Proposition 7.2.3.

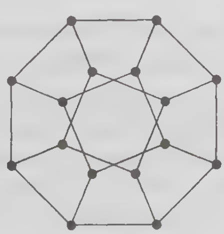
7.2.10. (!) *Hamiltonian vs. Eulerian.*

a) Find a 2-connected non–Eulerian graph whose line graph is Hamiltonian.

b) Prove that $L(G)$ is Hamiltonian if and only if G has a closed trail that contains at least one endpoint of each edge. (Harary and Nash-Williams [1965])

7.2.11. Construct a 3-regular 3-connected graph whose line graph is not Hamiltonian. (Hint: Replace each vertex in the Petersen graph with an appropriate graph and apply Exercise 7.2.10.)

7.2.12. Determine whether the graph below is Hamiltonian.

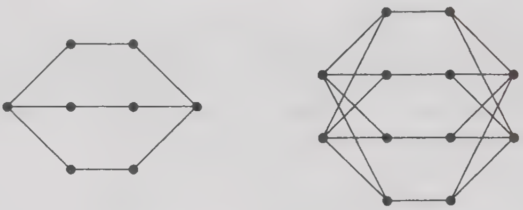


7.2.13. Let G be the 3-regular graph obtained from the Petersen graph by replacing one vertex with a triangle, matching the vertices of the triangle to the former neighbors of the deleted vertex. Prove that G is not Hamiltonian. (Comment: Except for this graph and the Petersen graph, every 2-connected, k-regular graph with at most $3k+3$ vertices is Hamiltonian.) (Hilbig [1986])

7.2.14. A graph G is **uniquely** k-edge-colorable if all proper k-edge-colorings of G induce the same partition of the edges. Prove that every uniquely 3-edge-colorable 3-regular graph is Hamiltonian. (Greenwell–Kronk [1973])

7.2.15. Place n points around a circle. Let G_n be the 4-regular graph obtained by joining each point to the nearest two points in each direction. If $n \geq 5$, prove that G_n is the union of two Hamiltonian cycles.

7.2.16. For $k \geq 3$, let G_k be the graph obtained from two disjoint copies of $K_{k,k-2}$ by adding a matching between the two "partite sets" of size k. Determine all values of k such that G_k is Hamiltonian.



7.2.17. (!) Prove that the cartesian product of two Hamiltonian graphs is Hamiltonian. Conclude that the k-dimensional cube Q_k is Hamiltonian for $k \geq 2$.

7.2.18. Prove that the cartesian product of two graphs with Hamiltonian paths fails to have a Hamiltonian cycle if and only if both graphs are bipartite and have odd order, in which case the product has a Hamiltonian path.

7.2.19. (+) For each odd natural number k, construct a $k-1$-connected k-regular simple bipartite graph that is not Hamiltonian.

7.2.20. (!) The kth **power** of a simple graph G is the simple graph G^k with vertex set $V(G)$ and edge set $\{uv\colon d_G(u, v) \leq k\}$.

a) Suppose that $G - x$ has at least three nontrivial components in each of which x has exactly one neighbor. Prove that G^2 is not Hamiltonian. (Hint: Consider the second graph in Example 7.2.5.)

b) Prove that the cube of each connected graph (with at least three vertices) is Hamiltonian. (Hint: Reduce this to the special case of trees, and prove it for trees by proving the stronger result that if xy is an edge of the tree T, then T^3 has a Hamiltonian cycle using the edge xy. Comment: Fleischner [1974] proved that the square of each 2-connected graph is Hamiltonian.)

7.2.21. Let $n = k(2l+1)$. Construct a non-Hamiltonian complete k-partite graph with n vertices and minimum degree $\frac{n}{2}\frac{k-1}{k}\frac{2l}{2l+1}$. (Snevily)

7.2.22. Let $\mathbf{G}(k, t)$ be the class of connected k-partite graphs in which each partite set has size t and each subgraph induced by two partite sets is a matching of size t. For $k \geq 4$ and $t \geq 4$, construct a graph in $\mathbf{G}(k, t)$ that is not Hamiltonian. (Hint: There is a graph in $\mathbf{G}(4, 4)$ with a 3-set whose deletion leaves four components; generalize this example. Comment: $\mathbf{G}(3, t) = \{C_{3t}\}$, and also every graph in $\mathbf{G}(k, 3)$ is Hamiltonian.) (Ayel [1982])

7.2.23. (∗) Prove that the Petersen graph has toughness 4/3.

7.2.24. (∗) Let $t(G)$ denote the toughness of G.

a) Prove that $t(G) \le \kappa(G)/2$. (Chvátal [1973])

b) Prove that equality holds in part (a) for claw-free graphs. (Hint: Consider a set S such that $|S| = t(G) \cdot c(G - S)$.) (Matthews–Sumner [1984])

7.2.25. (!) Let G be a simple graph that is not a forest and has girth at least 5. Prove that $\overline{G}$ is Hamiltonian. (Hint: Use Ore's condition.) (N. Graham)

7.2.26. (!) Prove that if G fails Chvátal's condition, then $\overline{G}$ has at least $n-2$ edges. Conclude from this that the maximum number of edges in a simple non-Hamiltonian n-vertex graph is $\binom{n-1}{2} + 1$. (Ore [1961], Bondy [1972b])

7.2.27. Prove directly by induction on n that the maximum number of edges in a simple non-Hamiltonian n-vertex graph is $\binom{n-1}{2} + 1$.

7.2.28. *Generalization of the edge bound.*

a) Let $f(i) = 2i^2 - i + (n-i)(n-i-1)$, and suppose that $n \ge 6k$. Prove that on the interval $k \le i \le n/2$, the maximum value of $f(i)$ is $f(k)$.

b) Let G be a simple graph with minimum degree k. Use part (a) and Chvátal's condition to prove that if G has at least $6k$ vertices and has more than $\binom{n(G)-k}{2} + k^2$ edges, then G is Hamiltonian. (Erdős [1962])

7.2.29. (!) Let G be a simple graph with vertex degrees $d_1 \le \cdots \le d_n$, and let $d'_1 \le \cdots \le d'_n$ be the vertex degrees in $\overline{G}$. Prove that if $d_i \ge d'_i$ for all $i \le n/2$, then G has a Hamiltonian path. Conclude that every simple graph isomorphic to its complement has a Hamiltonian path. (Clapham [1974])

7.2.30. Obtain Lemma 7.2.9 (sufficiency of Ore's condition) from Theorem 7.2.13 (sufficiency of Chvátal's condition). (Bondy [1978])

7.2.31. (!) Prove or disprove: If G is a simple graph with at least three vertices, and G has at least $\alpha(G)$ vertices of degree $n(G) - 1$, then G is Hamiltonian.

7.2.32. (+) Suppose that n is even and G is a simple bipartite graph with partite sets X, Y of size $n/2$. Let the vertex degrees of G be $d_1, \ldots, d_n$. Let G' be the supergraph of G obtained by adding edges so that $G[Y] = K_{n/2}$.

a) Prove that G is Hamiltonian if and only if G' is Hamiltonian, and describe the relationship between the degree sequences of G and G'.

b) Suppose that $d_k > k$ or $d_{n/2} > n/2 - k$ whenever $k \le n/4$. Prove that G is Hamiltonian. (Hint: Assume that the degree sequence of G' fails Chvátal's condition for some $i < n/2$, and obtain a contradiction.) (Chvátal [1972])

7.2.33. (!) A graph is **Hamiltonian-connected** if for every pair of vertices u, v there is a Hamiltonian path from u to v. Prove that a simple graph G is Hamiltonian if $e(G) \ge \binom{n(G)-1}{2} + 2$ and Hamiltonian-connected if $e(G) \ge \binom{n(G)-1}{2} + 3$. (Proving the two together permits a simpler proof.) (Ore [1963])

7.2.34. *Necessary condition for Hamiltonian-connected.* (Moon [1965a])

a) Prove that every Hamiltonian-connected graph G with at least four vertices has at least $\lceil 3n(G)/2 \rceil$ edges.

b) Prove that the bound in part (a) is best possible by showing that $C_m \,\square\, K_2$ is Hamiltonian-connected if m is odd.

7.2.35. (!) *Sufficient condition for Hamiltonian-connected.* (Ore [1963])

a) Prove that a simple graph G is Hamiltonian-connected if $x \not\leftrightarrow y$ implies $d(x) +$

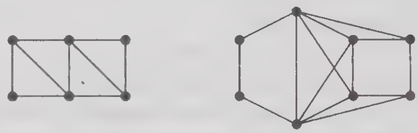
$d(y) > n(G)$. (Hint: Prove that appropriate graphs related to G are Hamiltonian by considering their closures.)

b) Prove that part (a) is sharp by constructing, for each even n greater than 2, a simple n-vertex graph with minimum degree $n/2$ that is not Hamiltonian-connected.

7.2.36. Las Vergnas' condition for a simple n-vertex graph is the existence of a vertex ordering $v_1, \ldots, v_n$ such that there is no nonadjacent pair v_i, v_j satisfying $i < j, d(v_i) \le i$, $d(v_j) < j, d(v_i) + d(v_j) < n$, and $i + j \ge n$. Las Vergnas [1971] proved that this condition is sufficient for the existence of a spanning cycle.

a) Prove that Chvátal's condition (Theorem 7.2.13) implies Las Vergnas' condition, which means that Las Vergnas' theorem strengthens Chvátal's theorem.

b) Prove that each of the graphs below fails Chvátal's condition but has a complete graph as its Hamiltonian closure. Prove that the smaller graph satisfies Las Vergnas' condition but the larger one does not.



7.2.37. For $\emptyset \ne S \subset V(G)$, let $t(S) = \left|\overline{S} \cap N(S)\right| / \left|\overline{S}\right|$. Let $\theta(G) = \min t(S)$. Lu [1994] proved that if $\theta(G)n(G) \ge \alpha(G)$, then G is Hamiltonian. Prove that $\kappa(G) \ge \alpha(G)$ implies $\theta(G)n(G) \ge \alpha(G)$. (Comment: This shows that Lu's theorem implies the Chvátal–Erdős Theorem and is a stronger result.)

7.2.38. (!) *Long paths and cycles.* Let G be a connected simple graph with $\delta(G) = k \ge 2$ and $n(G) > 2k$.

a) Let P be a maximal path in G (not a subgraph of any longer path). If $n(P) \le 2k$, prove that the induced subgraph $G[V(P)]$ has a spanning cycle (this cycle need not have its vertices in the same order as P).

b) Use part (a) to prove that G has a path with at least $2k + 1$ vertices. Give an example for each odd value of n to show that G need not have a cycle with more than $k + 1$ vertices.

7.2.39. Prove that if a simple graph G has degree sequence $d_1 \le \cdots \le d_n$ and $d_1 + d_2 < n$, then G has a path of length at least $d_1 + d_2 + 1$ unless G is the join of $n - (d_1 + 1)$ isolated vertices with a graph on $d_1 + 1$ vertices or $G = pK_{d_1} \vee K_1$ for some $p \ge 3$. (Ore [1967b])

7.2.40. (!) Dirac [1952b] proved that every 2-connected simple graph G has a cycle of length at least $\min\{n(G), 2\delta(G)\}$. Use this to prove that every $2k$-regular graph with $4k + 1$ vertices is Hamiltonian. (Nash-Williams)

7.2.41. Scott Smith conjectured that any two longest cycles in a k-connected graph have at least k common vertices. The approach below works for small k.

a) Suppose that G is a 4-regular graph with n vertices that is the union of two cycles (multiple edges may arise). Let G' be the 4-regular graph on $n+2$ vertices obtained from G by subdividing two edges and adding a double edge between the two new vertices. Show that G' is also the union of two spanning cycles if $n \le 5$.

b) Use part (a) to conclude that any pair of longest cycles in a k-connected graph intersect in at least k points if $k \le 6$. (Smith, Burr)

7.2.42. (+) Let G be an Eulerian graph. Let V' be the set of Eulerian circuits of G, considering a circuit and its reversal to be the same. Let G' be the graph with vertex set V' such that two circuits are adjacent if and only if one arises from the other by reversing the edge order on a proper closed subcircuit. Prove that G' is Hamiltonian if $\Delta(G) \leq 4$. (Hint: Use induction on the number of vertices of degree 4, proving that there is a Hamiltonian cycle through every edge of G'. Comment: The conclusion also holds without restriction on $\Delta(G)$.) (Xia [1982], Zhang–Guo [1986])

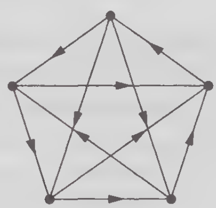
7.2.43. Prove that the Eulerian circuit graph G' of Exercise 7.2.42 is regular, and derive a formula for its vertex degree. Compare $\delta(G')$ and $n(G')$ when $n(G) = 2$ to show that the preceding problem cannot be solved by applying general results on Hamiltonicity of regular graphs with specified degree.

7.2.44. Prove that every tournament has a Hamiltonian path (a spanning directed path). (Hint: Use extremality). (Rédei [1934])

7.2.45. Let T be a strong tournament. For each $u \in V(T)$ and each k such that $3 \leq k \leq n$, prove that u belongs to a cycle of length k in T. (Hint: Use induction on k.) (Moon [1966])

7.2.46. Let G be a 7-vertex tournament in which every vertex has outdegree 3. Use Exercise 7.2.45 to prove that G has two vertex-disjoint cycles.

7.2.47. (+) Prove that every tournament has a Hamiltonian path that is not contained in a Hamiltonian cycle, except the cyclic tournament on three vertices and the tournament T_5 on five vertices drawn below. (Hint: Induction works, but some care is needed to prove the claim for six vertices. In all cases, find the desired configuration or $G = T_5$.) (Grünbaum, in Harary [1969, p211])



7.2.48. (∗) Prove that Theorem 7.2.22 is best possible by showing that the strictness condition on the digraph cannot be weakened to allow loops. In particular, construct for each even n an n-vertex digraph D that is not Hamiltonian even though at most one copy of each ordered pair is an edge and $\min\{\delta^-(D), \delta^+(D)\} \geq n/2$.

7.2.49. (∗) Obtain Theorem 7.2.8 (sufficiency of Dirac's condition in graphs) from Theorem 7.2.22 (sufficiency of Ghouilà-Houri's condition on digraphs). (Hint: Transform a simple graph G into a strict digraph by replacing each edge with a pair of directed edges in opposite directions.)

7.3. Planarity, Colorings, and Cycles

We return to the Four Color Problem to explore its historical relationship with the problems of edge-coloring and Hamiltonian cycles. We then consider ways in which the problem generalizes.

TAIT'S THEOREM

In 1878, Tait proved a theorem relating face-coloring and edge-coloring of plane graphs, and he used this in an approach to the Four Color Theorem. This stimulated interest in edge-coloring. We first define face-coloring precisely.

7.3.1. Definition. A **proper face-coloring** of a 2-edge-connected plane graph is an assignment of colors to its faces so that faces having a common edge in their boundaries have distinct colors.

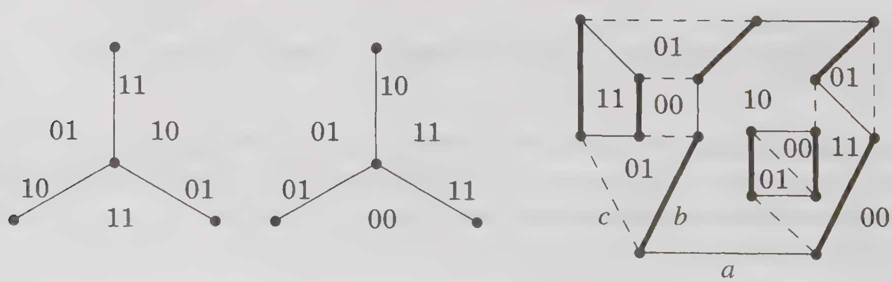
We often think of a face-coloring as a coloring of the dual graph. For this reason, we restrict our attention to face-colorings of 2-edge-connected graphs. When a plane graph has a cut-edge, its dual has a loop. We say that graphs with loops do not have proper colorings. In a plane graph with a cut-edge, a face shares a boundary with itself and is thus uncolorable.

Since adding edges does not make ordinary coloring easier, to prove the Four Color Theorem it suffices to prove that all triangulations are 4-colorable. Equivalently, we could show that all duals of triangulations are 4-face-colorable. The dual G^* of a plane triangulation G is a 3-regular, 2-edge-connected plane graph (Exercise 6.1.11). Tait showed that for such graphs, proper 4-face-colorings are equivalent to proper 3-edge-colorings.

7.3.2. Theorem. (Tait [1878]) A simple 2-edge-connected 3-regular plane graph is 3-edge-colorable if and only if it is 4-face-colorable.

Proof: Let G be such a graph. Suppose first that G is 4-face-colorable; we obtain a 3-edge-coloring. Let the four colors be denoted by binary ordered pairs: $c_0 = 00$, $c_1 = 01$, $c_2 = 10$, $c_3 = 11$. Color $E(G)$ by assigning to the edge between faces with colors c_i and c_j the color obtained by adding c_i and c_j coordinatewise using addition modulo 2. (Thus $c_2 + c_3 = c_1$, for example.) We show that this is a proper 3-edge-coloring.

Because G is 2-edge-connected, each edge bounds two distinct faces. Hence the color 00 never occurs as a sum. We check that the edges at a vertex receive distinct colors. At vertex v the faces bordering the three incident edges must have distinct colors $\{c_i, c_j, c_k\}$, as illustrated below. If color 00 is not in this set, then the sum of any two of these is the third, and hence $\{c_i, c_j, c_k\}$ is also the set of colors on the edges. If $c_k = 00$, then c_i and c_j appear on two of the edges, and the third receives color $c_i + c_j$, which is the color not in $\{c_i, c_j, c_k\}$.

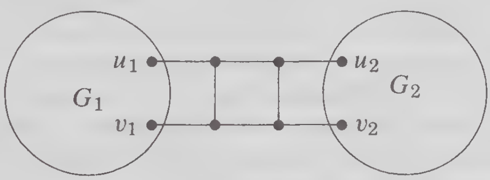


For the converse, suppose that G has a proper 3-edge-coloring using colors a, b, c (shown bold, solid, and dashed). Let E_a, E_b, E_c be the edge sets having the three colors, respectively. We construct a 4-face-coloring using the four colors defined above. Since G is 3-regular, each color appears at every vertex, and the union of any two of E_a, E_b, E_c is 2-regular, which makes it a union of disjoint cycles. Each face of this subgraph is a union of faces of the original graph. Let $H_1 = E_a \cup E_b$ and $H_2 = E_b \cup E_c$. To each face of G, assign the color whose ith coordinate ($i \in \{1, 2\}$) is the parity of the number of cycles in H_i that contain it (0 for even, 1 for odd).

We claim that this is a proper 4-face-coloring, as illustrated above. Faces F, F' sharing an edge e are distinct faces, since G is 2-edge-connected. Edge e belongs to a cycle C in at least one of H_1, H_2 (in both if e has color b). By the Jordan Curve Theorem, one of F, F' is inside C and the other is outside. All other cycles in H_1 and H_2 fail to separate F and F', leaving them on the same side. Hence if e has color a, c, or b, then the parity of the number of cycles containing F and F' is different in H_1, in H_2, or in *both*, respectively. Thus F and F' receive different colors in the face-coloring we have constructed. ■

Due to this theorem, a proper 3-edge-coloring of a 3-regular graph is called a **Tait coloring**. The problem of showing that every 2-edge-connected 3-regular planar graph is 3-edge-colorable reduces to showing that every 3-connected 3-regular planar graph is 3-edge-colorable.

7.3.3.* Lemma. If G is a 3-regular graph with edge-connectivity 2, then G has subgraphs G_1, G_2 and vertices $u_1, v_1 \in V(G_1)$ and $u_2, v_2 \in V(G_2)$ such that $u_1 \not\leftrightarrow v_1$, also $u_2 \not\leftrightarrow v_2$, and G consists of G_1, G_2 and a *ladder* of some length joining G_1, G_2 at u_1, v_1, u_2, v_2 as shown below.



Proof: If G has an edge cut of size 2 in which the two edges are incident, then the third edge incident to their common vertex is a cut-edge, contradicting $\kappa' = 2$. Hence we may assume that the four endpoints in our minimum edge cut xy, uv are distinct. If $x \not\leftrightarrow y$ and $u \not\leftrightarrow v$, then these are the four desired vertices and the ladder has only these two edges.

When $x \leftrightarrow y$, we extend the ladder (a similar argument applies when $u \leftrightarrow v$). Let w be the third neighbor of x and z the third neighbor of y. If $w = z$, then the third edge incident to this vertex is a cut-edge. Hence $w \neq z$ and the ladder extends. If $w \not\leftrightarrow z$, then we are finished in this direction; otherwise, we repeat the argument till we obtain a nonadjacent pair at the base of the ladder. ■

7.3.4.* Theorem. All 2-edge-connected 3-regular simple planar graphs are 3-edge-colorable if and only if all 3-connected 3-regular simple planar graphs are 3-edge-colorable.

Proof: The second family is contained in the first. Hence it suffices to show that 3-edge-colorability for all graphs in the smaller family implies it also for the larger family. We use induction on $n(G)$.

Basis step ($n(G) = 4$): The only 2-edge-connected 3-regular simple planar graph with at most 4 vertices is K_4, which is 3-edge-colorable.

Induction step ($n(G) > 4$): Since $\kappa(G) = \kappa'(G)$ when G is 3-regular (Theorem 4.1.11), we may restrict our attention to 3-regular graphs with edge-connectivity 2. Lemma 7.3.3 gives us a decomposition of G into G_1, G_2, and a ladder joining them. The *length* of the ladder is the distance from G_1 to G_2.

Both $G_1 + u_1v_1$ and $G_2 + u_2v_2$ are 2-edge-connected and 3-regular. By the induction hypothesis, they are 3-edge-colorable; let f_i be a proper 3-edge-coloring of $G_i + u_iv_i$. Permute names of colors so that $f_1(u_1v_1) = 1$ and so that $f_2(u_2v_2)$ is chosen from $\{1, 2\}$ to have the same parity as the length of the ladder.

Returning to G, color each G_i as in f_i. Beginning from the end of the ladder at G_1, color the rungs of the ladder with 3, and color the paths forming the sides of the ladder alternately with 1 and 2. The edges of the ladder at u_i and v_i now have the color $f_i(u_iv_i)$. Thus we have assembled a proper 3-edge-coloring of G. ■

Thus the Four Color Theorem reduces to finding Tait colorings of 3-edge-connected 3-regular planar graphs. The statement of their existence was known as **Tait's conjecture** and is equivalent to the Four Color Theorem.

GRINBERG'S THEOREM

Every Hamiltonian 3-regular graph has a Tait coloring (Exercise 1). Tait believed that this completed a proof of the Four Color Theorem, because he assumed that every 3-connected 3-regular planar graph is Hamiltonian. Not until 1946 was an explicit counterexample found, although the gap in the proof was noticed earlier. Later, Grinberg [1968] discovered a simple necessary condition that led to many 3-regular 3-connected non-Hamiltonian planar graphs, including the Grinberg graph of Exercise 16.

7.3.5. Theorem. (Grinberg [1968]) If G is a loopless plane graph having a Hamiltonian cycle C, and G has f_i' faces of length i inside C and f_i'' faces of length i outside C, then $\sum_i (i-2)(f_i' - f_i'') = 0$.

Proof: Considering the faces inside and outside C separately, we want to show that $\sum_i (i-2) f_i' = \sum_i (i-2) f_i''$. No changes on one side affect the sum on the other side. Furthermore, we can switch inside and outside by projecting the embedding onto a sphere and puncturing a face inside C.

Hence we need only show that $\sum_i (i-2) f_i'$ is constant. When there are no inside edges, the sum is $n-2$. With this as the basis step, we prove by induction on the number of inside edges that the sum is always $n-2$.

Suppose that $\sum_i (i-2) f_i' = n-2$ when there are k edges inside C. We can obtain any graph with $k+1$ edges inside C by adding an edge to such a graph.

The added edge cuts a face of some length r into two faces of lengths s and t. We have $s + t = r + 2$, because the new edge contributes to both new faces and each edge on the old face contributes to one new face.

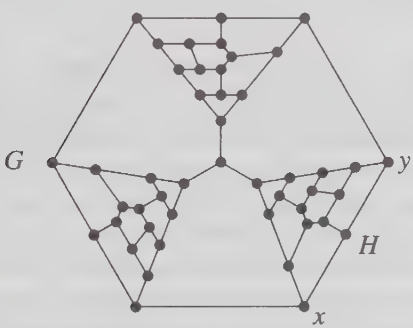
No other contribution to the sum changes. Since $(s-2)+(t-2)=(r-2)$, the contribution from these faces also remains the same. By the induction hypothesis, the sum is $n-2$. ■

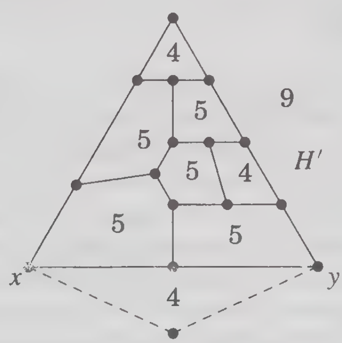
Being a necessary condition, Grinberg's condition can be used to show that graphs are *not* Hamiltonian. The arguments can often be simplified using modular arithmetic. Two numbers that are not congruent mod k are not equal.

We apply this to the first known non-Hamiltonian 3-connected 3-regular planar graph (Tutte [1946]). Tutte used an *ad hoc* argument to prove that this graph is not Hamiltonian. For many years it was the only known example (see Exercise 17 for the smallest now known).

7.3.6. Example. *Grinberg's condition and the Tutte graph.* The Tutte graph G appears on the left below. Let H denote each component obtained by deleting the central vertex and the three long edges. Since a Hamiltonian cycle must visit the central vertex of G, it must traverse one copy of H along a Hamiltonian path joining the other entrances to H, which we call x and y.

We therefore study a graph that has a Hamiltonian cycle if and only if H has a Hamiltonian x, y-path. Such a graph H' (on the right below) is obtained by adding an x, y-path of length two through a new vertex.





The plane graph H' has five 5-faces, three 4-faces, and one 9-face. Grinberg's condition becomes $2a_4 + 3a_5 + 7a_9 = 0$, where $a_i = f_i' - f_i''$. Since the unbounded face is always outside, the equation reduces mod 3 to $2a_4 \equiv 7 \bmod 3$. Since $f_4' + f_4'' = 3$, the possibilities for a_4 are $+3, +1, -1, -3$. The only choice satisfying $2a_4 \equiv 7 \bmod 3$ is $a_4 = -1$, which requires that two of the 4-faces lie outside the Hamiltonian cycle. However, the 4-faces having having a vertex of degree 2 cannot lie outside the cycle, since the edges incident to the vertex of degree 2 separate the face from the outside face.

We can reach a contradiction faster by subdividing one edge incident to each vertex of degree 2. This does not change the existence of a spanning cycle. The resulting graph has seven 5-faces, one 4-face, and one 11-face. The required equation becomes $2 \cdot (\pm 1) = 9 - 3a_5$, which has no solution since the left side is not a multiple of 3. ■

We have not presented a systematic procedure for proving the nonexistence of solutions to equations with integer variables. Our arguments involving divisibility are merely tricks to avoid listing cases, but such tricks often work.

High connectivity makes it harder to avoid spanning cycles. Tutte [1956] (extended by Thomassen [1983]) proved that every 4-connected planar graph is Hamiltonian. Barnette [1969] conjectured that every planar 3-connected 3-regular bipartite graph is Hamiltonian.

SNARKS (optional)

Another approach to the Four Color Theorem is to study which 3-regular graphs are 3-edge-colorable. In a discussion focusing on 3-regular graphs and graphs without cut-edges, it is convenient to have simple adjectives to describe these properties.

7.3.7. Definition. A **bridgeless graph** is a graph without cut-edges. A **cubic graph** is a graph that is regular of degree 3.

7.3.8. Conjecture. (3-edge-coloring Conjecture—Tutte [1967]) Every bridgeless cubic non-3-edge-colorable graph contains a subdivision of the Petersen graph.

Conjecture 7.3.8 has been proved! Like the Four Color Theorem, its computer-assisted proof uses discharging methods. The proof will appear in a series of five papers by Robertson, Sanders, Seymour, and Thomas [2001].

Since every subdivision of the Petersen graph is nonplanar, Conjecture 7.3.8 implies Tait's Conjecture and hence the Four Color Theorem. One natural approach to the conjecture, like the idea of reducibility for the Four Color Theorem, is to derive properties that a minimal counterexample must have. In this language, Theorem 7.3.4 says that a minimal counterexample must be 3-edge-connected. In the next lemma, we make this statement precise and obtain several other properties.

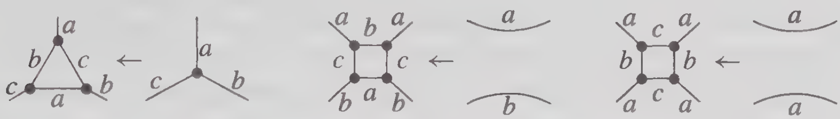
7.3.9. Definition. A **trivial edge cut** is an edge cut whose deletion isolates a single vertex. Other edge cuts are **nontrivial**.

7.3.10. Lemma. If a non-3-edge-colorable cubic graph G has connectivity 2 or girth less than 4 or a nontrivial 3-edge cut, then G contains a subdivision of a smaller non-3-edge-colorable cubic graph.

Proof: Suppose first that G has an edge cut of size 2. As discussed in Lemma 7.3.3, these edges have no common vertices. Deleting the edge cut and adding one edge to each piece yields cubic graphs $G_1 + u_1v_1$ and $G_2 + u_2v_2$. As argued in Theorem 7.3.4, at least one of these graphs is not 3-edge-colorable. Since the added edge can be replaced by a path through the other piece, G contains a subdivision of this smaller non-3-edge-colorable graph.

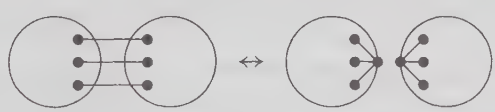
Next suppose that G contains a triangle. Let G' be the graph obtained from G by contracting the triangle to a single vertex. A proper 3-edge-coloring of G' could be expanded into a proper 3-edge-coloring of G as shown below. Also, G contains a subdivision of G', obtained by deleting one edge of the triangle.

Suppose that G contains a 4-cycle but no triangle. Let G' be the cubic graph obtained from G by deleting two opposite edges of the 4-cycle and replacing the resulting paths of length 3 with single edges. Since G has no triangle, the new edges are not loops. A proper 3-edge-coloring of G' yields a proper 3-edge-coloring of G via the two cases shown below. Also G contains a subdivision of G', so G' is the desired smaller graph.



Finally, suppose that G contains a nontrivial 3-edge cut $[S, \overline{S}]$. Since we may assume that G is 3-edge-connected, the three edges of the cut are pairwise disjoint. The two graphs obtained by contracting $G[S]$ or $G[\overline{S}]$ to a single vertex are also 3-regular. If both are 3-edge-colorable, then the colors can be renamed to agree on the edges of the cut, yielding a proper 3-edge-coloring of G. Thus at least one of these graphs is not 3-edge-colorable.

It remains only to show that G contains a subdivision of $G[S]$ (and similarly of $G[\overline{S}]$). Let a, b, c be the endpoints in $\overline{S}$ of the edges in the cut. Since G is 3-edge-connected, the cut is a bond, and $G[\overline{S}]$ is connected (Proposition 4.1.15). Thus $G[\overline{S}]$ contains an a, b-path P and a path from c to P. Adding these paths and the edges of the cut to $G[S]$ completes a subdivision of $G[S]$. ■



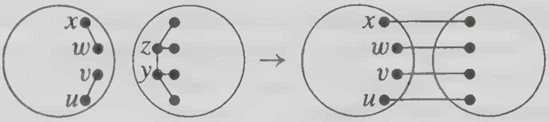
7.3.11. Definition. A **snark** is a 2-edge-connected 3-regular graph that is not 3-colorable, has girth at least 5, and has no non-trivial 3-edge cut. A **prime snark** is one that contains no subdivision of a smaller snark.

In this language, we have reduced Tutte's 3-edge-coloring Conjecture to the statement that the Petersen graph is the only prime snark. Again, we note that the conjecture has been proved (Robertson–Sanders–Seymour–Thomas [2001]).

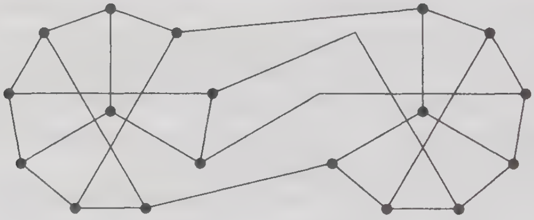
After the Petersen graph in 1898, by 1975 only three more snarks had been found: the 18-vertex Blanuša [1946] snark, the 210-vertex Descartes [1948] snark, and the 50-vertex Szekeres [1973] snark. This prompted Martin Gardner [1976] to invent the term "snark", evoking the rarity of the creature in Lewis Carroll's "The Hunting of the Snark".

Isaacs [1975] then showed that the earlier snarks arise from the Petersen graph via an operation that generates infinite families of snarks.

7.3.12. Definition. The **dot product** of cubic graphs G and H is the cubic graph formed from $G + H$ by deleting disjoint edges uv and wx from G, deleting adjacent vertices y and z from H, and adding edges from u and v to $N_H(y) - \{z\}$ and from w and x to $N_H(z) - \{y\}$.

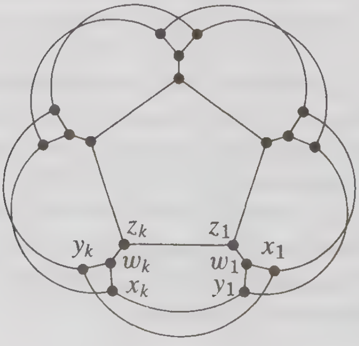


The dot product of two snarks is a snark (Exercise 23). Applying it to two copies of the Petersen graph yields the Blanuša snark shown below. This graph has a non-trivial 4-edge cut. Kochol [1996] introduced a more general operation that yields snarks with large girth and higher connectedness properties.



7.3.13. Example. *The flower snarks.* Isaacs also found an explicit infinite family of snarks (Exercise 21) that don't arise via the dot product. Independently discovered by Grinberg, they have $4k$ vertices, for odd $k \geq 5$.

Begin with three disjoint k-cycles. Let $\{x_i\}$, $\{y_i\}$, $\{z_i\}$ be the three vertex sets, indexed cyclically. For each i add a vertex w_i with $N(w_i) = \{x_i, y_i, z_i\}$. The resulting graph G_k is 3-edge-colorable. Let H_k be the graph obtained by replacing the edges $x_k x_1$ and $y_k y_1$ with $x_k y_1$ and $y_k x_1$. If k is odd and $k \geq 5$, then H_k is a snark. If k is even, then H_k is 3-edge-colorable. The drawing of H_k in which $\{z_i\}$ is a central cycle suggests the name "flower snark". ■



FLOWS AND CYCLE COVERS (optional)

Tait's Theorem (Theorem 7.3.2) states that 3-edge-colorability and 4-face-colorability are equivalent for plane triangulations. When extending this beyond planar graphs, we need a concept that makes sense for all graphs and is equivalent to 4-face-coloring on plane graphs. Additional information about this topic (and about snarks) appears in the monograph by Zhang [1997].

7.3.14. Definition. A **flow** on a graph G is a pair (D, f) such that

1) D is an orientation of G,
2) f is a weight function on $E(G)$, and
3) each $v \in V(G)$ satisfies $\sum_{w \in N_D^+(v)} f(vw) = \sum_{u \in N_D^-(v)} f(uv)$.

A k-**flow** is an integer-valued flow such that $|f(e)| \le k-1$ for all $e \in E(G)$. A flow is **nowhere-zero** or **positive** if $f(e)$ is nonzero or positive, respectively, for all $e \in E(G)$.

The usage of "flow" here is somewhat different from that in Chapter 4. In both contexts, the word "flow" suggests the conservation constraints imposed at each vertex. The bound of $k-1$ on flow value evokes the notion of capacity.

We can alter the orientation to make all weights positive.

7.3.15. Proposition. For a graph G, the following are equivalent:

A) G has a positive k-flow.
B) G has a nowhere-zero k-flow.
C) G has a nowhere-zero k-flow for each orientation of G.

Proof: Simultaneously changing the orientation of an edge and the sign of its weight does not affect the conservation constraints. ■

Thus the existence of a nowhere-zero k-flow does not depend on the choice of the orientation. We can also take linear combinations of flows.

7.3.16. Proposition. If $(D, f_1), \dots, (D, f_r)$ are flows on G, and $g = \sum_{i=1}^{r} \alpha_i f_i$, then (D, g) is a flow on G.

Proof: For each $v \in V(G)$, the net flow out of v under each f_i is zero, and hence it is also zero under g. ■

7.3.17. Proposition. For a flow on G, the net flow out of any set $S \subseteq V(G)$ is zero. Thus a graph with a nowhere-zero flow has no cut-edge.

Proof: We sum the net flows out of vertices of S. Edges leaving S contribute with positive weight, edges entering S contribute with negative weight, and edges within S contribute positively at their tails and negatively at their heads. The net flow out of S is thus the sum of the net flows out of the vertices of S, which is zero.

This implies that the net flow across any edge cut is zero, so it cannot consist of a single edge with nonzero weight. ■

Thus we restrict our attention to graphs without cut-edges (bridgeless graphs). What distinguishes flows here from circulations in Section 4.3 is that we forbid zero as a weight. Nowhere-zero flows enable us to extend Tait's Theorem. We begin by interpreting Eulerian graphs in the context of nowhere-zero flows; connectedness is no longer important.

7.3.18. Definition. A graph is an **even graph** if every vertex has even degree.

7.3.19. Proposition. A graph has a nowhere-zero 2-flow if and only if it is an even graph.

Proof: Given a nowhere-zero 2-flow, we obtain a positive 2-flow. Since this assigns weight 1 to every edge, the orientation must have as many edges entering each vertex as leaving it. Thus each vertex degree is even.

Conversely, when each vertex degree is even, each component has an Eulerian circuit. Orienting the edges to follow such a circuit and assigning weight 1 to each edge yields a positive 2-flow. ■

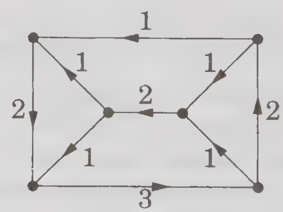
Nowhere-zero 3-flows are more subtle, even for 3-regular graphs.

7.3.20. Proposition. (Tutte [1949]) A cubic graph has a nowhere-zero 3-flow if and only if it is bipartite.

Proof: Let G be a cubic X, Y-bigraph. Every regular bipartite graph has a 1-factor. Orient the edges of a 1-factor from X to Y, and give them weight 2. Orient all other edges from Y to X, and give them weight 1. The flow in and out of every vertex is 2, so this is a nowhere-zero 3-flow.

Conversely, let G be a cubic graph with a nowhere-zero 3-flow. By Proposition 7.3.15, we may assume that the flow is 1 or 2 on each edge. Since the net flow is 0, there must be one edge with flow 2 and two edges with flow 1 at each vertex. Thus the edges with flow 2 form a matching. The X be the set of tails and Y the set of heads of these edges. Since the net flow is 0 at each vertex, each edge with flow 2 points from X to Y, and each edge with flow 1 points from Y to X. Thus X, Y is a bipartition of G. ■

7.3.21. Example. Since the Petersen graph is cubic and not bipartite, it has no nowhere-zero 3-flow. We will see that it also has no nowhere-zero 4-flow. Below we show a nowhere-zero 4-flow in the 3-regular simple graph $C_3 \square K_2$. ■

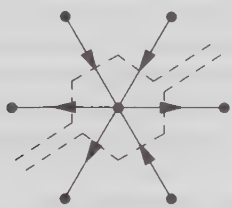


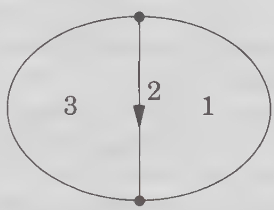
To understand the duality between flows and colorings, we characterize the plane graphs with nowhere-zero k-flows.

7.3.22. Theorem. (Tutte [1954b]) A plane bridgeless graph is k-face-colorable if and only if it has a nowhere-zero k-flow.

Proof: (Younger [1983], refined by Seymour) Let f be a flow on a plane graph G. We define a function g on the set of faces by letting $g(F)$ be the net flow accumulated by traveling from face F out to the unbounded face. Each time we cross an edge e we count $+f(e)$ if e is directed toward our right, $-f(e)$ if e is directed toward our left. The value assigned to the outside face is 0.

The function g is well-defined; that is, $g(F)$ is independent of our route to the outside face. We can change a route into any other by a succession of changes where we go the "other way" around some vertex v (shown on the left below). The change increases or decreases our accumulation for this portion by the net flow out of v, which is 0. Note that the difference between the values on faces with a common edge e is $\pm f(e)$.





Conversely, given a function g defined on the faces, we can invert the process to obtain a flow (shown on the right above). As we stand on face F and look and face F' across edge e, we let $f(e) = g(F) - g(F')$ if e is directed toward our right, $f(e) = g(F') - g(F)$ if e is directed toward our left.

Thus flows correspond to face-colorings. The face-coloring is proper if and only if the flow is nowhere-zero. If the flow is a nowhere-zero k-flow, then reducing the labels in the coloring to congruence classes in $\{0, \ldots, k-1\}$ produces a proper k-coloring. Conversely, a proper k-face-coloring using these colors produces a nowhere-zero k-flow. ■

The correspondence between face-labelings and flows in Theorem 7.3.22 is valid when the labels come from any abelian group. Applied using the group of binary ordered pairs under addition ((0, 0) is the identity), the statement proved by this argument is precisely Tait's Theorem itself.

Since we can study flows on all graphs, we can consider the flow problem as a general dual notion to vertex coloring. "Nowhere-zero" is the analogue of "proper". Since every nowhere-zero k-flow *is* a nowhere-zero $k+1$-flow, the natural problem is to minimize k such that G has a nowhere-zero k-flow. This minimum is the **flow number** of G, by analogy with "chromatic number". Since we say "G is k-colorable" when G has a proper k-coloring, the natural analogue would be to say "G is k-**flowable**" instead of "G has a nowhere-zero k-flow". This language is not yet common, so we will use it sparingly.

By Tait's Theorem, Theorem 7.3.22 states that a cubic bridgeless planar graph is 3-edge-colorable if and only if it has a nowhere-zero 4-flow. We want to extend this correspondence by dropping the condition on planarity. A simple observation about parity will be useful.

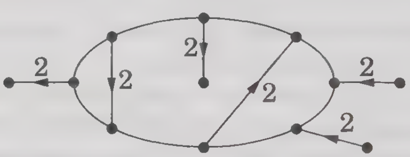
7.3.23. Lemma. In a nowhere-zero k-flow, every vertex is incident to an even number of edges of odd weight.

Proof: Since at each vertex the total weight on entering edges equals the total weight on exiting edges, the sum of the weights is even. ■

7.3.24. Theorem. Let G be a cubic graph. If G has a nowhere-zero 4-flow, then G is 3-edge-colorable.

Proof: By Proposition 7.3.15, we may assume that G has a positive 4-flow (D, f), and thus $f(e) \in \{1, 2, 3\}$ for each edge e. By Lemma 7.3.23, each vertex is incident to exactly one edge of weight 2. Thus the edges of weight 2 form a 1-factor in G, and deleting them leaves a union of disjoint cycles. To complete a 1-factorization, it suffices to show that each of these cycles has even length.

Let C be such a cycle. The edges of weight 2 that are incident to vertices of C are chords or join $V(C)$ with $\overline{V(C)}$. The chords occupy an even size subset of $V(C)$. Thus it suffices to show that the number of edges between $V(C)$ and $\overline{V(C)}$ is even. These edges all have weight 2. Since the net flow out of $V(C)$ must be 0 and all edges between $V(C)$ and $\overline{V(C)}$ have flow 2, the number of edges leaving $V(C)$ must equal the number of edges entering it. ■



Since the Petersen graph is not 3-edge-colorable, Theorem 7.3.24 implies that it is not 4-flowable. Existence of nowhere-zero k-flows is preserved by subdivision: when an edge e of weight j in a nowhere-zero k-flow is subdivided, replacing it with a path of length 2 oriented in the same direction with weight j on both edges yields a nowhere-zero k-flow in the new graph. Thus subdivisions of the Petersen graph also have no nowhere-zero 4-flows.

The converse of Theorem 7.3.24 is true but not trivial, since it may not be possible to treat the color classes as edge sets of fixed weight and orient the graph to make this a 4-flow. In the graph $C_3 \square K_2$ of Example 7.3.21, there is essentially only one proper 3-edge-coloring, and when the color classes are labeled $1, 2, 3$ it is not possible to obtain a 4-flow. In the positive 4-flow in Example 7.3.21, the edges of weight 1 do not form a matching.

Nevertheless, we can apply the next theorem to guarantee nowhere-zero 4-flows in cubic graphs. The characterization is more general, since it does not require regularity.

7.3.25. Theorem. A graph has a nowhere-zero 4-flow if and only if it is the union of two even graphs.

Proof: Let G_1, G_2 be even graphs with $G = G_1 \cup G_2$. Let D be an orientation of G, restricting to D_i on G_i. By Proposition 7.3.19 and Proposition 7.3.15, G_i has a nowhere-zero 2-flow (D_i, f_i). Extend f_i to $E(G)$ by letting $f_i(e) = 0$ for $e \in E(G) - E(G_i)$. Let $f = f_1 + 2f_2$. This weight function is odd on $E(G_1)$ and is ± 2 on $E(G) - E(G_1)$, so it is nowhere-zero. Its magnitude is always at most 3, and by Proposition 7.3.16 (D, f) is a flow; thus it is a nowhere-zero 4-flow.

Conversely, let (D, f) be a nowhere-zero 4-flow on G. Let $E_1 = \{e \in E(G)\colon f(e) \text{ is odd}\}$. By Lemma 7.3.23, E_1 forms an even subgraph of G. Thus there is a nowhere-zero 2-flow (D_1, f_1) on E_1, where D_1 agrees with D. Extend f_1 to $E(G)$ by letting $f_1(e) = 0$ for $e \in E(G) - E_1$; now (D, f_1) is a 2-flow on G.

Define f_2 on $E(G)$ by $f_2 = (f - f_1)/2$. By Proposition 7.3.16, (D, f_2) is a flow on G. It is an integer flow, since $f(e) - f_1(e)$ is always even. By Lemma 7.3.23, the set $E_2 = \{e \in E(G)\colon f_2(e) \text{ is odd}\}$ forms an even subgraph of G. For $e \in E(G) - E_1$, we have $f(e) = \pm 2$ and $f_1(e) = 0$, which yields $f_2(e) = \pm 1$, so $E(G) - E_1 \subseteq E_2$. Now G is the union of two even subgraphs. ■

7.3.26. Corollary. If G is a cubic graph, then G is 3-edge-colorable if and only if G has a nowhere-zero 4-flow.

Proof: Every 3-edge-colorable cubic graph is the union of two even subgraphs: the edges of colors 1 and 2, and the edges of colors 1 and 3. ■

In light of Theorem 7.3.22, Corollary 7.3.26 generalizes Tait's Theorem.

We have seen that subdivisions of the Petersen graph are not 4-flowable. Among bridgeless graphs, Tutte conjectured that excluding such subgraphs yields nowhere-zero 4-flows.

7.3.27. Conjecture. (Tutte's 4-flow Conjecture—Tutte [1966b]) Every bridgeless graph containing no subdivision of the Petersen graph is 4-flowable. ■

Since every graph containing a subdivision of the Petersen graph is nonplanar, Tutte's 4-flow Conjecture implies the Four Color Theorem. Since nowhere-zero 4-flows are equivalent to 3-edge-colorings on cubic graphs, the 4-flow Conjecture also implies the 3-edge-coloring Conjecture (which has been proved). Researchers have hoped for an elegant proof of Tutte's 4-flow Conjecture as a way of obtaining a shorter proof of the Four Color Theorem.

We close this section by describing of several other famous conjectures related to these. Every nowhere-zero k-flow is a nowhere-zero $k+1$-flow, so conditions for nowhere-zero 3-flows or 5-flows should be more or less restrictive, respectively, than conditions for a nowhere-zero 4-flow. Statements of Tutte's 3-flow Conjecture appear in Steinberg [1976] and in Bondy–Murty [1976, Unsolved Problem 48].

7.3.28. Conjecture. (Tutte's 3-flow Conjecture) Every 4-edge-connected graph has a nowhere-zero 3-flow. ■

7.3.29. Conjecture. (Tutte's 5-flow Conjecture—Tutte [1954b]) Every bridgeless graph has a nowhere-zero 5-flow. ■

Kilpatrick [1975] and Jaeger [1979] proved that every bridgeless graph is 8-flowable. Seymour [1981] proved that these graphs are 6-flowable. We sketch the ideas of the 8-flow Theorem; details are requested in exercises.

Both proofs reduce to the 3-edge-connected case, by showing that a smallest bridgeless graph without a nowhere-zero k-flow is simple, 2-connected, and 3-edge-connected (Exercise 26). The main step is then to express a 3-edge-connected graph as a union of subgraphs with good flows. A generalization of Theorem 7.3.25 then applies: If G_1 has a nowhere-zero k_1-flow and G_2 has a nowhere-zero k_2-flow, then $G_1 \cup G_2$ has a nowhere-zero k_1k_2-flow (Exercise 24). (The converse also holds but is not needed.)

For the 8-flow Theorem, it then suffices to prove that a 3-edge-connected graph can be expressed as the union of three even subgraphs. First, adding an additional copy of each edge in G yields a 6-edge-connected graph G'. Then, the Tree–Packing Theorem of Nash-Williams (Corollary 8.2.59) yields three pairwise edge-disjoint spanning trees in G'. These correspond to three spanning trees in G. Since we obtained them as edge-disjoint trees in G', each edge of G appears in at most two of them.

Within a spanning tree of G, one can find a **parity subgraph** of G, meaning a spanning subgraph H such that $d_H(v) \equiv d_G(v) \bmod 2$ for all $v \in V(G)$ (Exercise 25). The complement within $E(G)$ of the edge set of a parity subgraph is an even subgraph of G. Since our three spanning trees have no common edge, the complements of their parity subgraphs express G as a union of three even subgraphs. By Proposition 7.3.19, each has a nowhere-zero 2-flow, and hence G has a nowhere-zero 8-flow.

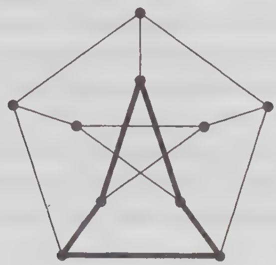
The approach in Seymour [1981] is similar; the task is to express a 3-edge-connected graph as a union of an even graph and a 3-flowable graph. This uses more subtle concepts, including a notion of "modular" flows originally introduced by Tutte [1949]. Seymour's proof was refined by Younger [1983] and Jaeger [1988]. We refer the reader to Zhang [1997] for an exposition.

Celmins [1984] proved that if the 5-flow Conjecture is false, then the smallest counterexample is a snark having girth at least 7 and no nontrivial edge cut with four edges.

We describe one additional conjecture and its relation to earlier topics. In a 2-edge-connected plane graph, all facial boundaries are cycles. Each edge lies in the boundary of two faces, so the facial cycles together cover every edge exactly twice. It is reasonable to ask whether such a covering can be obtained also for graphs that are not planar.

7.3.30. Definition. A **cover** of a graph G is a list of subgraphs whose union is G. A **double cover** is a cover with each edge appearing in exactly two subgraphs in the list. A **cycle double cover (CDC)** is a double cover consisting of cycles.

7.3.31. Example. Together with the outer 5-cycle, the 5 rotations of the 5-cycle illustrated below form a CDC of the Petersen graph. The Petersen graph also has CDCs using cycles of other lengths (Exercise 36). ∎



Since cut-edges appear in no cycles, only bridgeless graphs have CDCs.

7.3.32. Conjecture. (Cycle Double Cover Conjecture—Szekeres [1973], Seymour [1979b]) Every bridgeless graph has a cycle double cover. ∎

One might think that the CDC Conjecture follows immediately using embeddings on surfaces with handles, but such embeddings may have facial boundaries that traverse the same edge twice. The **Strong Embedding Conjecture** asserts that every 2-connected graph has an embedding (on some surface) in which the boundary of each face is a single cycle. Applying this to each block of a 2-edge-connected graph would yield the CDC Conjecture.

In discussing the CDC, we must alert the reader to an unfortunate conflict in terminology. Throughout this book, we use the definition of *cycle* that is common in discussing connectivity, girth, circumference, planarity, etc. In this language, a *circuit* is an equivalence class of closed trails (ignoring the starting vertex), and an *even graph* is a graph whose vertex degrees are all even. A circuit traverses a connected even graph.

The literature on cycle covers generally reverses this terminology, using "circuit" to mean what we call a cycle and "cycle" to mean what we call an even graph. Since the term "even graph" strongly evokes its definition, we hope that our usage will be clear.

The alternative usage arises from other contexts. In a matroid (Section 8.2), the circuits are the minimal dependent sets, and in the cycle matroid of a graph these are the edge sets of the cycles. The cycle space of a graph is a vector space (using scalars $\{0, 1\}$) where the coordinates are indexed by the edges and the vectors correspond to the even subgraphs.

The original CDC Conjecture states that every bridgeless graph has a double cover by even subgraphs. That phrasing is equivalent to ours, since every even graph is an edge-disjoint union of cycles.

Thus we might seek a double cover by using a small number of even subgraphs. The cycles in a cycle double cover are even subgraphs; when cycles are pairwise edge-disjoint, they can be combined to form a single even subgraph. This leads to the connection between integer flows and cycle double covers.

7.3.33. Proposition. A graph has a nowhere-zero 4-flow if and only if it has a cycle double cover forming three even subgraphs.

Proof: Theorem 7.3.25 states that a graph has a nowhere-zero 4-flow if and only if it is the union of two even subgraphs E_1, E_2. Let $E_3 = E_1 \triangle E_2$. At each vertex v the degree in E_3 is the sum of the degrees in E_1 and E_2 minus twice the number of common incident edges; hence it is even. Hence E_3 is an even subgraph, and it contains precisely the edges that appear in just one of $\{E_1, E_2\}$. Cycle decompositions of E_1, E_2, E_3 thus combine to yield a CDC.

Conversely, if a CDC forms three even subgraphs, then omitting one of them leaves the graph expressed as the union of two even subgraphs, and hence a nowhere-zero 4-flow exists. ■

Let **P** denote the family of graphs that do not contain a subdivision of the Petersen graph. By Proposition 7.3.33, Tutte's 4-flow Conjecture implies that every graph in **P** has a CDC. Alspach–Goddyn–Zhang [1994] proved a deep result that yields cycle double covers for graphs in **P**. (They proved that a stronger covering property holds for G if and only if $G \in \mathbf{P}$.) In light of Proposition 7.3.33, this is a partial result toward Tutte's 4-flow Conjecture.

The CDC Conjecture is also related to snarks. Goddyn [1985] proved that if the CDC Conjecture is false, then the smallest counterexample is a snark with girth at least 8.

EXERCISES

7.3.1. (−) Prove that every Hamiltonian 3-regular graph has a Tait coloring.

7.3.2. (−) Exhibit 3-regular simple graphs with the following properties.
a) Planar but not 3-edge-colorable.
b) 2-connected but not 3-edge-colorable.
c) Planar with connectivity 2, but not Hamiltonian.

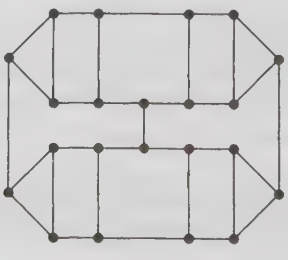
• • • • •

7.3.3. Prove that every maximal plane graph other than K_4 is 3-face-colorable.

7.3.4. Without using the Four Color Theorem, prove that every Hamiltonian plane graph is 4-face-colorable (nothing is assumed about the vertex degrees).

7.3.5. Prove that a 2-edge-connected plane graph is 2-face-colorable if and only if it is Eulerian.

7.3.6. Use Tait's Theorem (Theorem 7.3.2) to prove that $\chi'(G) = 3$ for the graph G below.

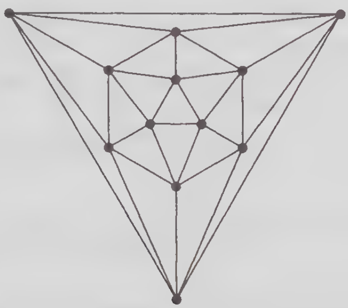


7.3.7. (!) Let G be a plane triangulation.

a) Prove that the dual G^* has a 2-factor.

b) Use part (a) to prove that the vertices of G can be 2-colored so that every face has vertices of both colors. (Hint: Use the idea in the proof of Theorem 7.3.2.) (Buršteĭn [1974], Penaud [1975])

7.3.8. (+) It has been conjectured that every planar triangulation has edge-chromatic number $\Delta(G)$, and this has been proved when $\Delta(G)$ is high enough. Show that $\chi'(G) = \Delta(G)$ for the graph of the icosahedron, illustrated below.



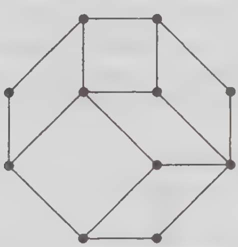
7.3.9. Prove that a proper 4-coloring of the icosahedron uses each color exactly 5 times.

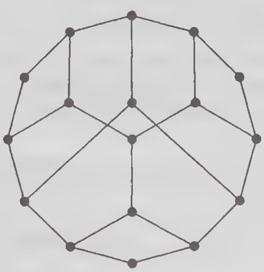
7.3.10. Whitney [1931] proved that every 4-connected planar triangulation is Hamiltonian. Use this to reduce the Four Color Problem to the problem of proving that every Hamiltonian planar graph is 4-colorable.

7.3.11. Find a 5-connected planar graph. Does there exist a 6-connected planar graph?

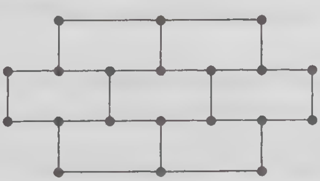
7.3.12. Let G be a planar graph with at least three faces. Prove that G has a vertex partition into two sets whose induced subgraphs are trees if and only if G^* is Hamiltonian.

7.3.13. (!) For each of the planar graphs below, present a Hamiltonian cycle or use planarity (Grinberg's condition) to prove that it is non-Hamiltonian.



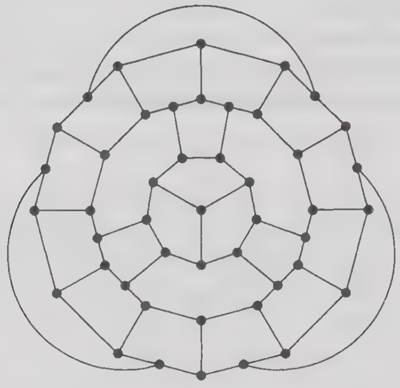


7.3.14. Let G be the graph below. Prove that G has no Hamiltonian cycle. Explain why Grinberg's Theorem cannot be used directly to prove this.

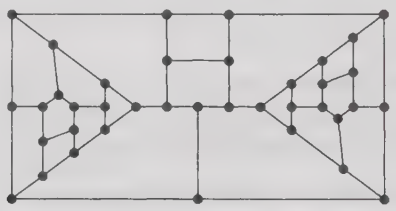


7.3.15. (!) Prove Grinberg's Theorem using Euler's Formula.

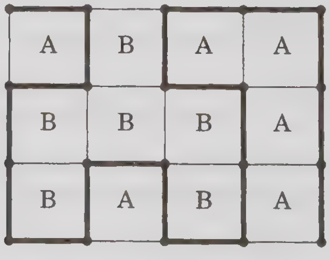
7.3.16. (!) Use Grinberg's condition to prove that the Grinberg graph (below) is not Hamiltonian.



7.3.17. (!) The smallest known 3-regular 3-connected planar graph that is not Hamiltonian has 38 vertices and appears below. Prove that this graph is not Hamiltonian. (Lederberg [1966], Bosák [1966], Barnette)



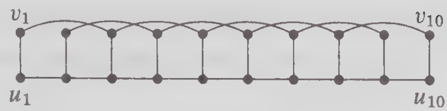
7.3.18. Let G be the grid graph $P_m \square P_n$. Let Q be a Hamiltonian path from the upper left corner vertex to the lower right corner vertex, such as that shown in bold below. Note that Q partitions the grid into regions, of which some open to the left or downward and others open to the right or upward. Prove that the total area of the up-right regions (B) equals the total area of the down-left regions (A). (Fisher–Collins–Krompart [1994])



7.3.19. (!) The **generalized Petersen graph** $P(n, k)$ is the graph with vertices $\{u_1, \ldots, u_n\}$ and $\{v_1, \ldots, v_n\}$ and edges $\{u_iu_{i+1}\}$, $\{u_iv_i\}$, and $\{v_iv_{i+k}\}$, where addition is modulo n. The Petersen graph itself is $P(5, 2)$.

a) Prove that the subgraph of $P(n, 2)$ induced by k consecutive pairs $\{u_i, v_i\}$ has a spanning cycle if $k \equiv 1 \bmod 3$ and $k \geq 4$.

b) Use part (a) to prove that $\chi'(P(n,2)) = 3$ if $n \geq 6$.



7.3.20. (−) Let G be a 3-regular graph. Prove that if G is the union of three cycles, then G is 3-edge-colorable.

7.3.21. (+) *"Flower snarks".* Let G_k and H_k be as constructed in (Example 7.3.13).

a) Prove that G_k is 3-edge-colorable.

b) Prove that H_k is not 3-edge-colorable when k is odd. (Isaacs [1975])

7.3.22. Prove that every edge cut of $K_k \square C_l$ that does not isolate a vertex has at least $2k$ edges.

7.3.23. (∗) Prove that applying the dot product operation (Definition 7.3.12) to two snarks yields a third snark. (Isaacs [1975])

7.3.24. (∗!) Let G_1 and G_2 be graphs. Prove that if G_1 has a nowhere-zero k_1-flow and G_2 has a nowhere-zero k_2-flow, then $G_1 \cup G_2$ has a nowhere-zero k_1k_2-flow.

7.3.25. (!) A **parity subgraph** of G is a subgraph H such that $d_H(v) \equiv d_G(v) \bmod 2$ for all $v \in V(G)$. Prove that every spanning tree of a connected graph G contains a partity subgraph of G. (Itai–Rodeh [1978])

7.3.26. (∗) For $k \geq 3$, prove that a smallest nontrivial 2-edge-connected graph G having no nowhere-zero k-flow must be simple, 2-connected, and 3-edge-connected. (Hint: First exclude loops and vertices of degree 2 and reduce to consideration of blocks. Then exclude multiple edges and finally edge cuts of size 2. In each case, compare G to a graph obtained from it by deleting or contracting edges.)

7.3.27. (∗) Prove that every Hamiltonian graph has a nowhere-zero 4-flow.

7.3.28. (∗) Prove that every bridgeless graph with a Hamiltonian path has a nowhere-zero 5-flow. (Jaeger [1978])

7.3.29. (∗) Embed K_6 on the torus, and let G be the dual graph. Find a nowhere-zero 5-flow on G.

7.3.30. (∗) Prove that a graph G is the union of r even subgraphs if and only if G has a nowhere-zero 2^r-flow. (Matthews [1978])

7.3.31. (∗) Let G be a graph having a cycle double cover forming 2^r even subgraphs. Prove that G has a nowhere-zero 2^r-flow. (Jaeger [1988])

7.3.32. (∗!) A **modular 3-orientation** of a graph G is an orientation D such that $d_D^+(v) \equiv d_D^-(v) \bmod 3$ for all $v \in V(G)$. Prove that a bridgeless graph has a nowhere-zero 3-flow if and only if it has a modular 3-orientation. (Steinberg–Younger [1989])

7.3.33. (∗) *Characterization of nowhere-zero k-flows.* Let G be a bridgeless graph, let D be an orientation of G, and let a and b be positive integers. Prove that the following statements are equivalent. (Hoffman [1958])

a) $\frac{a}{b} \leq \frac{|[S,\overline{S}]|}{|[\overline{S},S]|} \leq \frac{b}{a}$ for every nonempty proper vertex subset S.

b) G has an integer flow using weights in the interval $[a, b]$.

c) G has a real-valued flow using weights in the interval $[a, b]$.

7.3.34. $(*)$ Find cycle double covers for the graphs $C_m \vee K_1$, $C_m \vee 2K_1$, and $C_m \vee K_2$.

7.3.35. $(*)$ Find the cycle double covers with fewest cycles for every 3-regular simple graph with 6 vertices.

7.3.36. $(*-)$ Let G be the Petersen graph. Find a cycle double cover of G whose elements are not all 5-cycles. Find a double cover of G consisting of 1-factors. (Hint: Consider the drawing of G having a 9-cycle on the "outside". Comment: Fulkerson [1971] conjectured that every bridgeless cubic graph has a double cover consisting of 6 perfect matchings.)

7.3.37. $(*)$ Prove that any two 6-cycles in the Petersen graph must have at least two common edges. Conclude that the Petersen graph has no CDC consisting of five 6-cycles. Use this and Exercise 7.3.20 to conclude that the Petersen graph has no CDC consisting of even cycles. (C.Q. Zhang)

7.3.38. $(*!)$ A cycle double cover is **orientable** if its cycles can be oriented as directed cycles so that for each edge, the two cycles containing it traverse it in opposite directions. A digraph is **even** if $d^-(v) = d^+(v)$ for each vertex v.

a) Suppose that G has a nonnegative k-flow (D, f). Prove that f can be expressed as $\sum_{i=1}^{k-1} f_i$, where each (D, f_i) is a nonnegative 2-flow on G. (Hint: Use induction on k.) (Little–Tutte–Younger [1988])

b) Prove that a graph G has a positive k-flow (D, f) if and only if D is the union of $k-1$ even digraphs such that each edge e in D appears in exactly $f(e)$ of them. (Little–Tutte–Younger [1988])

c) Prove that a graph G has a nowhere-zero 3-flow if and only if it has an orientable cycle double cover forming three even subgraphs. (Tutte [1949])

7.3.39. $(*)$ Let G be a graph having a CDC formed from four even subgraphs. Prove that G also has a CDC formed from three even subgraphs. (Hint: Use symmetric differences.)

7.3.40. $(*)$ In the Petersen graph, prove that the solution to the Chinese Postman Problem has total length 20, but the minimum total length of cycles covering the Petersen graph is 21.

7.3.41. $(*)$ Let M be a perfect matching in the Petersen graph. Prove that there is no list of cycles in the Petersen graph that together cover every edge of M exactly twice and all other edges exactly once. (Itai–Rodeh [1978], Seymour [1979b])

7.3.42. $(*)$ Let G be a graph in which a shortest covering walk (that is, an optimal solution to the Chinese Postman Problem) decomposes into cycles. Prove that G has a cycle cover of total length at most $e(G) + n(G) - 1$. Determine the minimum length of a cycle cover of $K_{3,t}$ in terms of the number of edges and vertices.

Chapter 8

Additional Topics

In this chapter we explore more advanced or specialized material. Each section gives a glimpse of a topic that deserves its own chapter (or book). Several sections treat more difficult material near the end.

8.1. Perfect Graphs

We have discussed the lower bound $\chi(G) \geq \omega(G)$ for chromatic number; the vertices of a clique need different colors. In Section 5.3, we discussed graphs whose induced subgraphs all achieve equality in this bound.

8.1.1. Definition. A graph G is **perfect** if $\chi(H) = \omega(H)$ for every induced subgraph H of G.

When discussing perfect graphs, it is common to use **stable set** to mean an independent set of vertices. As before, a **clique** is a set of pairwise adjacent vertices. As usual, **maximum** means maximum-sized.

Since we focus on vertex coloring, again in this section we restrict our attention to simple graphs. Complementation converts cliques to stable sets and vice versa, so $\omega(\overline{H}) = \alpha(H)$. Properly coloring $\overline{H}$ means expressing $V(H)$ as a union of cliques in H; such a set of cliques in H is a **clique covering** of H. Thus for every graph G we have four optimization parameters of interest.

independence number	$\alpha(G)$	max size of a stable set
clique number	$\omega(G)$	max size of a clique
chromatic number	$\chi(G)$	min size of a coloring
clique covering number	$\theta(G)$	min size of a clique covering

Berge actually defined two types of perfection:

G is γ-**perfect** if $\chi(G[A]) = \omega(G[A])$ for all $A \subseteq V(G)$.
G is α-**perfect** if $\theta(G[A]) = \alpha(G[A])$ for all $A \subseteq V(G)$.

Our definition of perfect is the same as this definition of γ-perfect (Berge used $\gamma(G)$ for chromatic number). Since $\overline{G}[A]$ is the complement of $G[A]$, the definition of α-perfect can be stated in terms of $\overline{G}$ as "$\chi(\overline{G}[A]) = \omega(\overline{G}[A])$ for all $A \subseteq V(G)$". Thus "G is α-perfect" has the same meaning as "$\overline{G}$ is γ-perfect".

We now use only one definition of perfection, because Lovász [1972a] proved "G is γ-perfect if and only if G is α-perfect". In terms of our original definition of perfection, this becomes "G is perfect if and only if $\overline{G}$ is perfect". This statement is the **Perfect Graph Theorem (PGT)**.

Always $\chi(G) \geq \omega(G)$ and $\theta(G) \geq \alpha(G)$, since a clique and a stable set share at most one vertex. A statement of perfection for a class of graphs is thus an integral min-max relation. We observed in Example 5.3.21 that several familiar min-max relations are statements that bipartite graphs, their line graphs, and the complements of such graphs are perfect.

If $k \geq 2$, then $\chi(C_{2k+1}) > \omega(C_{2k+1})$ and $\chi(\overline{C}_{2k+1}) > \omega(\overline{C}_{2k+1})$ (Exercise 1). Thus odd cycles and their complements (except C_3 and $\overline{C}_3$) are imperfect.

8.1.2. Conjecture. (**Strong Perfect Graph Conjecture (SPGC)**—Berge [1960]) A graph G is perfect if and only if both G and $\overline{G}$ have no induced subgraph that is an odd cycle of length at least 5. ■

The SPGC remains open. Since the condition in the conjecture is self-complementary, the SPGC implies the PGT.

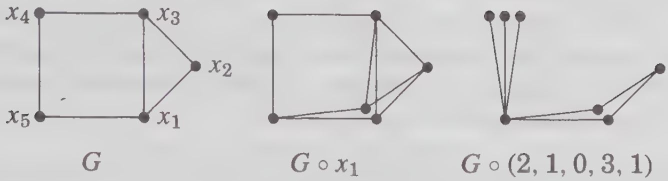
Having presented several classical families of perfect graphs in Section 5.3, our goal now is to prove the Perfect Graph Theorem. Later we also study properties of minimal imperfect graphs and classes of perfect graphs. For further reading, Golumbic [1980] provides a thorough introduction to the subject. Berge–Chvátal [1984] collects and updates many of the classical papers.

THE PERFECT GRAPH THEOREM

In 1960, Berge conjectured that γ-perfection and α-perfection are equivalent (see Berge [1961]). Lovász [1972a] stunned the world of combinatorics by proving this important and well-known conjecture at the age of 22. Fulkerson also studied it, reducing it to a statement he thought was too strong to be true. When Berge told him that Lovász had proved it, within hours he proved the missing lemma (Lemma 8.1.4), thus illustrating that a theorem becomes easier to prove when known to be true (Fulkerson [1971]).

We will prove the Perfect Graph Theorem using an operation that enlarges a graph without affecting the property of perfection.

8.1.3. Definition. **Duplicating a vertex** x of G produces a new graph $G \circ x$ by adding a vertex x' with $N(x') = N(x)$. The **vertex multiplication** of G by the nonnegative integer vector $h = (h_1, \ldots, h_n)$ is the graph $H = G \circ h$ whose vertex set consists of h_i copies of each $x_i \in V(G)$, with copies of x_i and x_j adjacent in H if and only if $x_i \leftrightarrow x_j$ in G.



8.1.4. Lemma. Vertex multiplication preserves γ-perfection and α-perfection.

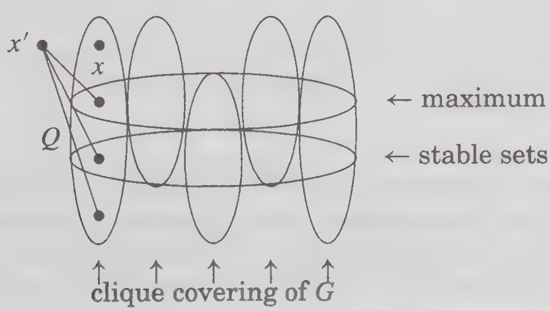
Proof: We first observe that $G \circ h$ can be obtained from an induced subgraph of G by successive vertex duplications. If every h_i is 0 or 1, then $G \circ h = G[A]$, where $A = \{i \colon h_i > 0\}$. Otherwise, start with $G[A]$ and perform duplications until there are h_i copies of x_i (for each i). Each vertex duplication preserves the property that copies of x_i and x_j are adjacent if and only if $x_i x_j \in E(G)$, so the resulting graph is $G \circ h$.

If G is α-perfect but $G \circ h$ is not, then some operation in the creation of $G \circ h$ from $G[A]$ produces a graph that is not α-perfect from an α-perfect graph. It thus suffices to prove that vertex duplication preserves α-perfection. The same reduction holds for γ-perfection. Since every proper induced subgraph of $G \circ x$ is an induced subgraph of G or a vertex duplication of an induced subgraph of G, we further reduce our claim to showing that $\chi(G \circ x) = \omega(G \circ x)$ when G is γ-perfect and that $\alpha(G \circ x) = \theta(G \circ x)$ when G is α-perfect.

When G is γ-perfect, we extend a proper coloring of G to a proper coloring of $G \circ x$ by giving x' the same color as x. No clique contains both x and x', so $\omega(G \circ x) = \omega(G)$. Hence $\chi(G \circ x) = \chi(G) = \omega(G) = \omega(G \circ x)$.

When G is α-perfect, we consider two cases. If x belongs to a maximum stable set in G, then adding x' to it yields $\alpha(G \circ x) = \alpha(G) + 1$. Since $\theta(G) = \alpha(G)$, we can obtain a clique covering of this size by adding x' as a 1-vertex clique to some set of $\theta(G)$ cliques covering G.

If x belongs to no maximum stable set in G, then $\alpha(G \circ x) = \alpha(G)$. Let Q be the clique containing x in a minimum clique cover of G. Since $\theta(G) = \alpha(G)$, Q intersects every maximum stable set in G. Since x belongs to no maximum stable set, $Q' = Q - x$ also intersects every maximum stable set. This yields $\alpha(G - Q') = \alpha(G) - 1$. Applying the α-perfection of G to the induced subgraph $G - Q'$ (which contains x) yields $\theta(G - Q') = \alpha(G - Q')$. Adding $Q' \cup \{x'\}$ to a set of $\alpha(G) - 1$ cliques covering $G - Q'$ yields a set of $\alpha(G)$ cliques covering $G \circ x$. ■



8.1.5. Lemma. In a minimal imperfect graph, no stable set intersects every maximum clique.

Proof: If a stable set S in G intersects every $\omega(G)$-clique, then perfection of $G - S$ yields $\chi(G - S) = \omega(G - S) = \omega(G) - 1$, and S completes a proper $\omega(G)$-coloring of G. This makes G perfect. ■

8.1.6. Theorem. (The Perfect Graph Theorem (PGT) - Lovász [1972a, 1972b]) A graph is perfect if and only if its complement is perfect.

Proof: It suffices to show that α-perfection of G implies γ-perfection of G; applying this to $\overline{G}$ yields the converse. If the claim fails, then we consider a minimal graph G that is α-perfect but not γ-perfect. By Lemma 8.1.5, we may assume that every maximal stable set S in G misses some maximum clique $Q(S)$.

We design a special vertex multiplication of G. Let $\mathbf{S} = \{S_i\}$ be the list of maximal stable sets of G. We weight each vertex by its frequency in $\{Q(S_i)\}$, letting h_j be the number of stable sets $S_i \in \mathbf{S}$ such that $x_j \in Q(S_i)$. By Lemma 8.1.4, $H = G \circ h$ is α-perfect, yielding $\alpha(H) = \theta(H)$. We use counting arguments for $\alpha(H)$ and $\theta(H)$ to obtain a contradiction.

Let A be the 0,1-matrix of the incidence relation between $\{Q(S_i)\}$ and $V(G)$; we have $a_{i,j} = 1$ if and only if $x_j \in Q(S_i)$. By construction, h_j is the number of 1s in column j of A, and $n(H)$ is the total number of 1s in A. Since each row has $\omega(G)$ 1s, also $n(H) = \omega(G)\,|\mathbf{S}|$. Since vertex duplication cannot enlarge cliques, we have $\omega(H) = \omega(G)$. Therefore, $\theta(H) \ge n(H)/\omega(H) = |\mathbf{S}|$.

We obtain a contradiction by proving that $\alpha(H) < |\mathbf{S}|$. Every stable set in H consists of copies of elements in some stable set of G, so a maximum stable set in H consists of all copies of all vertices in some maximal stable set of G. Hence $\alpha(H) = \max_{T \in \mathbf{S}} \sum_{i:\, x_i \in T} h_i$. The sum counts the 1s in A that appear in the columns indexed by T. If we count these 1s instead by rows, we obtain $\alpha(H) = \max_{T \in \mathbf{S}} \sum_{S \in \mathbf{S}} |T \cap Q(S)|$. Since T is a stable set, it has at most one vertex in each chosen clique $Q(S)$. Furthermore, T is disjoint from $Q(T)$. With $|T \cap Q(S)| \le 1$ for all $S \in \mathbf{S}$, and $|T \cap Q(T)| = 0$, we have $\alpha(H) \le |\mathbf{S}| - 1$. ■

$$
\begin{array}{r|ccc|}
 & \multicolumn{3}{c}{V(G)} \\
\cline{2-4}
Q(S_1) & \vdots & \vdots & \vdots \\
Q(T) & 0 & 0 & 0 \\
Q(S_n) & \vdots & \vdots & \vdots \\
\cline{2-4}
\multicolumn{1}{c}{} & \uparrow & \uparrow & \multicolumn{1}{c}{\uparrow} \\
\multicolumn{1}{c}{} & T & T & \multicolumn{1}{c}{T}
\end{array}
$$

8.1.7.* Remark. *Linear optimization and duality.* Clique-vertex incidence matrices also arise in expressing α and θ as integer optimization problems. A linear (maximization) program can be written as "maximize $c \cdot x$ over nonnegative vectors x such that $Ax \le b$", where A is a matrix, b, c are vectors, and each row of $Ax \le b$ is a linear constraint $a_i \cdot x \le b_i$ on the vector x of variables. A vector x satisfying all the constraints is a **feasible solution**.

An **integer linear program** requires that each x_j also be an integer. Let A be the incidence matrix between maximal cliques and vertices in a graph G; we have $a_{i,j} = 1$ when $v_j \in Q_i$. By definition, $\alpha(G)$ is the solution to "max $\mathbf{1}_n \cdot x$ such that $Ax \le \mathbf{1}_m$" when the variables are required to be nonnegative integers. In the solution, x_j is 1 or 0 depending on whether v_j is in the maximum stable set; the constraints prevent choosing adjacent vertices. Similarly, when B is the incidence matrix between maximal stable sets and vertices, $\omega(G)$ is the solution to "max $\mathbf{1}_n \cdot x$ such that $Bx \le \mathbf{1}_p$" with integer variables.

Every maximization program has a dual minimization program. When the max program is "max $c \cdot x$ such that $Ax \le b$", the dual is "min $y \cdot b$ such that $y^T A \ge c$". This program has a variable y_i for each original constraint and a constraint for each original variable x_j, and it interchanges $c, \max, \le$ with $b, \min, \ge$. When stated in this form, the variables in both programs must be nonnegative. The integer programs dual to ω and α seek the minimum number of stable sets that cover the vertices and the minimum number of cliques that cover the vertices, respectively; this describes $\chi(G)$ and $\theta(G)$.

Using the nonnegativity of the variables, the constraints yield

$$c \cdot x \le y^T Ax \le y \cdot b.$$

The statement "$c \cdot x \le y \cdot b$" for feasible solutions x, y is **weak duality**. The (strong) **Duality Theorem of Linear Programming** states that dual programs having feasible solutions have optimal solutions with the same value when integer solutions are not required.

The statements $\chi \ge \omega$ and $\theta \ge \alpha$ are statements of weak duality for dual pairs of linear programs. A guarantee of strong duality using solutions that have only integer values is a combinatorial min-max relation. We have presented many such relations and observed that they guarantee quick proofs of optimality. They also often lead to fast algorithms for finding optimal solutions, which is one motivation for studying families of perfect graphs. ■

8.1.8.* Example. *Fractional solutions for an imperfect graph.* For the 5-cycle, the linear programs for $\omega, \chi, \alpha, \theta$ all have optimal value 5/2. There are five maximal cliques and five maximal stable sets, each of size 2. Setting each $x_j = 1/2$ gives weight 1 to each clique and stable set, thereby satisfying the constraints for either maximization problem. Setting each $y_i = 1/2$ in the dual programs covers each vertex with a total weight of 1, so again the constraints are satisfied. These programs have no optimal solution in integers, and the integer programs have a "duality gap": $\chi = 3 > 2 = \omega$ and $\theta = 3 > 2 = \omega$. ■

CHORDAL GRAPHS REVISITED

Like trees, the more general class of chordal graphs has many characterizations. The definition by forbidding chordless cycles is a **forbidden substructure characterization**. A finite list of forbidden substructures such as induced subgraphs yields a fast algorithm for testing membership in the class, but for chordal graphs the list is infinite and other methods are needed.

A chordal graph can be built from a single vertex by iteratively adding a vertex joined to a clique; this is the reverse of a simplicial elimination ordering, and we have seen that greedy coloring with respect to such a construction ordering yields an optimal coloring. Many classes of perfect graphs have such a **construction procedure** that produces the graphs in the class and no others. A construction procedure or the reverse **decomposition procedure** may lead to fast algorithms for computations on graphs in the class.

Next we consider another type of characterization.

8.1.9. Definition. An **intersection representation** of a graph G is a family of sets $\{S_v: v \in V(G)\}$ such that $u \leftrightarrow v$ if and only if $S_u \cap S_v \neq \emptyset$. If $\{S_v\}$ is an intersection representation of G, then G is the **intersection graph** of $\{S_v\}$.

Interval graphs are the graphs having intersection representations where each set in the family is an interval on the real line. Line graphs also form an intersection class; the allowed sets are pairs of natural numbers, corresponding to edges of the graph H such that $G = L(H)$. An intersection characterization for chordal graphs was found independently by Walter [1972, 1978], Gavril [1974], and Buneman [1974].

8.1.10. Lemma. If $T_1, \dots, T_k$ are pairwise intersecting subtrees of a tree T, then there is a vertex belonging to all of $T_1, \dots, T_k$.

Proof: (Lehel) We prove the contrapositive. If each vertex v misses some $T(v)$ among $T_1, \dots, T_k$, we mark the edge that leaves v on the unique path to $T(v)$. If T has n vertices, then we make n marks, so some edge uw has been marked twice. Now $T(u)$ and $T(w)$ have no common vertex. ∎

8.1.11. Theorem. A graph is chordal if and only if it has an intersection representation using subtrees of a tree (a **subtree representation**).

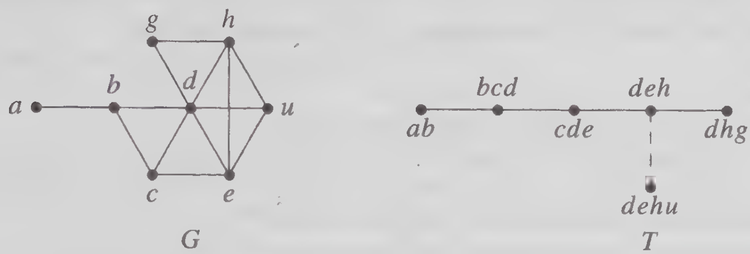
Proof: We prove that the condition is equivalent to the existence of a simplicial elimination ordering. We use induction, with trivial basis K_1.

Let $v_1, \dots, v_n$ be a simplicial elimination ordering for G. Since $v_2, \dots, v_n$ is a simplicial elimination ordering for $G - v_1$, the induction hypothesis yields a subtree representation of $G - v_1$ in a host tree T. Since v_1 is simplicial in G, the set $S = N_G(v_1)$ induces a clique in $G - v_1$. Therefore, the subtrees of T assigned to vertices of S are pairwise intersecting.

By Lemma 8.1.10, these subtrees have a common vertex x. We enlarge T to a tree T' by adding a leaf y adjacent to x, and we add the edge xy to the subtrees representing vertices of S. We represent v_1 by the subtree consisting only of y. This completes a subtree representation of G in T'.

Conversely, let T be a smallest host tree for a subtree representation of G, with each $v \in V(G)$ represented by $T(v) \subseteq T$. If $xy \in E(T)$, then G must have a vertex u such that $T(u)$ contains x but not y; otherwise, contracting xy into y would yield a representation in a smaller tree.

Let x be a leaf of T, and let u be a vertex of G such that $T(u)$ contains x but not its neighbor. The subtrees for neighbors of u in G must contain x and hence are pairwise intersecting. Thus u is simplicial in G. Deleting $T(u)$ yields a subtree representation of $G - u$. We complete a simplicial elimination order of G using such an ordering of $G - u$ given by the induction hypothesis. ■



Because the class of chordal graphs is hereditary, a simplicial elimination ordering can start with any simplicial vertex. Thus a brute-force approach to finding such an ordering would be to examine neighborhoods until we find a simplicial vertex, delete it, and iterate.

Rose–Tarjan–Lueker [1976] found a faster way, which was simplified further by Tarjan [1976]. The idea here, because there is always a simplicial vertex among the vertices farthest from a given vertex (proof of Theorem 5.3.17), is that a simplicial elimination ordering can *end* at any vertex. Thus we start with an arbitrary vertex and list the vertices clumped around it. The result is a simplicial construction ordering (the reverse of a simplicial elimination ordering) if and only if the graph is chordal. The algorithm was published with several applications in Tarjan–Yannakakis [1984]; we follow Golumbic [1984].

8.1.12. Algorithm. *Maximum Cardinality Search (MCS)*
Input: A graph G.
Output: A vertex numbering - a bijection $f: V(G) \to \{1, \ldots, n(G)\}$.
Idea: For each unnumbered vertex v, maintain a label $l(v)$ that is its degree among the vertices already numbered. The vertices at the end of a simplicial elimination ordering are those clumped around the last vertex, so in a simplicial construction ordering the vertices with high labels should be added first.
Initialization: Assign label 0 to every vertex. Set $i = 1$.
Iteration: Select any unnumbered vertex with maximum label. Number it i and add 1 to the label of its neighbors. Augment i and iterate. ■

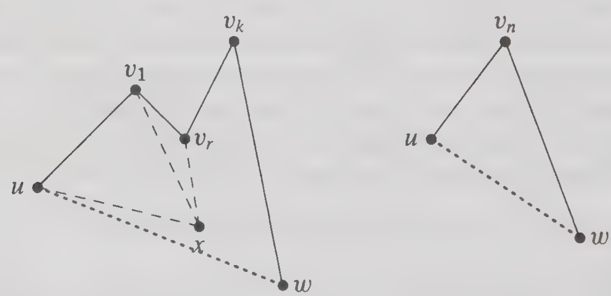
8.1.13. Example. The first vertex chosen in the MCS order is arbitrary. An application of MCS to the graph G above could start by setting $f(c) = 1$ and hence $l(b) = l(d) = l(e) = 1$. Next we could select $f(e) = 2$ and update $l(d) = 2$, $l(h) = l(u) = 1$. Now d is the only vertex with label as large as 2, and hence $f(d) = 3$. We update $l(b) = l(h) = l(u) = 2$, $l(g) = 1$, $l(a) = 0$. Continuing the procedure can produce the order c, e, d, b, h, g, a, u in increasing order of f. This is a simplicial construction ordering, and u, a, g, h, b, d, e, c is a simplicial elimination ordering. ■

8.1.14. Theorem. (Tarjan [1976]). A simple graph G is chordal if and only if the numbering $v_1, \ldots, v_n$ produced by the Maximum Cardinality Search algorithm is a simplicial construction ordering of G.

Proof: If MCS produces a simplicial construction ordering, then G is chordal. Conversely, suppose that G is chordal, and let $f\colon V(G) \to [n]$ be the numbering produced by MCS. A *bridge* of f is a chordless path of length at least 2 whose lowest numbers occur at the endpoints. We prove first that f has no bridge. Otherwise, let $P = u, v_1, \ldots, v_k, w$ be a bridge that minimizes $\max\{f(u), f(w)\}$. By symmetry, we may assume that $f(u) > f(w)$ (f is used as the vertical coordinate to position vertices in the illustration).

Since u is numbered in preference to v_k at time $f(u)$, and w is already numbered at that time, there exists a vertex $x \in N(u) - N(v_k)$ with $f(x) < f(u)$. Letting $v_0 = u$, set $r = \max\{j\colon x \leftrightarrow v_j\}$. The path $P' = x, v_r, \ldots, v_k, w$ is chordless, since $x \leftrightarrow w$ would complete a chordless cycle. Since both of $f(x), f(w)$ are less than $f(u)$, P' is a bridge that contradicts the choice of P. Hence f has no bridge.

With this claim, the proof follows by induction on $n(G)$. It suffices to show that v_n is simplicial, since the application of MCS to $G - v_n$ produces the same numbering $v_1, \ldots, v_{n-1}$ that leaves v_n at the end. If v_n is not simplicial, then v_n has nonadjacent neighbors u, w, in which case u, v_n, w is a bridge of f. ■



The MCS algorithm runs in time $O(n(G)+e(G))$ with careful implementation. For each j, we maintain a doubly linked list of the vertices with label j. For each vertex we store its label and pointers to its neighbors and to its position in the lists. When v is numbered, in time $O(1+d(v))$ we remove v from its list, augment its neighbors labels, and move its neighbors into the next higher lists. To complete the chordality test, we must also check whether the MCS order is a simplicial construction ordering (Exercise 10). Simplicial elimination or construction orderings quickly yield optimal colorings, cliques, stable sets, and clique coverings (Exercise 9).

The alternative algorithm found by Rose, Tarjan, and Leuker is known as Lexicographic Breadth First Search (LBFS). Closely related to the proof of Theorem 5.3.17, LBFS has been used for many applications in testing graph properties and computing graph parameters. Corneil–Olariu–Stewart [2000] provides a good introduction to this topic.

Given a simplicial elimination ordering, Theorem 8.1.14 computes a subtree representation. When the list of maximal cliques is known, Kruskal's algorithm (Theorem 2.3.3) can be used to compute a subtree representation without knowing a simplicial elimination ordering.

8.1.15. Definition. A tree T is a **clique tree** of G if there is a bijection between $V(T)$ and the maximal cliques of G such that for each $v \in V(G)$ the cliques containing v induce a subtree of T.

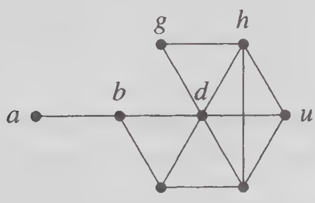
8.1.16. Lemma. Every tree of minimum order in which G has a subtree representation is a clique tree of G.

Proof: Let T be a host tree of minimum order for a subtree representation of G. By Lemma 8.1.10, the vertices of a maximal clique Q in G occur at a common vertex q of T. If the vertices of G assigned to some $q' \in V(T)$ form a proper subclique Q' of Q, then the subtrees for these vertices contain the entire q', q-path in T. The first edge of T on that path can be contracted without changing the intersection graph, which yields a smaller host tree. ∎

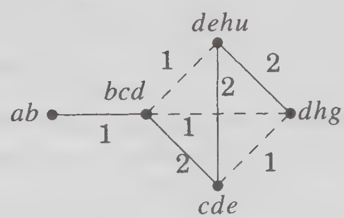
The **weighted intersection graph** of a collection **A** of finite sets is a weighted clique in which the elements of **A** are the vertices and the weight of each edge AA' is $|A \cap A'|$.

8.1.17. Theorem. (Acharya–Las Vergnas [1982]) Let $M(G)$ be the weighted intersection graph of the set of maximal cliques $\{Q_i\}$ of a simple graph G. If T is a spanning tree of $M(G)$, then $w(T) \le \sum n(Q_i) - n(G)$, with equality if and only T is a clique tree.

Proof: (McKee [1993]) Let T be a spanning tree of $M(G)$. Let T_v be the subgraph of T induced by $\{Q_i\colon v \in Q_i\}$. Each vertex $v \in V(G)$ contributes once to the weight of T for each edge of T_v; hence $w(T) = \sum_{v \in V(G)} e(T_v)$. Each T_v is a forest, so $e(T_v) \le n(T_v) - 1$, with equality if and only if T_v is a tree. The term $n(T_v)$ contributes 1 to the size of each clique containing v. Summing the inequality for each vertex yields $w(T) \le \sum n(Q_i) - n(G)$. Equality holds if and only if each T_v is a tree, which is true if and only if T is a clique tree. ∎



G



$M(G)$ (T solid)

$$w(T) = 7 = 15 - 8 = \sum n(Q_i) - n(G)$$

As a consequence of Theorem 8.1.17, we can test whether G is a chordal graph by finding the maximum weight of a spanning tree in $M(G)$. Furthermore, when G is chordal the clique trees are precisely the maximum-weight spanning trees of $M(G)$ (Bernstein–Goodman [1981], Shibata [1988]; see McKee [1993] for related material).

OTHER CLASSES OF PERFECT GRAPHS

Interval graphs are the intersection graphs of collections of intervals on a line. We proved directly in Proposition 5.1.16 that they are perfect; this also follows from being a subclass of the chordal graphs (Exercise 26). Interval graphs arise in linear scheduling problems having constraints on concurrent events (recall Example 5.1.15).

8.1.18. Example. *Classical applications of interval graphs.*

Analysis of DNA chains. Interval graphs were invented for the study of DNA. Benzer [1959] studied the linearity of the chain for higher organisms. Each gene is encoded as an interval, except that the relevant interval may contain a dozen or more irrelevant junk pieces called "introns" among the relevant pieces called "exons". Under the hypothesis that mutations arise from alterations of connected segments, changes in traits of microorganisms can be studied to determine whether their determining amino-acid sets could intersect. This establishes a graph with traits as vertices and "common alterations" as edges. Under the hypotheses of linearity and contiguity, the graph is an interval graph, and this aids in locating genes along the DNA sequence.

Timing of traffic lights. Given traffic streams at an intersection, a traffic engineer (or a person with common sense) can determine which pairs of streams may flow simultaneously. Given an "all-stop" moment in the cycle, the intersection graph of the green-light intervals must be an interval graph whose edges are a subset of the allowable pairs. These can be studied to optimize some criterion such as average waiting time (see Roberts [1978]).

Archeological seriation. Given pottery samples at an archeological dig, we seek a time-line of what styles were used when. Assume that each style was used during one time interval and that two styles appearing in the same grave were used concurrently. Let two styles be an edge if they appear together in a grave. If this graph is an interval graph, then its interval representations are the possible time-lines. Otherwise, the information is incomplete, and the desired interval graph requires additional edges. ■

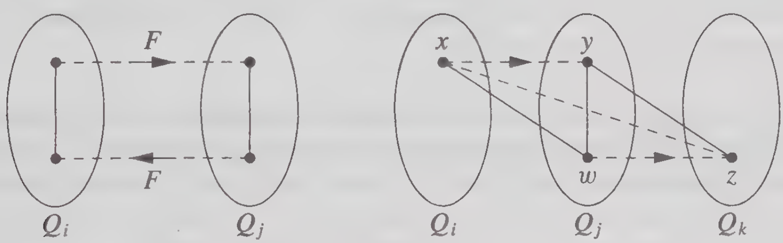
We present two characterizations of interval graphs. Property B in Theorem 8.1.20 is due to Gilmore and Hoffman [1964], and property C is due to Fulkerson and Gross [1965].

8.1.19. Definition. A 0,1-matrix has the **consecutive 1s property** (for columns) if its rows can be permuted so that the 1s in each column appear consecutively. The **clique-vertex incidence matrix** of G is the incidence matrix with rows indexed by the maximal cliques and columns indexed by $V(G)$.

8.1.20. Theorem. The following equivalent conditions on a graph G characterize the interval graphs.
A) G has an interval representation.
B) G is a chordal graph, and $\overline{G}$ is a comparability graph.
C) The clique-vertex incidence matrix has the consecutive 1s property.

Proof: We leave A $\Rightarrow$ B and A $\Leftrightarrow$ C to Exercises 26–27, proving B $\Rightarrow$ C here. Let G be a chordal graph such that $\overline{G}$ has a transitive orientation F. We use F and the absence of chordless cycles in G to establish an ordering on the maximal cliques of G that exhibits the consecutive 1s property for the clique-vertex incidence matrix M.

Let Q_i and Q_j be maximal cliques in G. By maximality, each vertex of one clique has a nonneighbor in the other. Suppose that under F, some edge of $\overline{G}$ points from Q_i to Q_j and some edge of $\overline{G}$ points from Q_j to Q_i. If these edges have a common vertex, then the transitivity of F forces an edge of a clique in G to belong to $\overline{G}$. Hence the situation is as on the left below, with the (dashed) edges of F having four distinct vertices. If the two remaining pairs among these four vertices form edges in G, then G has an induced C_4. Hence at least one diagonal is in $\overline{G}$, but each possible orientation of it in F contradicts transitivity. We conclude that all the edges of $\overline{G}$ between vertex sets Q_i and Q_j point in the same direction in F.



We can now define a tournament T with vertices corresponding to the maximal cliques of G. We put $Q_i \to Q_j$ in T when all edges of F between Q_i and Q_j point from Q_i to Q_j. By the preceding paragraph, T is an orientation of a complete graph. We claim that T is transitive. To prove this we need to show that $Q_i \to Q_j$ and $Q_j \to Q_k$ imply $Q_i \to Q_k$. Suppose that $x \to y$ and $w \to z$ in F with $x \in Q_i$, $y, w \in Q_j$, and $z \in Q_k$. If $y = w$, transitivity of F immediately implies $x \to z$. Otherwise, consider a pair xz as shown on the right above. Joining x and z in G would form an induced C_4 in G, so $x \not\leftrightarrow z$. Hence this pair appears in F, and it must be directed from $x \to z$ to avoid violating transitivity. We conclude that $Q_i \to Q_k$ in T.

A transitive tournament specifies a unique linear ordering of the vertices consistent with the edges; use the transitive tournament T to order the rows of M as $Q_1 \to \cdots \to Q_m$. Suppose that under this ordering there is some column x where the 1s do not appear consecutively. Then we have Q_i, Q_j, Q_k such that

$i < j < k$, $x \in Q_i, Q_k$, $x \notin Q_j$. Since $x \notin Q_j$, the clique Q_j must have some vertex y not adjacent to x, else Q_j could absorb x and would not be maximal. Now $x \in Q_i$ implies $x \to y$ in F, and $x \in Q_k$ implies $y \to x$ in F, which cannot both happen. ■

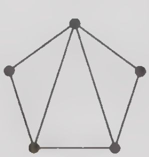
The interval graphs form a relatively small family of perfect graphs. We next discuss larger classes that maintain some of the nice properties of chordal graphs and comparability graphs.

8.1.21. Definition. *Classes of perfect graphs* (conditions on odd cycles apply only for length at least 5).
o-triangulated: every odd cycle has a noncrossing pair of chords.
parity: every odd cycle has a crossing pair of chords.
Meyniel: every odd cycle has at least two chords.
weakly chordal: no induced cycle of length at least 5 in G or $\overline{G}$.
strongly perfect: every induced subgraph has a stable set meeting all its maximal cliques.

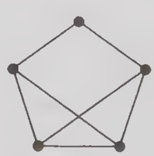
Gallai [1962] proved that o-triangulated graphs are perfect. Every chordal graph is o-triangulated (Exercise 34) and weakly chordal (Exercise 40). All o-triangulated and parity graphs are Meyniel graphs. Meyniel graphs are perfect (Meyniel [1976], Lovász [1983]) and also strongly perfect (Ravindra [1982]).

Parity graphs, shown to be perfect in Olaru [1969] and Sachs [1970], carry that name due to a later characterization by Burlet and Uhry [1984]: G is a parity graph if and only if, for every pair $x, y \in V(G)$, the chordless x, y-paths are all even or all odd (Exercise 36).

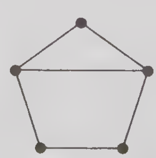
8.1.22. Example. The graphs below exhibit differences among these classes. Here **T**, **C**, **O**, **P**, **M**, **W** respectively denote the classes of chordal (**T**riangulated), comparability, o-triangulated, parity, Meyniel, and weakly chordal graphs. ■



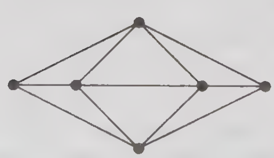
T − P



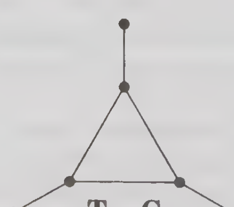
P − O



(W ∩ C) − M



M − O − P



T − C

Strongly perfect graphs were introduced by Berge and Duchet [1984]. Changing maximal to maximum in the definition yields a weaker requirement equivalent to γ-perfection; a stable set meeting all maxim*um* cliques can be used as the first color class in an $\omega(G)$-coloring constructed inductively. Thus strongly perfect graphs are perfect.

The class of strongly perfect graphs does not contain all Meyniel graphs or all weakly chordal graphs (Exercises 39–40), but it does contain all chordal graphs and all comparability graphs. (As observed in Proposition 5.3.25, when G has a transitive orientation, each induced subgraph inherits a transitive orientation, and the vertices with indegree 0 in this orientation form a stable set that meets all the maximal cliques.)

Our next class is a subclass of the strongly perfect graphs (Exercises 37–38) that still contains all chordal graphs and comparability graphs. Introduced by Chvátal [1984], it has played an important role in the theory of perfect graphs.

8.1.23. Definition. A **perfect order** on a graph is a vertex ordering such that greedy coloring with respect to the ordering inherited by each induced subgraph produces an optimal coloring of that subgraph. A **perfectly orderable graph** is a graph having a perfect order.

In an orientation of G, an **obstruction** is an induced 4-vertex path a, b, c, d whose first and last edges are oriented toward the leaves. The orientation of G *associated with* a vertex ordering L orients each edge toward the vertex earlier in L: $u \leftarrow v$ if $u < v$. A vertex ordering is **obstruction-free** if its associated orientation has no obstruction.

The orientation associated with a perfect order is obstruction-free, because on an obstruction the greedy coloring would use three colors instead of two. Chvátal proved that a graph is perfectly orderable if and only if it has an obstruction-free ordering. The characterization implies that perfectly orderable graphs are perfect and that chordal graphs and comparability graphs are perfectly orderable.

8.1.24. Example. *Chordal graphs and comparability graphs are perfectly orderable.* The orientation of a chordal graph associated with a simplicial construction ordering has no induced $u \leftarrow v \rightarrow w$. A transitive orientation of a comparability graph has no induced $u \rightarrow v \rightarrow w$.

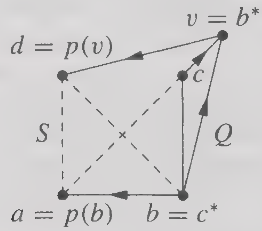
Every orientation with an obstruction has both an induced $u \rightarrow v \rightarrow w$ and an induced $u \leftarrow v \rightarrow w$. Hence if G is a comparability graph or a chordal graph, then G has an obstruction-free ordering. By Chvátal's characterization, such graphs are perfectly orderable. ■

8.1.25. Lemma. (Chvátal [1984]) Let G have a clique Q and a stable set S that are disjoint, and suppose that each $w \in Q$ is adjacent to some $p(w) \in S$. If

L is an obstruction-free ordering of G such that $p(w) < w$ for all $w \in Q$, then some $p(w) \in S$ is adjacent to all of Q.

Proof: We use induction on $n(G)$. For the basis step $n(G) = 1$, there is nothing to prove. Consider $n(G) > 1$. For each $w \in Q$, the graph $G - w$ satisfies the hypotheses using the clique $Q - w$ and the stable set $\{p(u) : u \in Q - w\}$. By the induction hypothesis, there is a vertex $w^* \in Q - w$ such that $p(w^*) \leftrightarrow Q - w$. We obtain $w \in Q$ such that $p(w^*) \leftrightarrow Q$ unless $p(w^*) \not\leftrightarrow w$ for all $w \in Q$. This assigns a unique w^* to every w, since $p(w^*)$ is nonadjacent only to w among Q. Mapping w to w^* thus defines a permutation on Q. Since $p(w) \leftrightarrow w$, the permutation has no fixed point.

We seek an obstruction in the orientation associated with L. Let v be the least vertex of Q in L. Let $b, c \in Q$ be the vertices such that $b^* = v$ and $c^* = b$ (possibly $c = v$). Let $a = p(b)$ and $d = p(v)$. Because $p(w^*) \not\leftrightarrow w$, we have $a \not\leftrightarrow c$ and $d \not\leftrightarrow b$, which implies $a \neq d$ in the stable set S and yields the picture below for the orientation associated with L.



Because $d = p(b^*)$, the only vertex of Q nonadjacent to d is b; thus $c \leftrightarrow d$. Since $d = p(v) < v \le c$ in L, we have $d \leftarrow c$. Now a, b, c, d induce an obstruction, which contradicts the hypothesis for L. Hence $p(w^*) \leftrightarrow w$ for some w, and $p(w^*)$ is the desired vertex of S. ■

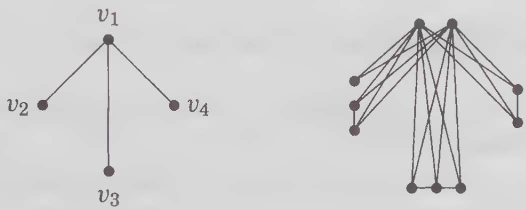
8.1.26. Theorem. (Chvátal [1984]) A vertex ordering of a simple graph G is a perfect order if and only if it is obstruction-free, and every graph with such an ordering is perfect.

Proof: We have observed that the condition is necessary. Since the class of graphs with obstruction-free orderings is hereditary (the inherited ordering for an induced subgraph is also obstruction-free), it suffices to show that the greedy coloring of G relative to an obstruction-free ordering L is optimal. Let k be the number of colors used by the greedy coloring relative to L. To prove optimality, we show that G has a k-clique; this also inductively proves perfection.

Let $f\colon V(G) \to [k]$ be the resulting coloring. Let i be the least integer such that G has a clique $w_{i+1}, \ldots, w_k$ such that $f(w_j) = j$. Since f uses color k on some vertex, i is well-defined. If $i = 0$, then G has a k-clique.

If $i > 0$, then for each w_j there is a vertex $p(w_j)$ such that $p(w_j) < w_j$ in L and $f(p(w_j)) = i$; otherwise the greedy coloring would use a lower color on w_j. Let $S = \{p(w_{i+1}), \ldots, p(w_k)\}$. Since all of S has the same color, S is a stable set. Hence the conditions of Lemma 8.1.25 are satisfied, and some vertex of S can be added to the clique to become w_i. This contradicts the minimality of i. ■

Next we consider a different way of generating perfect graphs. An operation that preserves perfection can enlarge a class of perfect graphs. Vertex multiplication, which expands each vertex into an independent set, is such a property. We generalize this. If $V(G) = \{v_1, \dots, v_n\}$, and $H_1, \dots, H_n$ are pairwise disjoint graphs, then the **composition** $G[H_1, \dots, H_n]$ is the graph $H_1 + \cdots + H_n$ together with $\{xy\colon x \in V(H_i), y \in V(H_j), v_iv_j \in E(G)\}$. The special case $G[\overline{K}_{h1}, \dots, \overline{K}_{hn}]$ is $G \circ h$. The example below uses $H_1 = 2K_1$, $H_2 = K_2 + K_1$, $H_3 = P_3$, $H_4 = K_2$, and $G = K_{1,3}$ with central vertex v_1.



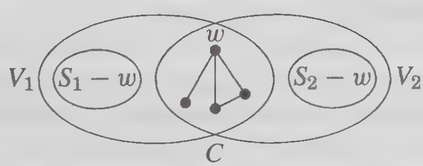
Lovász proved that composition preserves perfection. This is one corollary of Chvátal's Star-Cutset Lemma.

8.1.27. Definition. A **star-cutset** of G is a vertex cut S containing a vertex x adjacent to all of $S - \{x\}$. A **minimal imperfect graph** is an imperfect graph whose proper induced subgraphs are all perfect.

8.1.28. Lemma. (The Star-Cutset Lemma Lemma) If G has no stable set intersecting every maximum clique, and every proper induced subgraph of G is $\omega(G)$-colorable, then G has no star-cutset.

Proof: Suppose that G has a star-cutset C, with w adjacent to all of $C - \{w\}$. Since $G - C$ is disconnected, we can partition $V(G - C)$ into sets V_1, V_2 with no edge between them. Let $G_i = G[V_i \cup C]$, and let f_i be a proper $\omega(G)$-coloring of G_i. Let S_i be the set of vertices in G_i with the same color in f_i as w; this includes w but no other vertex of C. Since there are no edges between V_1 and V_2, the union $S = S_1 \cup S_2$ is a stable set.

If Q is a clique in $G - S$, then Q is contained in $G_1 - S_1$ or in $G_2 - S_2$. Since f_i provides an $\omega(G) - 1$-coloring of $G_i - S_i$, we have $|Q| \le \omega(G) - 1$. Since this applies to every clique Q in $G - S$, the stable set S meets every $\omega(G)$-clique of G, which contradicts the hypotheses. ■



8.1.29. Theorem. (The Star-Cutset Lemma, Chvátal [1985b]) No minimal imperfect graph has a star-cutset.

Proof: If G is a minimal imperfect graph, then $\chi(G) > \omega(G)$ and deletion of any stable set S leaves a perfect graph. Hence we have

$$1 + \omega(G) \le \chi(G) \le 1 + \chi(G - S) = 1 + \omega(G - S) \le 1 + \omega(G).$$

This yields $\omega(G - S) = \omega(G)$, which means that no stable set meets every maximum clique. Furthermore, since G is minimally imperfect, every proper induced subgraph G' satisfies $\chi(G') = \omega(G') \le \omega(G)$, making it $\omega(G)$-colorable. Lemma 8.1.28 now implies that G has no star-cutset. ■

The Replacement Lemma generalizes Lemma 8.1.4.

8.1.30. Corollary. (Replacement Lemma—Lovász [1972b]) Every composition of perfect graphs is perfect.

Proof: A composition can be constructed by a sequence of substitutions in which a single vertex v of G_1 is replaced by a graph G_2 and all edges added between $V(G_2)$ and $U = N_{G_1}(v)$ to form a graph G. Hence it suffices to prove that this operation preserves perfection. If the resulting graph G is not perfect, then it contains a minimal imperfect induced subgraph F. Such a subgraph cannot be contained in G_1 or G_2, which forces it to have at least two vertices of G_2 and at least one vertex of G_1.

If F has no vertex of G_1 outside U, then $F = F[U] \vee (F \cap G_2)$. The join operation preserves perfection, since $\chi(H \vee H') = \chi(H) + \chi(H')$ and $\omega(H \vee H') = \omega(H) + \omega(H')$ for all H, H'. Hence we may assume that F has a vertex of G_1 outside U. In this case, $V(F) \cap U$ together with one vertex of G_2 in F is a star-cutset of F. Hence the replacement of v with G_2 introduces no minimal imperfect subgraph F. ■

The Star-Cutset Lemma also yields perfection of weakly chordal graphs. Hayward [1985] proved that G or $\overline{G}$ has a star-cutset when G is a weakly chordal graph that is not a clique or stable set. With the Star-Cutset Lemma and the Perfect Graph Theorem, this implies that no weakly chordal graph is a minimal imperfect graph. Since the class is hereditary, it follows that every weakly chordal graph is perfect.

IMPERFECT GRAPHS

The **p-critical** graphs are the minimal imperfect graphs. The Strong Perfect Graph Conjecture (SPGC) states that the only p-critical graphs are the odd cycles (of length at least 5) and their complements. With enough properties of p-critical graphs, perhaps we could prove that only odd cycles and their complements have all these properties; this would prove the SPGC. We begin with simple observations about p-critical graphs, some used earlier in discussing star-cutsets. (This presentation was originally modeled after Shmoys [1981].)

8.1.31. Lemma. If G is p-critical, then G is connected, $\overline{G}$ is p-critical, $\omega(G) \geq 2$, and $\alpha(G) \geq 2$. Furthermore, for every $x \in V(G)$, $\chi(G - x) = \omega(G)$ and $\theta(G - x) = \alpha(G)$.

Proof: G is perfect if and only if every component of G is perfect, and G is perfect if and only if $\overline{G}$ is perfect. Cliques and their complements are perfect. Finally, we observed in proving Theorem 8.1.29 that deleting a stable set from a p-critical graph cannot decrease the clique number. Since $G - x$ is perfect, we thus have $\chi(G - x) = \omega(G - x) = \omega(G)$. The condition $\theta(G - x) = \alpha(G)$ is this statement for $\overline{G}$. ■

More subtle properties of p-critical graphs follow from Lovász's extension of the PGT.

8.1.32. Theorem. (Lovász [1972b]) A graph G is perfect if and only if $\omega(G[A])\alpha(G[A]) \geq |A|$ for all $A \subseteq V(G)$. ■

The property "$\omega(G[A])\alpha(G[A]) \geq |A|$ for all $A \subseteq V(G)$" was suggested by Fulkerson; we call it **β-perfection**. It is implied by α-perfection or γ-perfection; if we can color G with $\omega(G)$ stable sets, then some stable set has at least $n(G)/\omega(G)$ vertices. The converse involves counting arguments like those we gave for the PGT, but more delicate. Since β-perfection is unchanged under complementation, Theorem 8.1.32 immediately implies the PGT.

8.1.33. Theorem. If G is p-critical, then $n(G) = \alpha(G)\omega(G) + 1$. Furthermore, for every $x \in V(G)$, $G - x$ has a partition into $\omega(G)$ stable sets of size $\alpha(G)$ and a partition into $\alpha(G)$ cliques of size $\omega(G)$.

Proof: When G is p-critical, the condition for β-perfection fails only for the full vertex set $A = V(G)$. Hence for each $x \in V(G)$ we have

$$n(G) - 1 \leq \alpha(G - x)\omega(G - x) = \alpha(G)\omega(G) \leq n(G) - 1.$$

Therefore, $n(G) = \alpha(G)\omega(G) + 1$. Since $\chi(G - x) = \omega(G - x) = \omega(G)$, we can cover $G - x$ by $\omega(G)$ stable sets. Having size at most $\alpha(G)$, these sets partition the $\alpha(G)\omega(G)$ vertices of $G - x$ into $\omega(G)$ stable sets of size $\alpha(G)$. Similarly, $\theta(G - x) = \alpha(G - x) = \alpha(G)$ yields a partition of $V(G - x)$ into $\alpha(G)$ cliques of size $\omega(G)$. ■

Study of p-critical graphs has benefitted by enlarging the class to include other graphs satisfying the properties in Theorem 8.1.33. Structural properties of the larger class are useful when proving the SPGC for special classes of graphs. Padberg [1974] began the study of these graphs. Several definitions were suggested to extend the class of p-critical graphs but turned out to be alternative characterizations of the same class. The definition we use originates in Bland–Huang–Trotter [1979].

8.1.34. Definition. For integers $a, w \geq 2$, a graph G is **a, w-partitionable** if it has $aw + 1$ vertices and for each $x \in V(G)$ the subgraph $G - x$ has a partition into a cliques of size w and a partition into w stable sets of size a.

8.1.35. Theorem. (Buckingham–Golumbic [1983]) A graph G of order $aw+1$ is a,w-partitionable if and only if $\chi(G-x)=w$ and $\theta(G-x)=a$ for every $x\in V(G)$. Furthermore, $\omega(G)=w$ and $\alpha(G)=a$ for such graphs, and the inequalities $\chi(G-x)\le w$ and $\theta(G-x)\le a$ suffice.

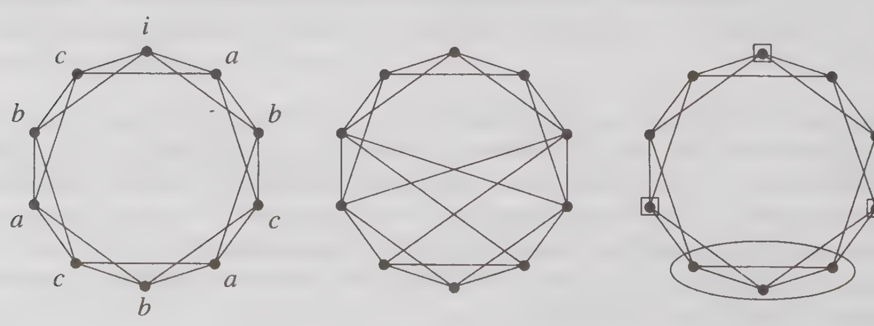
Proof: Let G be partitionable. Since $G-x$ is w-colorable and has a w-clique, $\chi(G-x)=w=\omega(G-x)$. Since $a\ge 2$, G is not a complete graph. Deleting a vertex x outside a maximum clique Q in G yields $\omega(G)=\omega(G-x)=w$. The same arguments for $\overline{G}$ yield the results for a.

Conversely, suppose that $\chi(G-x)\le w$ and $\theta(G-x)\le a$ for every x in $V(G)$. The latter inequality yields $\alpha(H)\le a$. Hence an optimal coloring of $G-x$ uses at most w stable sets of size at most a. Since $n(G-x)=aw$, such a coloring partitions $V(G-x)$ into w stable sets of size a. Similarly, a covering of $G-x$ by a cliques yields the desired clique partition. ■

By Theorem 8.1.33 and Theorem 8.1.35, every p-critical graph is partitionable and every partitionable graph is imperfect. Furthermore, G is a,w-partitionable if and only if $\overline{G}$ is w,a-partitionable.

8.1.36. Example. *Cycle-powers.* The graph C_n^d is constructed by placing n vertices on a circle and making each vertex adjacent to the d nearest vertices in each direction on the circle. When $d=1$, $C_n^d=C_n$. We view the vertices as the integers modulo n, in order. The graph C_{10}^2, shown on the left below, is neither perfect nor p-critical (the vertices $0,2,4,6,8$ induce C_5), but C_{10}^2 is 3,3-partitionable. When i is removed, the unique partition of the remaining nine vertices into three triangles is $\{(i+1,i+2,i+3),(i+4,i+5,i+6),(i+7,i+8,i+9)\}$, and the unique partition into three stable sets is $\{(i+1,i+4,i+7),(i+2,i+5,i+8),(i+3,i+6,i+9)\}$.

Always C_{aw+1}^{w-1} is a,w-partitionable. Every w consecutive vertices in $G-x$ form a clique, and every a vertices spaced by jumps of length w form a stable set. Showing that C_{aw+1}^{w-1} is p-critical if and only if $w=2$ or $a=2$ reduces the SPGC to the statement that G is p-critical if and only if $G=C_{\alpha(G)\omega(G)+1}^{\omega(G)-1}$. ■



8.1.37. Example. *Other partitionable graphs.* Other partitionable graphs arise by adding unimportant edges to C_{aw+1}^{w-1}. In C_{10}^2, we can add any diagonal without changing the set of maximum cliques or the set of maximum stable sets, so the resulting graph is still partitionable. We will see that the SPGC would follow if all partitionable graphs came from cycle-powers by adding unimportant edges of this type.

Nevertheless, there are other partitionable graphs, such as the graph in the middle above (Chvátal–Graham–Perold–Whitesides [1979], Huang [1976]). Every edge in this graph belongs to a maximum clique, but it has two more edges than C_{10}^2. The partitions demonstrating that it is partitionable differ from those used for C_{10}^2 (Exercise 42). ■

8.1.38. Example. *Further properties of C_{aw+1}^{w-1}.* The graph C_{aw+1}^{w-1} has exactly n maximum cliques, each using w consecutive vertices on the cycle. Each vertex lies in w consecutive w-cliques. There are also exactly n maximum stable sets, each having $a-1$ gaps of length w and one gap of length $w+1$ between successive vertices. A maximum stable set containing x has a places for the larger gap, so each vertex x lies in a maximum stable sets.

Finally, a w-clique can avoid a maximum stable set only by fitting inside the gap of length $w+1$ (shown above Example 8.1.37 on the right). Thus there is a pairing $\{(Q_i, S_i)\}$ between the maximum stable sets and maximum cliques such that $Q_i \cap S_j = \varnothing$ if and only if $i = j$. ■

These "further properties" comprise the next characterization. The arguments are due to Padberg [1974], who used them in a polyhedral characterization of perfect graphs. Here combinatorial conclusions follow from properties of matrices in linear algebra. Other characterizations of partitionable graphs appeared in Bland–Huang–Trotter [1979], Golumbic [1980, p58-62], Tucker [1977], Chvátal–Graham–Perold–Whitesides [1979], and Buckingham [1980].

8.1.39. Theorem. A graph G of order $n = aw+1$ is a, w-partitionable if and only if both conditions below hold:

1) $\alpha(G) = a$ and $\omega(G) = w$, and each vertex of G belongs to exactly w cliques of size w and a stable sets of size a.

2) G has exactly n maximum cliques $\{Q_i\}$ and exactly n maximum stable sets $\{S_j\}$, with $Q_i \cap S_j = \varnothing$ if and only if $i = j$ (Q_i and S_i are **mates**).

Proof: *Necessity.* We have proved $\chi(G-x) = w = \omega(G)$ and $\theta(G-x) = a = \alpha(G)$ for each $x \in V(G)$. Choose a clique Q of size w. For each $x \in Q$, $G-x$ has a partition into a cliques of size w. Together, Q and these w partitions form a list of $n = aw+1$ maximum cliques $Q_1, \ldots, Q_n$. Each vertex outside Q appears in one clique in each partition. Each vertex in Q appears in Q and once in $w-1$ partitions. Hence every vertex appears in exactly w cliques in the list.

For each Q_i, we obtain a maximum stable set S_i disjoint from Q_i. Choose $x \in Q_i$. The w maximum stable sets that partition $V(G-x)$ can meet Q_i only at the $w-1$ vertices other than x. Therefore, one of these stable sets misses Q_i; call it S_i. We will show that these two lists contain all the cliques and stable sets and have the desired intersection properties.

Let A be the incidence matrix with $a_{i,j} = 1$ if $x_j \in Q_i$ and $a_{i,j} = 0$ otherwise. Let B be the matrix with $b_{i,j} = 1$ if $x_j \in S_i$ and $b_{i,j} = 0$ otherwise. The ijth entry of AB^T is the dot product of row i of A with row j of B, which equals $|Q_i \cap S_j|$. By proving that $AB^T = J - I$, where J is the matrix of all 1s, we obtain $Q_i \cap S_j \neq \emptyset$ if and only if $i \neq j$. Since $J - I$ is nonsingular, this will also imply that A and B are nonsingular. Nonsingular matrices have distinct rows, and hence $Q_1, \ldots, Q_n$ and $S_1, \ldots, S_n$ will be distinct.

By construction, $|Q_i \cap S_i| = 0$. Since cliques and stable sets intersect at most once, to prove that $AB^T = J - I$ we need only show that each column of AB^T sums to $n-1$. Multiplying by the row vector $\mathbf{1}_n^T$ on the left computes these sums. We constructed A so that each column has w 1s (because each vertex appears in w cliques in the list) and B so that each row has a 1s (because each stable set has size a). Therefore,

$$\mathbf{1}_n^T(AB^T) = (\mathbf{1}_n^T A)B^T = w\mathbf{1}_n^T B^T = wa\mathbf{1}_n = (n-1)\mathbf{1}_n^T.$$

To prove that G has no other maximum cliques, we let q be the incidence vector of a maximum clique Q and show that q must be a row of A. Since A is nonsingular, its rows span $\mathbb{R}^n$, and we can write q as a linear combination: $q = tA$. To solve for t, we need A^{-1}. Since every row of A sums to ω, we have $A(\omega^{-1}J - B^T) = \omega^{-1}\omega J - (J - I) = I$, and hence $A^{-1} = \omega^{-1}J - B^T$. Thus,

$$t = qA^{-1} = q(\omega^{-1}J - B^T) = \omega^{-1}qJ - qB^T = \omega^{-1}\omega\mathbf{1}_n^T - qB^T.$$

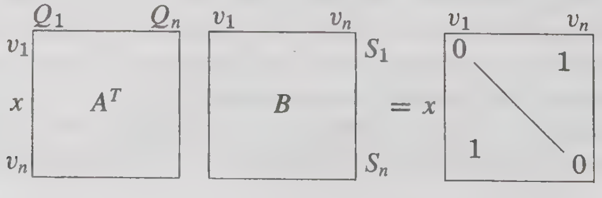
The ith column of B^T is the incidence vector of S_i; hence coordinate i of qB^T equals $|Q \cap S_i|$, which is 0 or 1. Hence t is a 0,1-vector and q is a sum of rows of A. Since q sums to ω, only one row can be used. Thus q is a row of A and $Q_1, \ldots, Q_n$ are the only maximum cliques.

The same argument applied to $\overline{G}$ shows that G has exactly n maximum stable sets, with each vertex appearing in a of them.

Sufficiency. By Theorem 8.1.35, we need only prove that $\chi(G - x) \le w$ and $\theta(G - x) \le a$ for all $x \in V(G)$. Given the cliques and stable sets as guaranteed by condition (2), define the incidence matrices A, B as above. By condition (1), each column of B has a 1s, and hence $JB = aJ = BJ$. The intersection requirements in condition (2) yield $AB^T = J - I$. This is nonsingular, so B is nonsingular and

$$A^T B = B^{-1}BA^T B = B^{-1}(J - I)B = B^{-1}BJ - I = J - I.$$

In the product $A^T B = J - I$, the row corresponding to $x \in V(G)$ states that $V(G - x)$ is covered by the mates of the w maximum cliques containing x (illustrated below), and hence $\chi(G - x) \le w$. Similarly, the column corresponding to x states that $V(G - x)$ is covered by the mates of the a maximum stable sets containing x, and hence $\theta(G - x) \le a$. ∎



8.1.40. Corollary. If G is a, w-partitionable and $w = 2$, then $G = C_{2a+1}$; if $a = 2$, then $G = \overline{C}_{2w+1}$. Hence the SPGC reduces to showing that p-critical graphs have $\omega = 2$ or $\alpha = 2$.

Proof: If $\omega = 2$, then every vertex belongs to exactly two cliques of size 2, so G is 2-regular. Furthermore, G is connected and has odd order $(2\alpha + 1)$, so G is an odd cycle. For $a = 2$, consider $\overline{G}$. ∎

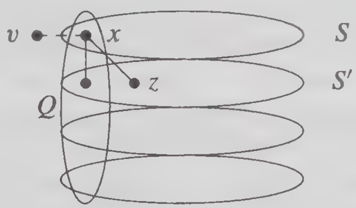
Henceforth we use $w, \omega(G), \omega$ interchangeably and $a, \alpha(G), \alpha$ interchangeably for partitionable graphs.

8.1.41. Theorem. (Tucker [1977]) Let x be a vertex in a partitionable graph G. The subgraph $G - x$ has a unique minimum coloring; denoted $\mathrm{X}(G - x)$, it consists of the mates of the maximum cliques containing x. Similarly, $G - x$ has a unique minimum clique covering $\mathrm{X}(G - x)$ consisting of the mates of the maximum stable sets containing x.

Proof: Since G is a, w-partitionable, $G - x$ is w-colorable using w stable sets of size a. Every w-clique containing x misses some color class, since the clique has only $w - 1$ vertices in $G - x$. Thus all w-cliques containing x have mates as color classes in the coloring. There are exactly w of these, so the coloring is unique. The other statement follows by complementation. ∎

8.1.42. Theorem. (Buckingham–Golumbic [1983]) If x is a vertex of an α, ω-partitionable graph G, then $2\omega - 2 \le d(x) \le n - 2\alpha + 1$.

Proof: Select a vertex $v \not\leftrightarrow x$ (see illustration above). Let S be the stable set in $\mathrm{X}(G - v)$ that contains x, and let S' be another stable set in $\mathrm{X}(G - v)$. Choose $z \in N(x) \cap S_2$. In $\Theta(G - z)$, some clique Q contains x. Since $v \not\leftrightarrow x$, Q has one vertex in each stable set of $\mathrm{X}(G - v)$, including S'. Since $Q \in \Theta(G - z)$ implies $z \notin Q$, this yields a second neighbor of x in S'. Thus x has at least two neighbors in each of the $\omega - 1$ stable sets in $\mathrm{X}(G - v)$, yielding $d(x) \ge 2\omega - 2$. The same argument in $\overline{G}$ yields $n - 1 - d(x) = |N_{\overline{G}}(x)| \ge 2\alpha - 2$. ∎



These bounds on vertex degrees in α, ω-partitionable graphs are sharp, as they hold with equality for powers of cycles.

8.1.43. Definition. An edge of a graph is **critical** if deleting it increases the independence number. A pair of nonadjacent vertices is **co-critical** if adding it increases the clique number.

The characterization of critical edges in partitionable graphs is implicit in the work of several authors.

8.1.44. Theorem. For an edge xy in a partitionable graph G, the following statements are equivalent.
A) xy is a critical edge.
B) $S \cup \{x\} \in \mathrm{X}(G - y)$.
C) xy belongs to $\omega - 1$ maximum cliques.

Proof: B $\Rightarrow$ A. $S \cup \{x, y\}$ is a stable set of size $\alpha + 1$ in $G - xy$.

A $\Rightarrow$ C. If xy is critical, then there is a set S such that $S \cup \{x\}$ and $S \cup \{y\}$ are maximum stable sets in G. Hence every maximum clique containing x but not y is disjoint from $S \cup \{y\}$. Since there are ω maximum cliques containing x and only one maximum clique disjoint from $S \cup \{y\}$, the remaining $\omega - 1$ maximum cliques containing x must also contain y.

C $\Rightarrow$ B. The stable sets in the unique coloring of $G - x$ are the mates of the cliques containing x. Since xy belongs to $\omega - 1$ maximum cliques, the mates of these $\omega - 1$ cliques belong to both $\mathrm{X}(G - x)$ and $\mathrm{X}(G - y)$. This leaves only $a + 1$ vertices in the graph, consisting of the vertices x, y and a stable set S such that $S \cup \{y\} \in \mathrm{X}(G - x)$ and $S \cup \{x\} \in \mathrm{X}(G - y)$. ■

8.1.45. Corollary. Let G be a partitionable graph. If xy is an edge appearing in no maximum clique, then $G - xy$ is partitionable. If x, y is a nonadjacent pair appearing in no maximum stable set, then $G + xy$ is partitionable.

Proof: By complementation, we need only prove the first statement. If we delete an edge appearing in no maximum clique, then by Theorem 8.1.44 it is not a critical edge, and we have $\omega(G - xy) = \omega(G)$ and $\alpha(G - xy) = \alpha(G)$. Since we have not destroyed any maximum clique and have not created a bigger stable set, we can use the optimal coloring and clique partition of $G - u$ to conclude that $\chi(G - xy - u) \le \omega$ and $\theta(G - xy - u) \le \alpha$. Hence $G - xy$ is partitionable, by Theorem 8.1.35. ■

The discussion in Example 8.1.37 suggests that edges appearing in no maximum clique are uninteresting "junk". Corollary 8.1.45 assures us that "junk is junk". The partitionable cycle-powers have no junk.

THE STRONG PERFECT GRAPH CONJECTURE

We have been proving properties of partitionable graphs in a "top down" approach to the SPGC, trying to find enough properties to eliminate all but odd cycles and their complements as p-critical graphs. The "bottom up" approach is to verify that the SPGC holds on larger and larger classes of graphs, until all are included.

8.1.46. Definition. An **odd hole** or **odd antihole** in G is an induced subgraph of G that is C_{2k+1} or $\overline{C}_{2k+1}$ (for some $k \ge 2$), respectively. A graph having no odd hole or antihole is a **Berge graph**.

One way to prove that a class **G** satisfies the SPGC is to prove that every Berge graph in **G** is perfect. A hereditary class **G** satisfies the SPGC if the odd cycles and their complements are the only p-critical graphs in **G**.

The SPGC holds for planar graphs (Tucker [1973]), toroidal graphs (Grinstead [1981]), graphs with $\Delta(G) \le 6$ (Grinstead [1978]) or $\omega(G) \le 3$ (Tucker [1977]), and for various classes defined by forbidding fixed small induced subgraphs (Meyniel [1976], Tucker [1977], Parthasarathy–Ravindra [1976, 1979], Chvátal–Sbihi [1988], Olariu [1989], Sun [1991]). We consider three families.

8.1.47. Definition. A **circular-arc graph** is the intersection graph of a family of arcs of a circle. A **circle graph** is the intersection graph of a family of chords of a circle. A $K_{1,3}$**-free** graph is a graph not having $K_{1,3}$ as an induced subgraph.

Every cycle is both a circle graph and a circular-arc graph, but neither of these classes contains the other (Exercise 47).

One way to prove the SPGC for a class **G** is to show that every partitionable graph in **G** belongs to another class **H** where the SPGC is known to hold. In this role we use the class $\{C_n^d\}$.

8.1.48. Theorem. (Chvátal [1976]) Cycle-powers satisfy the SPGC. In particular, the graph C_{aw+1}^{w-1} is p-critical if and only if $w = 2$ or $a = 2$, in which case the graph is an odd hole or antihole.

Proof: It suffices to consider the partitionable graph $G = C_{aw+1}^{w-1}$. This is p-critical when $a = 2$ or $w = 2$, so we may assume $a, w > 2$. Let the vertices be $\{v_0, \dots, v_{aw}\}$, and let $S = \{v_{iw+1}, v_{(i+1)w} \colon 0 \le i \le a-1\}$. The subgraph $G[S]$ is a cycle, since the indices of consecutive vertices in S are separated by 1 or $w-1$ (except that v_{aw} and v_1 are separated by 2), and indices of nonconsecutive vertices differ by at least w. To obtain C_{2a-1} as a proper induced subgraph, we replace $\{v_{(a-1)w}, v_{aw}, v_1, v_w\}$ with $\{v_{(a-1)w+1}, v_0, v_{w-1}\}$ in S. We conclude that G is not p-critical. ■

8.1.49. Theorem. (Tucker [1975]) The SPGC holds for circular-arc graphs.

Proof: Recall that $N[v]$ denotes $N(v) \cup \{v\}$, the closed neighborhood of v (Definition 3.1.29). When G is partitionable with distinct vertices x, y, we claim that $N[x] \not\subseteq N[y]$. Consider the clique Q containing x in $\Theta(G-y)$; we have $Q \subseteq N[x]$. If $N[y]$ contains $N[x]$, then $Q \cup \{y\}$ is a clique of size $\omega(G)+1$.

Now, if G is a partitionable circular-arc graph, it suffices to show that $G = C_n^{\omega(G)-1}$, because the SPGC holds for cycle-powers (Theorem 8.1.48). Consider a circular-arc representation that assigns arc A_x to $x \in V$. Since $N[y]$ cannot contain $N[x]$, the arc A_x cannot lie within another arc A_y of the representation. If no arc contains another, then every arc that intersects A_x contains exactly one of its endpoints. Since the vertices corresponding to the arcs containing one point induce a clique, there are at most $\omega - 1$ other arcs containing each endpoint of A_x. Equality holds, since Theorem 8.1.42 requires $\delta(G) \ge 2\omega - 2$.

Starting from a given point p on the circle, let v_i be the vertex represented by the ith arc encountered moving clockwise from p. Since each arc meets exactly $\omega - 1$ others at each endpoint, v_i is adjacent to $v_{i+1}, \ldots, v_{i+\omega-1}$ (addition modulo n) for each i. Hence $G = C_n^{\omega-1}$. ■

The original proof of the SPGC for claw-free graphs (Parthasarathy–Ravindra [1976]) was quite intricate. Further study of p-critical graphs has shortened both it and the proof of the next theorem, which we will apply.

8.1.50. Theorem. (Giles–Trotter–Tucker [1984]) If a partitionable graph G has a cycle consisting of critical edges, then the subgraph G' obtained by deleting the edges belonging to no maximum clique is $C_n^{\omega-1}$.

Proof: (Hartman [1995]) Suppose that G is a, w-partitionable. Deleting edges destroys no stable set. Deleting edges in no maximum clique destroys no maximum clique. Hence the coloring and clique covering of $G - x$ also yield $\chi(G' - x) \le w$ and $\theta(G' - x) \le a$ (regardless of whether $\alpha(G') > \alpha(G)$). By Theorem 8.1.35, G' is thus a, w-partitionable. Also, the clique coverings of $G' - x$ for various x force G' to be connected.

We next prove that if G has a u, v-path consisting of k critical edges, then u and v belong to at least $\omega - k$ common maximum cliques. We use induction on k, with Theorem 8.1.44 providing the basis step, $k = 1$. For $k > 1$, if y is the vertex before v on such a path, then the induction hypothesis puts u and y in $\omega - k + 1$ common maximum cliques. Since y belongs to exactly ω maximum cliques (by Theorem 8.1.39), and $\omega - 1$ of these contain v (by Theorem 8.1.44), at most one of the $\omega - k + 1$ cliques containing u and y can omit v.

Let C be a cycle of critical edges in G. Critical edges belong to maximum cliques, so C remains in G'. As shown above, ω vertices forming a path in G' induce a maximum clique in G'. If the length of C exceeds ω, then this establishes ω successive maximum cliques containing a given vertex x of C. By Theorem 8.1.39, these are all the maximum cliques of G containing x, and hence they include all the edges of G' incident to x. Hence C is a component of G', but G' is connected, so C contains all vertices of G'. This expresses G' as $C_n^{\omega-1}$.

If the length of C is at most ω, then $V(C)$ itself is a clique. If $x \in V(C)$, then the vertices of $C - x$ belong to distinct stable sets in the coloring $\mathrm{X}(G - x)$ defined by Theorem 8.1.41. Let $x_0, \ldots, x_k$ be the vertices of C in order. Let $S_1, \ldots, S_k$ be the stable sets in $G - V(C)$ such that $S_i \cup \{x_i\} \in \mathrm{X}(G - x_0)$. Because $x_i x_{i+1}$ is a critical edge, x_i and x_{i+1} belong to $\omega - 1$ common maximum cliques (Theorem 8.1.44), and hence by Theorem 8.1.41 the colorings $\mathrm{X}(G - x_i)$ and $\mathrm{X}(G - x_{i+1})$ have $\omega - 1$ common stable sets. The remaining set differs only in having x_i or x_{i+1}. Hence $\mathrm{X}(G - x_1)$ contains $S_i \cup \{x_i\}$ for $i \ge 2$, and it also contains $S_1 \cup \{x_0\}$.

Continuing these substitutions while following the edges of C, we find that $\mathrm{X}(G - x_k)$ contains $S_i \cup \{x_{i-1}\}$ for $1 \le i \le k$. Taking one more step to return to x_0, we find that $\mathrm{X}(G - x_0)$ contains $S_i \cup \{x_{i-1}\}$ for $2 \le i \le k$ and contains $S_1 \cup \{x_k\}$. Since $k \ge 2$ and $\alpha \ge 2$, these sets are different from our initial sets in $\mathrm{X}(G - x_0)$. Since the coloring $\mathrm{X}(G - x_0)$ is unique, we have obtained a contradiction, and the case $n(C) \le \omega$ does not arise. ■

8.1.51. Theorem. (Chvátal [1976]) If G is a p-critical graph such that the spanning subgraph G' obtained by deleting the edges of G belonging to no maximum clique is a cycle-power C_n^d, then G is an odd hole or odd antihole (and equals G').

Proof: A p-critical graph is partitionable. The stable sets and maximum cliques in G are stable sets and cliques in G', and by Theorem 8.1.35 we again conclude that G' is partitionable with $\alpha(G') = \alpha(G) = a$ and $\omega(G') = \omega(G) = w$. Hence $G' = C_{aw+1}^{w-1}$. We index the vertices so that the maximum cliques of G' (and G) consist of w cyclically consecutive vertices, and the maximum stable sets have the form $v_i, v_{i+w}, \ldots, v_{i+aw}$. In particular, vertices separated by a multiple of w on the cycle $v_0, \ldots, v_{aw}$ are nonadjacent in G' and in the full graph G.

If $G' = G$, then Theorem 8.1.48 implies that G is an odd hole or odd antihole. If $G' \neq G$, then $a, w > 2$, since otherwise deleting an edge increases the number of maximum stable sets or decreases the number of maximum cliques.

For $a, w \geq 3$, we exhibit an imperfect proper induced subgraph H of G (the induced odd cycle in G' obtained in Theorem 8.1.48 may have a chord in G). Let $S = \{v_{aw}, v_1, v_w, v_{w+2}\} \cup \{v_{iw+1}\colon 2 \leq i \leq a-1\}$, and let $T = \{v_{(a-1)w+1}, v_{aw}, v_1, v_w\} \cup \{v_{w+i}\colon 2 \leq i \leq w-1\}$. The sets S and T have sizes $a+2$ and $w+2$, and for $a, w \geq 3$ they share exactly the five vertices $\{v_{(a-1)w+}, v_{aw}, v_1, v_w, v_{w+2}\}$. Furthermore, S intersects every maximum clique of G' (and hence of G), and T intersects every maximum stable set of G' (and hence of G) (Exercise 49). Letting $H = G - (S \cup T)$, this yields $\alpha(H) = a - 1$ and $\omega(H) = w - 1$. Now imperfection follows from

$$n(H) \geq n(G) - (a + w + 4 - 5) > (a - 1)(w - 1). \qquad \blacksquare$$

8.1.52. Corollary. (Giles–Trotter–Tucker [1984]) If G is a p-critical graph and for each $v \in V(G)$ the minimum coloring $\mathrm{X}(G - v)$ has (at least) two sets that each contain exactly one neighbor of v, then G is an odd hole or an odd antihole.

Proof: When some set in $\mathrm{X}(G - v)$ has exactly one neighbor u of v, the edge uv is critical. Hence the hypothesis implies that the subgraph of critical edges has minimum degree at least 2 and therefore contains a cycle. By Theorem 8.1.50, the subgraph G' obtained by deleting the edges belonging to no maximum clique is $C_n^{\omega-1}$. By Theorem 8.1.51, G is an odd hole or an odd antihole. ■

8.1.53. Corollary. (Parthasarathy–Ravindra [1976]) The SPGC holds for $K_{1,3}$-free graphs.

Proof: (Giles–Trotter–Tucker [1984]) Let G be a p-critical $K_{1,3}$-free graph. For each $v \in V(G)$, $N(v)$ induces a perfect subgraph having no stable set of size 3. This means that $N(v)$ can be covered by two cliques, which implies $d(v) \leq 2\omega(G) - 2$. Each of the $\omega(G)$ stable sets in $\mathrm{X}(G - v)$ contains a neighbor of v, else adding v creates a larger stable set. With $d(v) \leq 2\omega(G) - 2$, at least two of these sets have exactly one neighbor of v. Hence G satisfies the hypothesis of Corollary 8.1.52, and G is an odd hole or antihole. ■

Corollary 8.1.53 also yields the SPGC for circle graphs (Exercise 50). The general SPGC remains open, but a result intermediate between it and the PGT is known (it is immediately implied by the SPGC and immediately implies the PGT). Chvátal conjectured that if G and H have the same vertex set and have the same 4-tuples of vertices that induce P_4, then G is perfect if and only if H is perfect. Reed [1987] proved this "Semi-Strong Perfect Graph Theorem".

EXERCISES

8.1.1. (−) Compute $\chi(G)$ and $\omega(G)$ for the complement of the odd cycle C_{2k+1}.

8.1.2. (−) Determine the smallest imperfect graph G such that $\chi(G) = \omega(G)$.

8.1.3. (!) P_4-free graphs are also called **cographs**, which stands for "complement reducible". A graph is **complement reducible** if it can be reduced to an empty graph by successively taking complements within components.

a) Prove that a graph G is P_4-free if and only if it is complement reducible.

b) Use part (a) and the Perfect Graph Theorem to prove that every P_4-free graph is perfect. (Seinsche [1974])

8.1.4. *Clique identification.* Suppose that $G = G_1 \cup G_2$, that $G_1 \cap G_2$ is a clique, and that G_1 and G_2 are perfect. Without using the Star-cutset Lemma, prove that G is perfect.

8.1.5. Find an imperfect graph G having a star-cutset C such that the C-lobes of G are perfect graphs. (Comment: Thus identification at star-cutsets does not preserve perfection, although no p-critical graph has a star-cutset.)

8.1.6. Let G be a cartesian product of complete graphs. Prove that $\alpha(G) = \theta(G)$. Prove that $K_2 \square K_2 \square K_3$ is not perfect.

8.1.7. Prove that $C_5 \vee K_1$ is the only color-critical 4-chromatic graph with six vertices.

8.1.8. (+) Prove that G is an odd cycle if and only if $\alpha(G) = (n(G) - 1)/2$ and $\alpha(G - u - v) = \alpha(G)$ for all $u, v \in V(G)$. (Melnikov–Vizing [1971], Greenwell [1978])

8.1.9. Let $v_1, \dots, v_n$ be a simplicial elimination ordering of G, and let $Q(v_i) = \{v_j \in N(v_i)\colon j > i\}$. Note that $Q(v_i)$ is the clique of neighbors of v_i at the time when v_i is deleted in the elimination ordering. Let $S = \{y_1, \dots, y_k\}$ be the stable set obtained "greedily" from the ordering $v_1, \dots, v_n$; that is, set $y_1 = v_1$, discard $N(y_1)$ from the remainder of the ordering, and proceed iteratively, at each step adding the least remaining element x to the stable set and discarding what remains of $Q(x)$.

a) Prove that applying the greedy coloring algorithm to the construction ordering $v_n, \dots, v_1$ yields an optimal coloring and that $\omega(G) = 1 + \max \sum_{x \in V(G)} |Q(x)|$. (Fulkerson–Gross [1965])

b) Prove that S is a maximum stable set and that the sets $\{y_i\} \cup Q(y_i)$ form a minimum clique covering. (Gavril [1972])

8.1.10. Add a test to the MCS algorithm to check whether the resulting ordering is a simplicial elimination ordering. (Tarjan–Yannakakis [1984])

8.1.11. Prove directly (without using a simplicial elimination ordering) that the intersection graph of a family of subtrees of a tree has no chordless cycle.

8.1.12. (−) Prove that every graph is the intersection graph of a family of subtrees of some graph.

8.1.13. Prove that every chordal graph has an intersection representation by subtrees of a host tree with maximum degree 3.

8.1.14. Let Q be a maximal clique in a connected chordal graph G. For all $x \in V(G)$, prove that Q has two vertices whose distances from x are different. (Voloshin [1982])

8.1.15. *Intersection graphs of subtrees of a graph.* A **fraternal orientation** of a graph is an orientation such that any pair of vertices with a common successor are adjacent.

a) (−) Prove that a graph is chordal if and only if it has an acyclic fraternal orientation.

b) (−) Obtain a graph with no fraternal orientation.

c) A family of trees in a graph is *rootable* if the trees can be assigned roots so that a pair of them intersects if and only if at least one of the two roots belongs to both subtrees. Prove that G has a fraternal orientation if and only if G is the intersection graph of a rootable family of subtrees of some graph. (Gavril–Urrutia [1994])

8.1.16. (!) Prove that a simple graph G is a forest if and only if every pairwise intersecting family of paths in G has a common vertex. (Hint: For sufficiency, use induction on the number of paths in the family.)

8.1.17. (!) *Forbidden subgraph characterization of split graphs.* A graph is a **split graph** if its vertices can be partitioned into a clique and a stable set.

a) Prove that if G is a split graph, then G and $\overline{G}$ are chordal graphs. Observe that if G and $\overline{G}$ are chordal graphs, then G has no induced subgraph in $\{C_4, 2K_2, C_5\}$.

b) Prove that if G is a simple graph with no induced subgraph in $\{C_4, 2K_2, C_5\}$, then G is a split graph. (Hint: Among the maximum-sized cliques, let Q be one such that $G - Q$ has the minimum number of edges. Prove that $G - Q$ is a stable set, using the choice of Q and the forbidden subgraph conditions.) (Hammer–Simeone [1981])

8.1.18. Let $d_1 \geq \cdots \geq d_n$ be the degree sequence of a simple graph G, and let m be the largest value of k such that $d_k \geq k - 1$. Prove that G is a split graph if and only if $\sum_{i=1}^{m} d_i = m(m-1) + \sum_{i=m+1}^{n} d_i$. (Comment: Compare with Exercise 3.3.28.) (Hammer–Simeone [1981])

8.1.19. (−) Determine the trees that are split graphs, and construct a pair of nonisomorphic split graphs with the same degree sequence.

8.1.20. The k**-trees** are the graphs that arise from a k-clique by 0 or more iterations of adding a new vertex joined to a k-clique in the old graph. Prove that G is a k-tree if and only if G satisfies the following three properties:

1) G is connected.
2) G has a k-clique but no $k+2$-clique.
3) Every minimal vertex separator of G is a k-clique.

8.1.21. Let G be an n-vertex chordal graph having no clique of order $k+2$. Prove that $e(G) \leq kn - \binom{k+1}{2}$, with equality if and only if G is a k-tree.

8.1.22. (+) Generalize Theorem 2.2.3 (Cayley's Formula) by proving that the number of k-trees with vertex set $[n]$ is $\binom{n}{k}[k(n-k)+1]^{n-k-2}$. (Hint: Generalize the Prüfer code for *rooted* trees, which generates a list with $n-1$ entries and never deletes the root. In a k-tree, the vertices belonging to exactly one $k+1$-clique are the *leaves*. A k-tree can be grown using any k-clique as a root. The lists generated from k-trees with a fixed root have as symbols 0 and pairs ij, where i comes from some k-set and j from some $n-k$-set.) (Greene–Iba [1975]; other proofs in Beineke–Pippert [1969], Moon [1969])

8.1.23. Suppose that G is a chordal graph with $\omega(G) = r$. Prove that G has at most $\binom{r}{j} + \binom{r-1}{j-1}(n-r)$ cliques of size j, with equality (for all j simultaneously) if and only if G is an $r-1$-tree.

8.1.24. *The Helly property of the real line.* Suppose that $I_1, \dots, I_k$ are pairwise intersecting real intervals. Prove that $I_1, \dots, I_k$ have a common point.

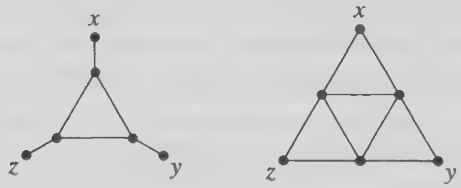
8.1.25. Prove directly that a tree is an interval graph if and only if it is a caterpillar (a tree having a path that contains at least one vertex of each edge).

8.1.26. (!) Let G be an interval graph. Prove that $\overline{G}$ is a comparability graph and that G is a chordal graph. (Hint: Establish a simplicial elimination ordering.)

8.1.27. Prove that a graph G has an interval representation if and only if the clique-vertex incidence matrix of G has the consecutive 1s property.

8.1.28. Prove that G is an interval graph if and only if the vertices of G can be ordered $v_1, \dots, v_n$ such that $v_i \leftrightarrow v_k$ implies $v_j \leftrightarrow v_k$ whenever $i < j < k$. (Jacobson–McMorris–Mulder [1991], for example)

8.1.29. An **asteroidal triple** in a graph is a triple of vertices x, y, z such that between any two there exists a path avoiding the neighborhood of the third. Prove that no asteroidal triple occurs in an interval graph. (Comment: Interval graphs are precisely the chordal graphs that have no asteroidal triples) (Lekkerkerker–Boland [1962]))



8.1.30. Six professors visited the library on the day the rare book was stolen. Each entered once, stayed for some time, and then left. For any two of them that were in the library at the same time, at least one of them saw the other. Detectives questioned the professors and gathered the following testimony:

PROFESSOR	CLAIMED TO HAVE SEEN
Abe	Burt, Eddie
Burt	Abe, Ida
Charlotte	Desmond, Ida
Desmond	Abe, Ida
Eddie	Burt, Charlotte
Ida	Charlotte, Eddie

In this situation, "lying" means providing false information, not omitting information. Assume that the culprit tried to frame another suspect by lying. If one professor lied, who was it? (Golumbic [1980, p20])

8.1.31. (+) Prove that G is a unit interval graph (representable by intervals of the same length) if and only if $A(G) + I$ has the consecutive 1s property. (Roberts [1968])

8.1.32. (+) Prove that G is a proper interval graph (representable by intervals such that none properly contains another) if and only if the clique-vertex incidence matrix of G has the consecutive 1s property for both rows and columns. (Fishburn [1985]

8.1.33. (−) Prove that every P_4-free graph is a Meyniel graph.

8.1.34. (!) Prove that every chordal graph is o-triangulated.

8.1.35. Let C be an odd cycle in a graph with no induced odd cycle. Prove that $V(C)$ has three pairwise-adjacent vertices such that paths joining them in C all have odd length.

8.1.36. (+) Prove that the conditions below are equivalent.

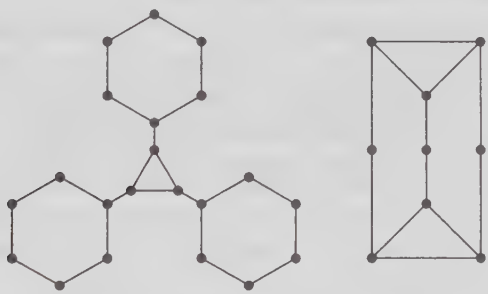
A) Every odd cycle of length at least 5 has a crossing pair of chords.

B) For every pair $x, y \in V(G)$, chordless x, y-paths are all even or all odd.

(Hint: For A $\Rightarrow$ B, consider a pair P_1, P_2 of x, y-paths with opposite parity such that the sum of their lengths is minimal.) (Burlet–Uhry [1984])

8.1.37. Prove that every perfectly orderable graph is strongly perfect. (Hint: Use Lemma 8.1.25) (Chvátal [1984])

8.1.38. (!) Prove that the graphs below are strongly perfect but not perfectly orderable.

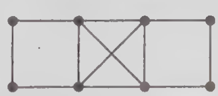


8.1.39. (−) Prove that the graph on the left above is a Meyniel graph but is not perfectly orderable. Prove that the graph $\overline{P}_5$ is perfectly orderable but is not a Meyniel graph.

8.1.40. (!) *Weakly chordal graphs.*

a) Prove that every chordal graph is weakly chordal.

b) Prove that the graph below is weakly chordal but not strongly perfect.



8.1.41. (−) A **skew partition** of G is a partition of $V(G)$ into two nonempty sets X, Y such that $G[X]$ is disconnected and $\overline{G}[Y]$ is disconnected. Chvátal [1985b] conjectured that no minimal imperfect graph has a skew partition. Prove that this implies the Star-Cutset Lemma and is implied by the SPGC.

8.1.42. Prove that the 10-vertex graph in Example 8.1.37 is 3, 3-partitionable. (Chvátal–Graham–Perold–Whitesides [1979])

8.1.43. (−) Let x and v be vertices of a partitionable graph G. Prove that if $x \not\leftrightarrow v$, then every maximum clique containing x consists of one vertex from each stable set that is the mate of a clique containing v. State the complementary assertion when $x \leftrightarrow v$. (Buckingham–Golumbic [1983])

8.1.44. (+) Prove that no p-critical graph has **antitwins**, which are a pair of vertices such that every other vertex is adjacent to exactly one of them. (Hint: Given a p-critical graph with antitwins $\{x, y\}$, let S be the stable set containing y in the unique optimal coloring of $G - x$. Find among the vertices of the $\omega - 1$-colorable subgraph $G - x - S$ an $\omega - 1$ clique in $N(x)$ that doesn't extend into $N(y)$. Similarly, find a stable set in $N(y)$ that doesn't extend into $N(x)$. Now build an induced 5-cycle.) (Note: The partitionable graph of Example 8.1.37 has antitwins.) (Olariu [1988])

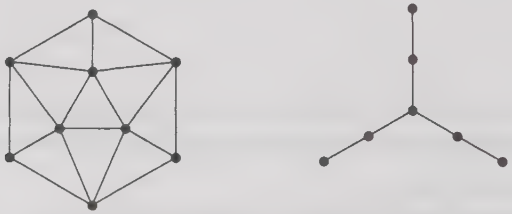
8.1.45. Vertices x, y form an **even pair** if every chordless x, y-path has even length (number of edges). **Twins** (nonadjacent vertices with the same neighborhood) are a special case.

a) Suppose that S_1, S_2 are maximum stable sets in a partitionable graph G. Prove that the subgraph of G induced by the symmetric difference of S_1 and S_2 is connected. (Bland–Huang–Trotter [1979])

b) Use part (a) to prove that no p-critical graph has an even pair. (Comment: Hence no p-critical graph has twins, which proves yet again that vertex duplication preserves perfection.) (Meyniel [1987], Bertschi–Reed [1988])

8.1.46. Let G be a partitionable graph, and let S_1, S_2 be stable sets in the optimal coloring of $G - x$. Use part (a) of the preceding problem to prove that the subgraph of G induced by $S_1 \cup S_2 \cup \{x\}$ is 2-connected. (Buckingham–Golumbic [1983])

8.1.47. Prove that one graph below is a circle graph but not a circular-arc graph, and prove that the other is a circular-arc graph but not a circle graph.



8.1.48. (!) The graph $K_{1,3} + e$ is the 4-vertex graph obtained by adding one edge to $K_{1,3}$. Using the perfection of Meyniel graphs, prove that $K_{1,3} + e$-free graphs satisfy the SPGC. (Meyniel [1976])

8.1.49. Let $G = C_{aw+1}^{w-1}$. Let $S = \{v_{aw}, v_1, v_w, v_{w+2}\} \cup \{v_{iw+1}\colon 2 \le i \le a-1\}$, and let $T = \{v_{(a-1)w+1}, v_{aw}, v_1, v_w\} \cup \{v_{w+i}\colon 2 \le i \le w-1\}$. Prove that S intersects every maximum clique of G and that T intersects every maximum stable set of G. (Chvátal [1976])

8.1.50. (!) *SPGC for circle graphs.* (Buckingham–Golumbic [1983])

a) Use Lemma 8.1.28 to prove that if x is a vertex in a partitionable graph G, then $G - N[x]$ is connected, where $N[x] = N(x) \cup \{x\}$.

b) Use part (a) to prove that partitionable circle graphs are $K_{1,3}$-free.

c) Conclude from part (b) and Corollary 8.1.53 that the SPGC holds for circle graphs.

8.2. Matroids

Many results of graph theory extend or simplify in the theory of matroids. These include the greedy algorithm for minimum spanning trees, the strong duality between maximum matching and minimum vertex cover in bipartite graphs, and the geometric duality relating planar graphs and their duals.

Matroids arise in many contexts but are special enough to have rich combinatorial structure. When a result from graph theory generalizes to matroids, it can then be interpreted in other special cases. Several difficult theorems about graphs have found easier proofs using matroids.

Matroids were introduced by Whitney [1935] to study planarity and algebraic aspects of graphs, by MacLane [1936] to study geometric lattices, and by van der Waerden [1937] to study independence in vector spaces. Most of the language comes from these contexts. Here we emphasize applications to graphs.

HEREDITARY SYSTEMS AND EXAMPLES

In many mathematical contexts, we study sets that avoid conflicts; often this is called "independence". Inherent in this notion is that subsets of independent sets are independent, and the empty set is independent.

8.2.1. Example. *Acyclic sets of edges.* Let E be the edge set of a graph G, and let $X \subseteq E$ be "independent" if it contains no cycle. Every subset of an independent set is independent, and the empty set is independent. The cycles are the minimal dependent sets.

Consider the kite $K_4 - e$, which has five edges. Since spanning trees of this graph have three edges, every set having more than three edges is dependent. Also the two triangles are dependent; this yields eight dependent sets and 24 independent sets among the subsets of E. There are three minimal dependent sets (the cycles) and eight maximal independent sets (the spanning trees). ■

8.2.2. Definition. A **hereditary family** or **ideal** is a collection of sets, $\mathbf{F}$, such that every subset of a set in $\mathbf{F}$ is also in $\mathbf{F}$. A **hereditary system** M on E consists of a nonempty ideal $\mathbf{I}_M$ of subsets of E and the various ways of specifying that ideal, called *aspects* of M.

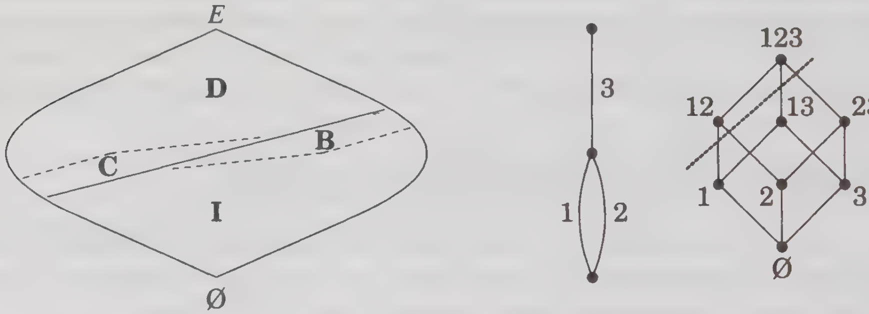
The elements of $\mathbf{I}_M$ are the **independent sets** of M. The other subsets of E (comprising the family $\mathbf{D}_M$) are **dependent**. The **bases** are the maximal independent sets, and the **circuits** are the minimal dependent sets; $\mathbf{B}_M$ and $\mathbf{C}_M$ denote these families of subsets of E.

The **rank** of a subset of E is the maximum size of an independent set in it. The **rank function** r_M is defined by $r(X) = \max\{|Y| : Y \subseteq X,\ Y \in \mathbf{I}\}$.

8.2.3. Example. *Hereditary systems.* Label each vertex $a = (a_1, \dots, a_n)$ of the hypercube Q_n by the corresponding set $X_a = \{i : a_i = 1\}$. Draw Q_n in the plane so that the vertical coordinates of vertices are in order by the size of the sets labeling them.

The diagram below illustrates the relationships among the independent sets, bases, circuits, and dependent sets of a hereditary system. The bases are the maximal elements of the family **I** and the circuits are the minimal elements not in **I**. In every hereditary system, Ø belongs to **I**. If every set is independent, then there is no circuit, but there is always at least one base.

In the example on the right, the independent sets are the acyclic edge sets in a graph with three edges. The only dependent sets are $\{1, 2\}$ and $\{1, 2, 3\}$, the only circuit is $\{1, 2\}$, and the bases are $\{1, 3\}$ and $\{2, 3\}$. The rank of an independent set is its size. For the dependent sets, we have $r(\{1, 2\}) = 1$ and $r(\{1, 2, 3\}) = 2$. ■



8.2.4. Remark. *Aspects of hereditary systems.* A hereditary system M is determined by any of $\mathbf{I}_M$, $\mathbf{B}_M$, $\mathbf{C}_M$, r_M, etc., because each aspect specifies the others. We have expressed $\mathbf{B}_M, \mathbf{C}_M, r_M$ in terms of $\mathbf{I}_M$. Conversely, if we know $\mathbf{B}_M$, then $\mathbf{I}_M$ consists of the sets contained in members of $\mathbf{B}_M$. If we know $\mathbf{C}_M$, then $\mathbf{I}_M$ consists of the sets containing no member of $\mathbf{C}_M$. If we know r_M, then $\mathbf{I}_M = \{X \subseteq E\colon r_M(X) = |X|\}$. ■

Hereditary systems are too general to behave nicely. We restrict our attention to hereditary systems having an additional property, and these we call matroids. We can translate any restriction on $\mathbf{I}_M$ into a corresponding restriction on some other aspect of the hereditary system. Because hereditary systems can be specified in many ways, we have many equivalent definitions of matroids. Using various motivating examples, we state several of these properties that characterize matroids. Later we prove that they are equivalent. We begin with the fundamental example from graphs.

8.2.5. Definition. The **cycle matroid** $M(G)$ of a graph G is the hereditary system on $E(G)$ whose circuits are the cycles of G. A hereditary system that is $M(G)$ for some graph G is a **graphic matroid**.

8.2.6. Example. *Bases in cycle matroids.* The bases of the cycle matroid $M(G)$ are the edge sets of the maximal forests in G. Each maximal forest contains a spanning tree from each component, so they have the same size. Consider $B_1, B_2 \in \mathbf{B}$ with $e \in B_1 - B_2$. Deleting e from B_1 disconnects some component of B_1; since B_2 contains a tree spanning that component of G, some edge $f \in B_2 - B_1$ can be added to $B_1 - e$ to reconnect it.

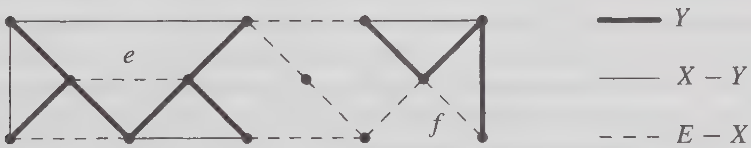
For a hereditary system M, the **base exchange property** is: if $B_1, B_2 \in \mathbf{B}_M$, then for all $e \in B_1 - B_2$ there exists $f \in B_2 - B_1$ such that $B_1 - e + f \in \mathbf{B}_M$. Matroids are the hereditary systems satisfying the base exchange property. ■

8.2.7. Remark. In this subject, we often discuss inclusion and omission of single elements from sets. For symmetry and simplicity, we use the symbols $+$ and $-$ instead of $\cup$ and $-$ for this, and we drop the set brackets on 1-element sets. ■

8.2.8. Example. *Rank function in cycle matroids.* Let G be a graph with n vertices. For $X \subseteq E(G)$, let G_X denote the spanning subgraph of G with edge set X. In $M(G)$, an independent subset of X is the edge set of a forest in G_X. When G_X has k components, the maximum size of such a forest is $n - k$. Hence $r(X) = n - k$. Below we show such a forest Y (bold) within X (bold and solid).

If $r(X + e) = r(X)$ for some $e \in E - X$, then the endpoints of e lie in a single component of G_X; adding e does not combine components. If we add two such edges, then again we do not combine components. Therefore, $r(X) = r(X + e) = r(X + f)$ implies $r(X) = r(X + e + f)$.

For a hereditary system M on E, the **(weak) absorption property** is: if $X \subseteq E$ and $e, f \in E$, then $r(X) = r(X+e) = r(X+f)$ implies $r(X+e+f) = r(X)$. Matroids are the hereditary systems satisfying the absorption property (name suggested by A. Kézdy). ■



Graphs may have loops and multiple edges. In cycle matroids, they lead to circuits of sizes 1 and 2. We use these terms for hereditary systems in general.

8.2.9. Definition. In a hereditary system, a **loop** is an element forming a circuit of size 1. **Parallel elements** are distinct non-loops forming a circuit of size 2. A hereditary system is **simple** if it has no loops or parallel elements.

8.2.10. Definition. The **vectorial matroid** on a set E of vectors in a vector space is the hereditary system whose independent sets are the linearly independent subsets of vectors in E. A matroid expressible in this way is a **linear matroid** (or **representable matroid**). The **column matroid** $M(A)$ of a matrix A is the vectorial matroid defined on its columns.

8.2.11. Example. *Circuits in vectorial matroids.* The set E may have repeated vectors; these would be parallel elements. The circuits are the minimal sets $\{x_1, \dots, x_k\} \subseteq E$ such that $\sum c_i x_i = 0$ using coefficients not all zero. Minimality forces all $c_i \neq 0$.

Let C_1, C_2 be distinct circuits containing x. Using the equations of dependence for C_1 and C_2, we can write x as a linear combination in terms of $C_1 - x$ and in terms of $C_2 - x$. Equating these expressions yields an equation of dependence for $C_1 \cup C_2 - x$; thus $C_1 \cup C_2 - x$ contains a circuit.

For a hereditary system M on E, the **(weak) elimination property** is: whenever C_1, C_2 are distinct circuits and $x \in C_1 \cap C_2$, another member of $\mathbf{C}_M$ is contained in $C_1 \cup C_2 - x$. Matroids are the hereditary systems satisfying the weak elimination property.

The column matroid of the matrix below is also the cycle matroid $M(K_4 - e)$.

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$ ■

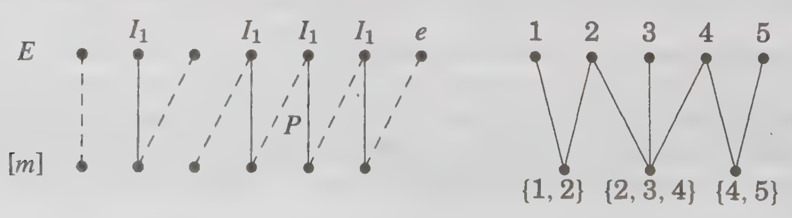
8.2.12. Definition. The **transversal matroid** induced by sets $A_1, \dots, A_m$ with union E is the hereditary system on E whose independent sets are the systems of distinct representatives of subsets of $\{A_1, \dots, A_m\}$. Equivalently, letting G be the $E, [m]$-bigraph defined by $e \leftrightarrow i$ if and only if $e \in A_i$, the independent sets are the subsets of E that are saturated by matchings in G.

8.2.13. Example. *Independent sets in transversal matroids.* When M, M' are matchings in G and $|M'| > |M|$, the symmetric difference $M \triangle M'$ contains an M-augmenting path P (Theorem 3.1.10). Replacing $M \cap P$ with $M' \cap P$ yields a matching of size $|M| + 1$ that saturates all vertices of M plus the endpoints of P.

Consider independent sets I_1, I_2 in the transversal matroid generated by $A_1, \dots, A_m$. In the associated bipartite graph, let M_1, M_2 be matchings saturating I_1, I_2, respectively (on the left below, M_1 is solid and M_2 is dashed). If $|I_2| > |I_1|$, then the matching obtained from M_1 by using an M_1-augmenting path in $M_2 \triangle M_1$ saturates I_1 plus an element $e \in I_2 - I_1$; this "augments" I_2.

For a hereditary system on E, the **augmentation property** is: for distinct $I_1, I_2 \in \mathbf{I}$ with $|I_2| > |I_1|$, there exists $e \in I_2 - I_1$ such that $I_1 \cup \{e\} \in \mathbf{I}$. Matroids are the hereditary systems satisfying the augmentation property.

The transversal matroid of the family $\mathbf{A} = \{\{1, 2\}, \{2, 3, 4\}, \{4, 5\}\}$, illustrated by the bipartite graph on the right, is again $M(K_4 - e)$. ■

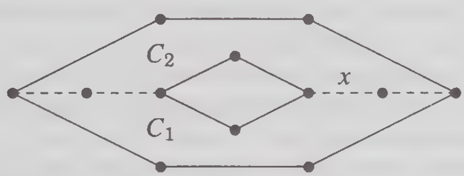


The name "transversal matroid" arises from the use of "transversal" in systems of distinct representatives. An SDR for a subset of $\{A_1, \dots, A_m\}$ is a **partial transversal** for the full system. The independent sets of the transversal matroid on $\bigcup A_i$ are the partial transversals of $\{A_1, \dots, A_m\}$. That these are matroids was discovered by Edmonds and Fulkerson [1965] and independently by Mirsky and Perfect [1967], who extended the result to infinite sets.

Every matroid must satisfy all properties of matroids. Once we show that the properties defined above are equivalent for hereditary systems, we need only verify one to use all. First we check that they all hold for cycle matroids.

8.2.14. Example. *Augmentation in cycle matroids.* Consider $I_1, I_2 \in \mathbf{I}_{M(G)}$. As in Example 8.2.8, the spanning subgraph G_{I_1} has $k = n - |I_1|$ components, and its largest forest has $n - k = |I_1|$ edges. Therefore, the forest I_2 has some edge with endpoints in two components of G_{I_1}. This edge can be added to I_1 to obtain a larger independent set. Hence the augmentation property holds. ■

8.2.15. Example. *Weak elimination in cycle matroids.* The circuits of $M(G)$ are the edge sets of cycles of G. Cycles have even degree at each vertex. If $C_1, C_2 \in \mathbf{C}$, then the symmetric difference $C_1 \triangle C_2$ also has even degree at each vertex. If $C_1 \neq C_2$, this implies that $C_1 \triangle C_2$ contains a cycle (see Proposition 1.2.27). This is stronger than the weak elimination property, since $C_1 \triangle C_2 \subseteq C_1 \cup C_2 - x$. In the figure below, C_1 and C_2 are face boundaries of length 9 sharing the dashed edges, and $C_1 \triangle C_2$ is the union of two disjoint cycles. ■



For transversal matroids, the base exchange property is similar to the augmentation property; Exercise 9 considers the weak elimination property. For linear matroids, directly verifying the augmentation or base exchange property requires the algebraic result that k linearly independent vectors cannot all be expressed as linear combinations of a smaller set. Instead, we can use Theorem 8.2.20. Since the weak elimination property holds for independent sets of vectors, many theorems of linear algebra follow from Theorem 8.2.20!

8.2.16. Remark. *Notational conventions*: Boldface $\mathbf{I}, \mathbf{B}, \mathbf{C}$ for families of subsets of E allows $I \in \mathbf{I}$, $B \in \mathbf{B}$, $C \in \mathbf{C}$ to denote members of the families. Roman letters I, B, C, R denote properties that yield matroids. We use e, f, x, y as elements of E, and we use X, Y, F as subsets of E. ■

Every hereditary family is the collection of independent sets of a hereditary system. A collection $\mathbf{B}$ is realizable as the set of bases of a hereditary system if and only if $\mathbf{B}$ is nonempty and no element of $\mathbf{B}$ contains another. A collection

$\mathbf{C}$ is realizable as the set of circuits of a hereditary system if and only if the elements of $\mathbf{C}$ are nonempty and no element of $\mathbf{C}$ contains another.

The characterization of rank functions is more subtle. It includes two properties (r1, r2 below) that we will need, plus an additional technical condition that forces r to be the rank function of the hereditary system M defined by $I_M = \{X \subseteq E \colon r(X) = |X|\}$.

8.2.17. Lemma. For the rank function r of a hereditary system on E,

(r1) $r(\emptyset) = 0$.

(r2) $r(X) \le r(X+e) \le r(X) + 1$ whenever $X \subseteq E$ and $e \in E$.

Proof: From the definition $r(X) = \max\{|Y| : Y \subseteq X, Y \in \mathbf{I}\}$, we have $r(\emptyset) = 0$. Because $X + e$ contains every independent subset of X, also $r(X+e) \ge r(X)$. Because the independent subsets of $X + e$ not contained in X consist of e plus an independent subset of X, we have $r(X+e) \le r(X) + 1$. ■

PROPERTIES OF MATROIDS

We have remarked that many equivalent conditions on hereditary systems yield matroids. We can show that a hereditary system is a matroid by verifying any of them, after which we can employ them all without additional proof. We obtained the same benefit from equivalent characterizations of trees.

Adding an edge to a forest creates at most one cycle. More generally, adding one element to an independent set in a matroid creates at most one circuit. Our proof of the greedy algorithm for spanning trees (Theorem 2.3.3) used *only* this property of graphs. This "induced circuit" property is one of the conditions that characterize matroids, as is the effectiveness of the greedy algorithm itself! Both properties appear in our list.

Given weights on the elements of a matroid, the **greedy algorithm** is the process of iteratively including an element of largest nonnegative weight whose addition to the independent set already selected yields a larger independent set. Rado [1957] proved that matroids are precisely the hereditary systems for which the greedy algorithm selects a maximum-weighted independent set regardless of the choice of weights.

8.2.18. Definition. A hereditary system M on E is a **matroid** if it satisfies any of the following additional properties, where $\mathbf{I}, \mathbf{B}, \mathbf{C}$, and r are the independent sets, bases, circuits, and rank function of M.

I: **augmentation**—if $I_1, I_2 \in \mathbf{I}$ with $|I_2| > |I_1|$, then $I_1 + e \in \mathbf{I}$ for some $e \in I_2 - I_1$.

U: **uniformity**—for every $X \subseteq E$, the maximal subsets of X belonging to $\mathbf{I}$ have the same size.

B: **base exchange**—if $B_1, B_2 \in \mathbf{B}$, then for all $e \in B_1 - B_2$ there exists $f \in B_2 - B_1$ such that $B_1 - e + f \in \mathbf{B}$.

R: **submodularity**—$r(X \cap Y) + r(X \cup Y) \le r(X) + r(Y)$ whenever $X, Y \subseteq E$.

A: **weak absorption**—$r(X) = r(X+e) = r(X+f)$ implies $r(X+e+f) = r(X)$ whenever $X \subseteq E$ and $e, f \in E$,

A′: **strong absorption**—if $X, Y \subseteq E$, and $r(X+e) = r(X)$ for all $e \in Y$, then $r(X \cup Y) = r(X)$.
C: **weak elimination**—for distinct circuits $C_1, C_2 \in \mathbf{C}$ and $x \in C_1 \cap C_2$, there is another member of $\mathbf{C}$ contained in $(C_1 \cup C_2) - x$.
J: **induced circuits**—if $I \in \mathbf{I}$, then $I + e$ contains at most one circuit.
G: **greedy algorithm**—for each nonnegative weight function on E, the greedy algorithm selects an independent set of maximum total weight.

The base exchange property implies that all bases have the same size: if $|B_1| < |B_2|$ for some $B_1, B_2 \in \mathbf{B}$, then we can iteratively replace elements of $B_1 - B_2$ by elements of $B_2 - B_1$ to obtain a base of size $|B_1|$ contained in B_2, but no base is contained in another.

8.2.19.* Remark. The rank of a set $X \subseteq E$ in a vectorial matroid is the dimension of the space spanned by X. Hence for vectorial matroids the submodularity inequality says that $\dim U \cap V + \dim U \oplus V \le \dim U + \dim V$, where $U, V, U \oplus V$ are the spaces spanned by subsets $X, Y, X \cup Y$ of E, respectively. The usual proof of this is the vector space statement of our proof of U$\Rightarrow$R below. Exercise 10 obtains submodularity directly for cycle matroids.

Various of these properties (together with requirements for a hereditary system) have been used as the defining condition for matroids. Examples include I (Welsh [1976], Schrijver [to appear]), U (Edmonds [1965b,c], Bixby [1981], Nemhauser–Wolsey [1988]), A (Whitney [1935]), C (Tutte [1970]), G (Papadimitriou–Steiglitz [1982]), and others (van der Waerden [1937], Rota [1964], Crapo–Rota [1970], Aigner [1979]). ■

Many authors include basic properties of hereditary systems in the set of axioms characterizing some aspect of a matroid. This can distract from the special additional properties of matroids and lead to extra work. Starting with hereditary systems yields more concise proofs. All properties of hereditary systems are always available.

8.2.20. Theorem. For a hereditary system M, the conditions defining matroids in Definition 8.2.18 are equivalent.

Proof: U $\Rightarrow$ B. By uniformity for $X = E$, all bases have the same size. We then apply uniformity to the set $(B_1 - e) \cup B_2$. This yields an augmentation of the independent set $B_1 - e$ from B_2 to reach size $|B_2|$.

B $\Rightarrow$ I. Given independent sets $I_1, I_2 \in \mathbf{I}$ with $|I_2| > |I_1|$, choose $B_1, B_2 \in \mathbf{B}$ such that $I_1 \subseteq B_1$, $I_2 \subseteq B_2$. We use base exchange to replace elements of $B_1 - I_1$ outside B_2 with elements of B_2. Hence we may assume that $B_1 - I_1 \subseteq B_2$. If $B_1 - I_1 \subseteq B_2 - I_2$, then $|B_1| < |B_2|$, which is forbidden by the base exchange property as remarked above. Hence I_2 has an element in $B_1 - I_1$, and we use such an element to augment I_1.

I $\Rightarrow$ A. Suppose that $r(X) = r(X+e) = r(X+f)$. If $r(X+e+f) > r(X)$, then let I_1, I_2 be maximum independent subsets of X and of $X+e+f$. Now $|I_2| > |I_1|$, and we can augment I_1 from I_2. Since I_1 is a maximum independent subset of

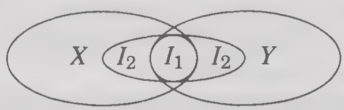
X, the augmentation can only add e or f, which contradicts the hypothesis that $r(X) = r(X+e) = r(X+f)$.

A $\Rightarrow$ A$'$. We use induction on $|Y - X|$. The statement is trivial when $|Y - X| = 1$. When $|Y - X| > 1$, choose $e, f \in Y - X$, and let $Y' = Y - e - f$. Applying the induction hypothesis to proper subsets of Y yields $r(X) = r(X \cup Y') = r(X \cup Y' + e) = r(X \cup Y' + f)$. Now weak absorption yields $r(X) = r(X \cup Y)$.

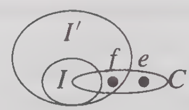
A$'$ $\Rightarrow$ U. If Y is a maximal independent subset of X, then $r(Y+e) = r(Y)$ for all $e \in X - Y$. By strong absorption, $r(X) = r(Y) = |Y|$. Hence all such Y have the same size.

U $\Rightarrow$ R. Given $X, Y \subseteq E$, choose a maximum independent set I_1 from $X \cap Y$. By uniformity, I_1 can be enlarged to a maximum independent subset of $X \cup Y$; call this I_2. Consider $I_2 \cap X$ and $I_2 \cap Y$; these are independent subsets of X and Y, and each includes I_1. Hence

$$r(X \cap Y) + r(X \cup Y) = |I_1| + |I_2| = |I_2 \cap X| + |I_2 \cap Y| \le r(X) + r(Y).$$



U$\Rightarrow$R



J$\Rightarrow$G

R $\Rightarrow$ C. Consider distinct circuits $C_1, C_2 \in \mathbf{C}$ with $x \in C_1 \cap C_2$. We have $r(C_1) = |C_1| - 1$ and $r(C_2) = |C_2| - 1$. Also $r(C_1 \cap C_2) = |C_1 \cap C_2|$, since every proper subset of a circuit is independent. If $(C_1 \cup C_2) - x$ does not contain a circuit, then $r((C_1 \cup C_2) - x) = |C_1 \cup C_2| - 1$, and hence $r(C_1 \cup C_2) \ge |C_1 \cup C_2| - 1$. Applying submodularity to C_1 and C_2 yields the contradiction

$$|C_1 \cap C_2| + |C_1 \cup C_2| - 1 \le |C_1| + |C_2| - 2.$$

C $\Rightarrow$ J. If $I + e$ contains $C_1, C_2 \in \mathbf{C}$ for some $I \in \mathbf{I}$, then C_1, C_2 both contain e. Now weak elimination guarantees a circuit in $(C_1 \cup C_2) - e$. On the other hand, $(C_1 \cup C_2) - e$ is independent, being contained in I.

J $\Rightarrow$ G. For weight function w, let I be the output of the greedy algorithm. Among the maximum-weight independent sets, let I^* be one having largest intersection with I. The algorithm cannot end with $I \subset I^*$. If $I \ne I^*$, then let e be the first element of $I - I^*$ chosen by the algorithm. By the choice of I^*, $I^* + e$ is dependent; hence it has a unique circuit C. Since $C \not\subseteq I$, we may choose $f \in C - I$. Since $I^* + e$ has no other circuit, $I^* + e - f \in \mathbf{I}$. The optimality of I^* yields $w(f) \ge w(e)$. Since f and the elements of I chosen earlier than e all lie in I^*, f does not complete a circuit with them. Thus f was available when the algorithm selected e, which yields $w(f) \le w(e)$. Now $w(f) = w(e)$ and $w(I^* + e - f) = w(I^*)$. With $|I^* + e - f \cap I| > |I^* \cap I|$, this contradicts the choice of I^*. Thus $I^* = I$.

G $\Rightarrow$ I. Given $I_1, I_2 \in \mathbf{I}$ with $k = |I_1| < |I_2|$, we design a weight function for which the success of the greedy algorithm yields the desired augmentation. Let $w(e) = k + 2$ for $e \in I_1$, and let $w(e) = k + 1$ for $e \in I_2 - I_1$. Let $w(e) =$

0 for $e \notin I_1 \cup I_2$. Now $w(I_2) \geq (k+1)^2 > k(k+2) = w(I_1)$, so I_1 is not a maximum-weighted independent set. However, the greedy algorithm chooses every element of I_1 before any element of $I_2 - I_1$. Because it finds a maximum-weighted independent set, it continues after absorbing I_1 and adds an element $e \in I_2 - I_1$ such that $I_1 + e \in \mathbf{I}$. ■

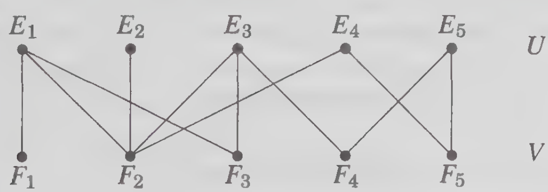
The property most often used to show that a hereditary system is a matroid is the augmentation property.

8.2.21. Example. The **uniform matroid** of rank k, denoted $U_{k,n}$ when $|E| = n$, is defined by $\mathbf{I} = \{X \subseteq E\colon |X| \leq k\}$. This immediately satisfies the base exchange and augmentation properties. The **free matroid** is the uniform matroid of rank $|E|$. Uniform matroids are used in building more interesting matroids and in characterizing classes of matroids. Few uniform matroids are graphic, and few graphic matroids are uniform (Exercise 6). Neither $M(K_4 - e)$ nor $M(K_4)$ is a uniform matroid.

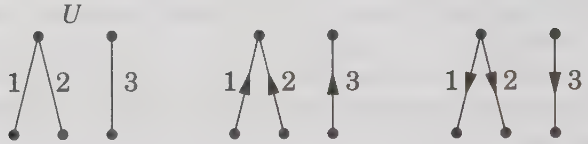
A linear matroid representable over the field $\mathbb{Z}_2$ or $\mathbb{Z}_3$ is **binary** or **ternary**, respectively. Every graphic matroid is binary (Exercise 43); $U_{2,4}$ is ternary (Exercise 44) but not binary (and hence not graphic). ■

8.2.22. Example. The **partition matroid** on E induced by a partition of E into blocks $E_1, \ldots, E_k$ is defined by $\mathbf{I} = \{X \subseteq E\colon |X \cap E_i| \leq 1 \text{ for all } i\}$. Since $\emptyset \in \mathbf{I}$, and since $X \in \mathbf{I}$ when its elements lie in distinct blocks, $\mathbf{I}$ is a hereditary family. Given $I_1, I_2 \in \mathbf{I}$ with $|I_2| > |I_1|$, the set I_2 must intersect more blocks than I_1; an element of I_2 in a block that I_1 misses yields the desired augmentation of I_1. Alternatively, $r(X)$ is the number of blocks having elements in X; this satisfies the absorption property. (Note: $M(K_4 - e)$ is *not* a partition matroid.)

Given a U, V-bigraph G, the incidences with $U = u_1, \ldots, u_k$ define a partition matroid on $E(G)$ (this differs from the transversal matroid on U induced by G). The blocks are the sets $E_i = \{e \in E(G)\colon u_i \in e\}$. A set $X \subseteq E(G)$ is a matching in G if and only if X is independent in the partition matroid induced by U *and* in the partition matroid induced by V. This is the motivation for our later discussion of matroid intersection.



When G has an odd cycle, G has no set of vertices whose incident sets partition $E(G)$. In a digraph, however, each edge has a head and a tail, and we can define the **head partition matroid** and the **tail partition matroid** using the edge partitions induced by incidences with heads and by incidences with tails. (Example: The matroid of Example 8.2.3 arises as the partition matroid on E induced by U in the bipartite graph below, as the head partition matroid in the first digraph, and as the tail partition matroid in the second digraph.) ■



THE SPAN FUNCTION

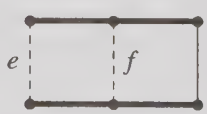
We next introduce several additional aspects of hereditary systems and matroid properties involving them. We use these aspects to illuminate matroid duality, which will lead to a characterization of planar graphs using matroids.

The algebraic concept of the space "spanned" by a set of vectors extends to hereditary systems. The definition is suggested by cycle matroids; a set spans itself and the elements that complete circuits with its subsets.

8.2.23. Definition. The **span function** of a hereditary system M is the function σ_M on the subsets of E defined by $\sigma_M(X) = X \cup \{e \in E\colon Y + e \in \mathbf{C}_M$ for some $Y \subseteq X\}$. If $e \in \sigma(X)$, then X **spans** e.

In a hereditary system, X is a dependent set if and only if it contains a circuit, which by Definition 8.2.23 holds if and only if $e \in \sigma(X - e)$ for some $e \in X$. We can therefore find the independent sets from the span function via $\mathbf{I} = \{X \subseteq E\colon (e \in X) \Rightarrow (e \notin \sigma(X - e))\}$. The properties of span functions that we use in studying matroids are (s1, s2, s3) below (an additional technical condition is needed to characterize the span functions of hereditary systems). First we illustrate property (s3) using graphs.

8.2.24. Example. In the cycle matroid $M(G)$, the meaning of $e \notin \sigma(X)$ is that X has no path between the endpoints of e. If also $e \in \sigma(X + f)$, then adding f completes such a path. The path completes a cycle with e, and hence also $f \in \sigma(X + e)$. In the figure below, X consists of the four bold edges. ■



8.2.25. Proposition. If σ is the span function of a hereditary system on E, and $X, Y \subseteq E$, then the following properties hold.

s1) $X \subseteq \sigma(X)$ (σ is **expansive**).

s2) $Y \subseteq X$ implies $\sigma(Y) \subseteq \sigma(X)$ (σ is **order-preserving**).

s3) $e \notin \sigma(X)$ and $e \in \sigma(X + f)$ imply $f \in \sigma(X + e)$ (**Steinitz exchange**).

Proof: Definition 8.2.23 implies immediately that σ is expansive and order-preserving. If $e \in \sigma(X + f)$, then e belongs to a circuit C in $X + f + e$. If also $e \notin \sigma(X)$, then $f \in C$. This circuit yields $f \in \sigma(X + e)$, and hence σ satisfies the Steinitz exchange property. ■

Properties of the span function lead to a short proof of a stronger form of the elimination property. The weak elimination property states that when $e \in C_1 \cap C_2$, there is a circuit in $(C_1 \cup C_2) - e$. Cycle matroids have the much stronger property that $C_1 \triangle C_2$ is an edge-disjoint union of cycles, since every vertex in $C_1 \triangle C_2$ has even degree. General matroids have the intermediate property that all elements of the symmetric difference belong to cycles in $(C_1 \cup C_2) - e$ when $e \in C_1 \cap C_2$ (Property C′ below).

We need a property relating rank and span in hereditary systems. The truth of the converse is our next characterization of matroids.

8.2.26.* Lemma. In a hereditary system, $[r(X + e) = r(X)] \Rightarrow e \in \sigma(X)$.

Proof: Let Y be a maximum independent subset of X. Since $|Y| = r(X) = r(X + e)$, also Y is a maximum independent subset of $X + e$. Hence e completes a circuit with some subset of X contained in Y, and $e \in \sigma(X)$. ■

8.2.27.* Theorem. If M is a hereditary system, then each condition below is necessary and sufficient for M to be a matroid.

P: **incorporation**—$r(\sigma(X)) = r(X)$ for all $X \subseteq E$.
S: **idempotence**—$\sigma^2(X) = \sigma(X)$ for all $X \subseteq E$.
T: **transitivity of dependence**—if $e \in \sigma(X)$ and $X \subseteq \sigma(Y)$, then $e \in \sigma(Y)$.
C′: **strong elimination**—whenever $C_1, C_2 \in \mathbf{C}$, $e \in C_1 \cap C_2$, and $f \in C_1 \triangle C_2$, there exists $C \in \mathbf{C}$ such that $f \in C \subseteq C_1 \cup C_2 - e$.

Proof: U $\Rightarrow$ P. Every element in $\sigma(X) - X$ completes a circuit with a subset of of X and thus lies in the span of every set between X and $\sigma(X)$. Thus it suffices to prove that $r(Y + e) = r(Y)$ when $e \in \sigma(Y)$. Let Z be a subset of Y such that $Z + e \in \mathbf{C}$. Augment Z to a maximal independent subset I of $Y + e$. By the uniformity property, $|I| = r(Y + e)$. Since $Z + e \in \mathbf{C}$, we have $e \notin I$. Thus $I \subseteq Y$, and we have $r(Y) \geq |I| = r(Y + e)$. (Absorption can be used instead.)

P $\Rightarrow$ S. Since σ is expansive, $\sigma^2(X) \supseteq \sigma(X)$, and we need only show that $e \in \sigma^2(X)$ implies $e \in \sigma(X)$. By the incorporation property, $r(\sigma(X) + e) = r(\sigma(X))$ and $r(\sigma(X)) = r(X)$. Since $X \subseteq \sigma(X)$, monotonicity of r yields $r(X) \leq r(X+e) \leq r(\sigma(X) + e) = r(X)$. Since equality holds throughout, Lemma 8.2.26 yields $e \in \sigma(X)$.

S $\Rightarrow$ T. If $X \subseteq \sigma(Y)$, then the order-preserving and idempotence properties of σ imply $\sigma(X) \subseteq \sigma^2(Y) = \sigma(Y)$.

T $\Rightarrow$ C′. Given distinct $C_1, C_2 \in \mathbf{C}$ with $e \in C_1 \cap C_2$ and $f \in C_1 - C_2$, we want $f \in \sigma(Y)$, where $Y =)C_1 \cup C_2) - e - f$. We have $f \in \sigma(X)$, where $X = C_1 - f$. By T, it suffices to show $X \subseteq \sigma(Y)$. Since $X - e \subseteq Y \subseteq \sigma(Y)$, we need only show $e \in \sigma(Y)$. Since σ is order-preserving, we have $e \in \sigma(C_2 - e) \subseteq \sigma(Y)$.

C′ $\Rightarrow$ C. C is a less restrictive statement than C′. ■

Like uniqueness of induced circuits (J), the incorporation property (P) relates two aspects of hereditary systems. These are well-known properties of matroids, and in the approach via hereditary systems they become characterizations. The equivalence of C and C′ was first proved by Lehman [1964].

Idempotence occurs naturally for graphic and linear matroids. The span of a set of vectors contains nothing additional in its span; similarly, every edge that can be added to the span of a set of edges joins two components. This suggests related aspects of hereditary systems.

8.2.28. Definition. The **spanning sets** of a hereditary system on E are the sets $X \subseteq E$ such that $\sigma(X) = E$. The **closed sets** are the sets $X \subseteq E$ such that $\sigma(X) = X$ (also called **flats** or **subspaces**). The **hyperplanes** are the maximal proper closed subsets of E.

8.2.29.* Remark. The span function of a matroid is also called its **closure function**. A **closure operator** is an expansive, order-preserving, idempotent function from the family of subsets of a set to itself. A closure operator is the span function of a matroid if and only if it has the Steinitz exchange property.

In every hereditary system, the span function satisfies Steinitz exchange. Thus treating matroids as hereditary systems with additional properties is not well suited for studying closure operators. The span function of a hereditary system M is a closure operator if and only if M is a matroid. Matroids are developed from lattice theory in MacLane [1936], Rota [1964], and Aigner [1979].

We have not considered all relationships among aspects of matroids. Brylawski [1986] presents a matrix describing the transformations among about a dozen aspects of matroids, calling these maps **cryptomorphisms**. ■

THE DUAL OF A MATROID

Duality in matroids generalizes the notion of duality for planar graphs. Every connected plane graph G has a natural dual graph G^* such that $(G^*)^* = G$. The dual is formed by associating a vertex of G^* with each face of G and including a dual edge e^* in G^* for each edge of G, such that the endpoints of the edge e^* are the vertices for the faces on the two sides of e.

A set of edges in a plane graph G forms a spanning tree in G if and only if the duals to the remaining edges form a spanning tree in G^* (Exercise 6.1.21). Hence the bases in the cycle matroid $M(G^*)$ are the complements of the bases in $M(G)$. We define duality for matroids and hereditary systems so that the properties of duality in planar graphs generalize.

8.2.30. Definition. The **dual** of a hereditary system M on E is the hereditary system M^* whose bases are the complements of the bases of M. The aspects $\mathbf{B}^*(\mathbf{B}_{M^*})$, $\mathbf{C}^*$, $\mathbf{I}^*$, r^*, σ^*, of M^* are the **cobases**, **cocircuits**, etc., of M.

The **supbases** $\mathbf{S}$ of M are the sets containing a base. The **hypobases** $\mathbf{H}$ are the maximal subsets containing no base. We write $\overline{X}$ for $E - X$.

8.2.31. Lemma. If M is a hereditary system, then

a) $\mathbf{B}^* = \{\overline{B}: B \in \mathbf{B}\}$ and $(M^*)^* = M$.

b) $\mathbf{I}^* = \{\overline{S}: S \in \mathbf{S}\}$ and $\mathbf{S}^* = \{\overline{I}: I \in \mathbf{I}\}$.

c) $\mathbf{C}^* = \{\overline{H}: H \in \mathbf{H}\}$ and $\mathbf{H}^* = \{\overline{C}: C \in \mathbf{C}\}$.

Proof: The statement about $\mathbf{B}^*$ is the definition of M^*. It immediately yields $(M^*)^* = M$ and both parts of (b). Also, X is a maximal (proper) subset of E containing no base (a hypobase of M) if and only if $\overline{X}$ is a minimal nonempty set contained in no cobase, which is a circuit of M^*. Similarly, the hypobases of M^* are the complements of the circuits of M. ■

We have chosen "supbase" and "hypobase" to share initials with "spanning" and "hyperplane", because for matroids the spanning sets and supbases are the same, and the hyperplanes and hypobases are the same.

8.2.32. Lemma. If M is a matroid, then the supbases are the spanning sets, and the hypobases are the hyperplanes.

Proof: A set X is spanning if and only if $\sigma(X) = E$. By the incorporation property, this is equivalent to $r(X) = r(E)$. By the uniformity property, this is equivalent to X containing a base. For hyperplanes, see Exercise 32.) ■

Consider $B_1, B_2 \subseteq E$. If neither of B_1, B_2 contains the other, then also neither of $\overline{B}_1, \overline{B}_2$ contains the other. Therefore, the dual of a hereditary system is a hereditary system. The notion of duality becomes useful when we prove that the dual of a matroid is a matroid. This follows easily from a dual version of the base exchange property.

8.2.33. Lemma. If M is a matroid and $B_1, B_2 \in \mathbf{B}$, then for each $e \in B_1 - B_2$ there exists $f \in B_2$ such that $B_2 + e - f$ is a base.

Proof: Since B_2 is a base, $B_2 + e$ contains exactly one circuit C. Since B_1 is independent, C also contains an element $f \in B_2 - B_1$. Now $B_2 + e - f$ contains no circuit and has size $r(E)$. ■

8.2.34. Theorem. (Whitney [1935]) The dual of a matroid M on E is a matroid with rank function $r^*(X) = |X| - (r(E) - r(\overline{X}))$.

Proof: We have observed that M^* is a hereditary system; now we prove the base exchange property for M^*. If $\overline{B}_1, \overline{B}_2 \in \mathbf{B}^*$ and $e \in \overline{B}_1 - \overline{B}_2$, then $B_1, B_2 \in \mathbf{B}$, with $e \in B_2 - B_1$. By Lemma 8.2.33, there exists $f \in B_1 - B_2$ such that $B_1 + e - f \in \mathbf{B}$. Now $\overline{B}_1 - e + f \in \mathbf{B}^*$ is the desired exchange.

To compute $r^*(X)$, let Y be a maximal coindependent subset of X, so $r^*(X) = r^*(Y) = |Y|$. By Lemma 8.2.31, $\overline{Y}$ is a minimal superset of $\overline{X}$ that contains a base of M. Since $\overline{Y}$ arises from $\overline{X}$ by augmenting a maximal independent subset of $\overline{X}$ to become a base, we have $|\overline{Y}| - |\overline{X}| = r(E) - r(\overline{X})$. With $|\overline{Y}| - |\overline{X}| = |X| - |Y|$, this yields the desired formula

$$r^*(X) = |Y| = |X| - (|\overline{Y}| - |\overline{X}|) = |X| - (r(E) - r(\overline{X})).$$ ■

We can restate any matroid property using dual aspects. Exercises 33–34 request characterizations of hyperplanes and closed sets by this method. More subtle results involve relationships between a matroid and its dual.

8.2.35. Proposition. (Dual augmentation property) Let M be a matroid. If $X \in \mathbf{I}$ and $X' \in \mathbf{I}^*$ are disjoint, then there are disjoint $B \in \mathbf{B}$ and $B' \in \mathbf{B}^*$ such that $X \subseteq B$ and $X' \subseteq B'$.

Proof: Since X' is coindependent in M, $\overline{X'}$ is spanning in M. Hence every maximal independent subset of $\overline{X'}$ is a base; we augment $X \subseteq \overline{X'}$ to a base B contained in $\overline{X'}$. The cobase $B' = \overline{B}$ contains X'. ■

We will use cycle matroids to characterize planar graphs. The next result enables us to describe the cocircuits of a cycle matroid.

8.2.36. Proposition. Cocircuits of a matroid are the minimal sets intersecting every base. Bases are the minimal sets intersecting every cocircuit.

Proof: The cocircuits are the minimal sets contained in no cobase. Because the cobases are the complements of the bases, a set is contained in no cobase if and only if it intersects every base. Similarly, the cobases are the maximal sets containing no cocircuit, so the complements of the cobases are the minimal sets intersecting every cocircuit. ■

8.2.37. Corollary. The cocircuits of the cycle matroid $M(G)$ are the bonds of G.

Proof: By Proposition 8.2.36, the cocircuits are the minimal sets intersecting every maximal forest. Hence they are the minimal sets whose deletion increases the number of components; these are the bonds. ■

8.2.38. Definition. The **bond matroid** or **cocycle matroid** of a graph G is the hereditary system whose circuits are the bonds of G.

By Corollary 8.2.37, the bond matroid of G is the the dual of the cycle matroid $M(G)$. Weak elimination now applies to bonds. Since a cycle must return to its starting point, it cannot intersect a bond in exactly one edge. This generalizes to matroids as another characterization of cocircuits.

8.2.39. Theorem. The cocircuits of a matroid M on E are the minimal nonempty sets $C^* \subseteq E$ such that $|C^* \cap C| \neq 1$ for every $C \in \mathbf{C}$.

Proof: To show that every cocircuit has this property, suppose that $C \in \mathbf{C}$, $C^* \in \mathbf{C}^*$, $C^* \cap C = e$. Then $C - e \in \mathbf{I}$ and $C^* - e \in \mathbf{I}^*$, and the dual augmentation property yields $B \in \mathbf{B}$ and $\overline{B} \in \mathbf{B}^*$ such that $C - e \subseteq B$ and $C^* - e \subseteq \overline{B}$. Since e must appear in B or B^*, we obtain $C \in \mathbf{I}$ or $C^* \in \mathbf{I}^*$.

For the converse, we show that every nonempty set in $\mathbf{I}*$ meets some $C \in \mathbf{C}$ in one element; since cocircuits do not, every *minimal* set that does not is a cocircuit. Choose $X^* \in \mathbf{I}^*$. Let B^* be a cobase containing X^*, and let $B = \overline{B^*}$. For each $e \in X^*$, $B + e$ contains a circuit C, and $X^* \cap C = \{e\}$. ■

MATROID MINORS AND PLANAR DUALS

From a graph G we can obtain smaller graphs by repeatedly deleting and/or contracting edges. The resulting graphs are the **minors** of G. Wagner [1937] proved that G is planar if and only if it does not have K_5 or $K_{3,3}$ as a minor (Exercise 6.2.12). Hadwiger [1943] conjectured that G is k-colorable if G has no minor isomorphic to K_{k+1}. A simple graph is a forest if and only if it does not have C_3 as a minor.

To generalize these operations to matroids, we need to know how deletion and contraction affect cycle matroids. The acyclic subsets of $E(G-e)$ are precisely the acyclic subsets of $E(G)$ that omit e. The acylic subsets of $E(G \cdot e)$ are the subsets of $E(G) - e$ whose union with e is acyclic in G. A dual description of contraction is more convenient: X contains a spanning tree of each component of $G \cdot e$ if and only if $X + e$ contains a spanning tree of each component of G.

We also want the notation to extend in a natural way. This causes difficulty, because discussion of graph minors often emphasizes the edges removed, while discussion of matroid minors emphasizes the elements that remain. We compromise by using matroid notation for the matroid on the set that remains while extending graph notation to describe matroids obtained by deleting or contracting one element.

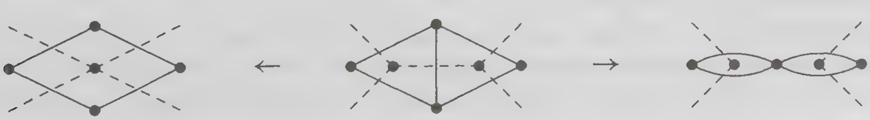
8.2.40. Definition. For a hereditary system M on E, the **restriction** *of M to $F \subseteq E$*, denoted $M|F$ and obtained by **deleting** $\overline{F}$, is the hereditary system defined by $\mathbf{I}_{M|F} = \{X \subseteq F\colon X \in \mathbf{I}_M\}$. The **contraction** *of M to $F \subseteq E$*, denoted $M.F$ and obtained by **contracting** $\overline{F}$, is the hereditary system defined by $\mathbf{S}_{M.F} = \{X \subseteq F\colon X \cup \overline{F} \in \mathbf{S}_M\}$. When $F = E - e$, we write $M - e = M|F$ and $M \cdot e = M.F$. The **minors** of M are the hereditary systems arising from M using deletions and contractions.

The definitions imply that $M|F$ and $M.F$ are hereditary systems. The operations of restriction and contraction commute (Exercise 41). The definition of contraction via supbases yields a natural duality between these operations.

8.2.41. Proposition. For hereditary systems, restriction and contraction are dual operations: $(M.F)^* = (M^*|F)$ and $(M|F)^* = (M^*.F)$.

Proof: $\mathbf{I}_{(M.F)^*} = \{X \subseteq F\colon F - X \in \mathbf{S}_{M.F}\} = \{X \subseteq F\colon (F-X) \cup \overline{F} \in \mathbf{S}_M\}$
$= \{X \subseteq F\colon \overline{X} \in \mathbf{S}_M\} = \{X \subseteq F\colon X \in \mathbf{I}_{M^*}\} = \mathbf{I}_{M^*|F}$.
For the second statement, apply the first to M^* and take duals. ■

The duality between deletion and contraction is most intuitive for plane graphs. Deleting an edge e in a plane graph G contracts the corresponding dual edge in G^*; contracting e deletes the edge in the dual.



8.2.42. Corollary. Under deletion or contraction of an edge e in a graph G, the cycle matroid and bond matroid behave as listed below.

$$\begin{array}{ll} M(G-e) = M(G) - e & M^*(G-e) = M^*(G)\cdot e \\ M(G\cdot e) = M(G)\cdot e & M^*(G\cdot e) = M^*(G) - e \end{array}$$

Proof: Matroid deletion and contraction are defined so that the statements in the first column describe the behavior of cycle matroids. Using these and Proposition 8.2.41, we compute

$$M^*(G-e) = [M(G-e)]^* = [M(G)-e]^* = M^*(G)\cdot e, \text{ and}$$
$$M^*(G\cdot e) = [M(G\cdot e)]^* = [M(G)\cdot e]^* = M^*(G) - e. \qquad \blacksquare$$

As desired, restrictions and contractions of matroids are matroids.

8.2.43. Theorem. Given $F \subseteq E$ and a matroid M on E, both $M|F$ and $M.F$ are matroids on F. In terms of r_M, their rank functions are $r_{M|F}(X) = r_M(X)$ and $r_{M.F}(X) = r_M(X \cup \overline{F}) - r_M(\overline{F})$.

Proof: The augmentation property from M applies to any pair of sets in $\mathbf{I}_{M|F}$; thus $M|F$ satisfies the augmentation property and is a matroid. Using duality, $M.F = (M^*|F)^*$ is also a matroid. The rank function for $M|F$ follows from the definition of $\mathbf{I}_{M|F}$. This and repeated application of Theorem 8.2.34 to $(M^*|F)^*$ yields the rank function for $M.F$ (Exercise 42). $\blacksquare$

The formula for $r_{M.F}$ yields a description of the independent sets: $X \in \mathbf{I}_{M.F}$ if and only if adding X to $\overline{F}$ increases the rank by $|X|$.

A set of edges in a plane graph G forms a cycle if and only if the corresponding dual edges form a bond in G^* (Theorem 6.1.14). Using the natural bijection between edges and dual edges, this tells us that the cycle matroid of a plane graph G is (isomorphic to) the bond matroid of G^*. By Corollary 8.2.37, the bond matroid of a graph H is $[M(H)]^*$. Applying this to G and to G^* tells us that the bond matroid of G is (isomorphic to) the cycle matroid of G^*. Thus the bond matroid of a planar graph G is graphic. Using Kuratowski's Theorem, we will prove that this condition characterizes planarity.

Whitney [1933a] approached this by defining a non-geometric notion of dual. Changing his definition slightly, we say that H is an **abstract dual** of G if there is a bijection $\phi\colon E(G) \to E(H)$ such that $X \subseteq E(G)$ is a bond in G if and only if $\phi(X)$ is the edge set of a cycle in H. With this definition, saying that G has an abstract dual is the same as saying that the bond matroid of G is graphic; the bijection ϕ establishes an isomorphism between $M^*(G)$ and $M(H)$.

8.2.44. Theorem. (Whitney [1933a]) A graph G is planar if and only if its bond matroid $M^*(G)$ is graphic.

Proof: We first prove that existence of an abstract dual is preserved under deletion and contraction of edges. Suppose that G has an abstract dual H, so that $M(H) \cong M^*(G)$. Let e' be the edge of H corresponding to e under the bijection. To prove that $H \cdot e'$ is an abstract dual of $G - e$ and that $H - e'$ is an abstract dual of $G \cdot e$, we use Corollary 8.2.42 to compute

$$M^*(G - e) = M^*(G) \cdot e \cong M(H) \cdot e' = M(H \cdot e'), \text{ and}$$
$$M^*(G \cdot e) = M^*(G) - e \cong M(H) - e' = M(H - e').$$

We have demonstrated that planar graphs have abstract duals. By Kuratowski's Theorem, a nonplanar graph contains a subdivision K_5 or $K_{3,3}$. Hence K_5 or $K_{3,3}$ is a minor of it. Since existence of abstract duals is preserved under deletion and contraction, showing that K_5 and $K_{3,3}$ have no abstract dual implies that every nonplanar graph has no abstract dual.

If H is an abstract dual of G, then also G is an abstract dual of H, since $M^*(G) \cong M(H)$ if and only if $M(G) \cong M^*(H)$. If G has girth g, then bonds of H have size at least g, so $\delta(H) \geq g$. Also $e(H) = e(G)$, and the degree-sum formula yields $n(H) \leq \lfloor 2e(H)/\delta(H) \rfloor \leq \lfloor 2e(G)/g \rfloor$.

Let H be an abstract dual of K_5. Since K_5 has girth 3, $n(H) \leq \lfloor 20/3 \rfloor = 6$. Since all bonds of K_5 have four or six edges, all cycles of H have four or six edges, and thus H is a simple bipartite graph. However, no simple bipartite graph with at most six vertices has ten edges.

Let H be an abstract dual of $K_{3,3}$. Since $K_{3,3}$ has girth 4, $n(H) \leq \lfloor 18/4 \rfloor = 4$. Since all bonds of $K_{3,3}$ have at least three edges, all cycles of H have at least three edges, and thus H is a simple graph. However, no simple graph with at most four vertices has nine edges. ■

The argument that bond matroids of plane graphs are graphic shows that every "geometric" dual of a planar graph is an abstract dual. We have seen that the geometric dual need not be unique. Nevertheless, the cycle matroid of every graph dual to G must be $M^*(G)$; hence all geometric duals of G have the same cycle matroid. Whitney [1933b] determined when graphs have the same cycle matroid (see Exercise 45, also Kelmans [1980, 1987, 1988]).

Minors have many applications. They will soon help us prove the Matroid Intersection Theorem. They are used in characterizing classes of matroids by forbidden substructures; for example, a matroid is binary if and only if it does not have $U_{2,4}$ as a minor. Minors also are used to produce a winning strategy for a matroid generalization of Bridg-it (Theorem 2.1.17).

8.2.45.* Definition. Given $e \in E$ and a matroid M on E, the **Shannon Switching Game** (M, e) is played by the Spanner and the Cutter. The Cutter deletes elements of $E - e$ and the Spanner seizes them, one per move. The Spanner aims to seize a set that spans e, and the Cutter aims to prevent this. The Cutter moves first.

Having the Spanner move first can be simulated by adding an element e' such that $\{e, e'\}$ is a circuit; the Cutter must begin by deleting e' to avoid losing immediately. Bridg-it occurs by letting M be the cycle matroid of the graph in Theorem 2.1.17 with e the "auxiliary edge" and e' an extra auxiliary edge. The spanning tree strategy for the Spanner results from the following sufficient condition for a winning strategy. The condition is also necessary, but proving that requires the Matroid Union Theorem (Theorem 8.2.55).

8.2.46.* Theorem. (Lehman [1964]) In the Shannon Switching Game (M, e), the Spanner has a winning strategy if there are disjoint subsets X_1, X_2 of $E - e$ such that $e \in \sigma(X_1) = \sigma(X_2)$.

Proof: We use X_1, X_2 to produce a winning strategy. Let $X = \sigma(X_1) = \sigma(X_2)$. Since the Spanner can ignore deletions outside X and play in $M|(X+e)$, we may assume that X_1, X_2 are disjoint bases. If the Cutter plays g and the Spanner plays f, then g is no longer available and f cannot be deleted; the effect is deletion and contraction. Letting $M' = (M - g) \cdot f$, we have $e \in \sigma_{M'}(X)$ if and only if $g \notin X$ and $e \in \sigma_M(X + f)$. The Spanner wins if e is a loop in M', which is equivalent to $e \in \sigma_M(F)$, where F is the set seized by the Spanner.

If $|E| = 1$, then e is a loop and the Spanner wins; we proceed by induction on $|E|$. It suffices to provide an immediate answer f to g so that $M' = (M - g) \cdot f$ has two disjoint bases. If the Cutter deletes g not in X_1 or X_2, then the Spanner seizes an arbitrary f, and the two sets $X_1 - g - f$ and $X_2 - g - f$ are disjoint and spanning in M'. Hence we may assume that $g \in X_1$. The base exchange property yields $f \in X_2$ such that $X' = X_1 - g + f \in \mathbf{B}$. Now $X' - f$ and $X_2 - f$ are disjoint bases avoiding e in the game (M', e). ■

MATROID INTERSECTION

Matroid theory took a great leap forward with the proof of the Matroid Intersection and Union Theorems by Edmonds. This provided a unified context for many well-known min-max relations, which became corollaries. We have proved some of these in earlier chapters. Yielding a simple unified proof for many important theorems, the Matroid Intersection Theorem can be considered among the most beautiful theorems of combinatorics.

The Matroid Intersection Theorem is a min-max relation for common independent sets in two matroids on the same ground set. We can view the intersection of two matroids as a hereditary system, but *not* as a matroid. For multiple matroids on a set E, we typically use subscripts to distinguish corresponding aspects, as in $\mathbf{B}_i$ for the bases of M_i, etc. We still use $\overline{X}$ to denote the complement of X within the ground set E.

8.2.47. Definition. Given hereditary systems M_1, M_2 on E, the **intersection** of M_1 and M_2 is the hereditary system whose independent sets are $\{X \subseteq E\colon X \in \mathbf{I}_1 \cap \mathbf{I}_2\}$.

For example, the intersection of the two natural partition matroids on the edges of a bipartite graph G has as its independent sets the matchings of G. These are generally not the independent sets of a matroid (see Exercises 1–2), and thus the greedy algorithm does not solve maximum-weighted matching.

Recall that a *loop* is an element forming a nonempty set of rank 0.

8.2.48. Theorem. (Matroid Intersection Theorem, Edmonds [1970]) For matroids M_1, M_2 on E, the size of a largest common independent set satisfies

$$\max\{|I| : I \in \mathbf{I}_1 \cap \mathbf{I}_2\} = \min_{X \subseteq E}\{r_1(X) + r_2(\overline{X})\}.$$

Proof: (Seymour [1976]) For weak duality, consider arbitrary $I \in \mathbf{I}_1 \cap \mathbf{I}_2$ and $X \subseteq E$. The sets $I \cap X$ and $I \cap \overline{X}$ are also common independent sets, and $|I| = |I \cap X| + \left|I \cap \overline{X}\right| \le r_1(X) + r_2(\overline{X})$.

To achieve equality, we use induction on $|E|$; when $|E| = 0$ both sides are 0. If every element of E is a loop in M_1 or in M_2, then $\max|I| = 0 = r_1(X) + r_2(\overline{X})$, where X consists of all loops in M_1. Hence we may assume that $|E| > 0$ and that some $e \in E$ is a non-loop in both matroids. Let $F = E - e$, and consider the matroids $M_1|F$, $M_2|F$, $M_1.F$, and $M_2.F$.

Let $k = \min_{X \subseteq E}\{r_1(X) + r_2(\overline{X})\}$; we seek a common independent k-set in M_1 and M_2. If there is none, then $M_1|F$ and $M_2|F$ have no common independent k-set, and $M_1.F$ and $M_2.F$ have no common independent $k-1$-set. The induction hypothesis and rank formulas (Theorem 8.2.43) yield

$$\begin{aligned} r_1(X) + r_2(F - X) &\le k - 1 && \text{for some } X \subseteq F, \text{ and} \\ r_1(Y + e) - 1 + r_2(F - Y + e) - 1 &\le k - 2 && \text{for some } Y \subseteq F. \end{aligned}$$

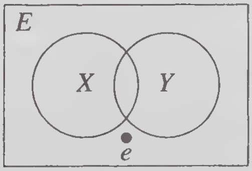
We use $(F - Y) + e = \overline{Y}$ and $F - X = \overline{X + e}$ and sum the two inequalities:

$$r_1(X) + r_2(\bar{X} + e) + r_1(Y + e) + r_2(\overline{Y}) \le 2k - 1.$$

Now we apply submodularity of r_1 to X and $Y + e$ and submodularity of r_2 to $\overline{Y}$ and $\overline{X + e}$. For clarity, write $U = X + e$ and $V = Y + e$. Applying this to the preceding inequality yields

$$r_1(X \cup V) + r_1(X \cap V) + r_2(\overline{Y} \cup \overline{U}) + r_2(\overline{Y} \cap \overline{U}) \le 2k - 1.$$

Since $\overline{Y} \cap \overline{U} = \overline{X \cup V}$ and $\overline{Y} \cup \overline{U} = \overline{X \cap V}$, the left side sums two instances of $r_1(Z) + r_2(\overline{Z})$, and the hypothesis $k \le r_1(Z) + r_2(\overline{Z})$ for all $Z \subseteq E$ yields $2k \le 2k - 1$. Hence M_1 and M_2 do have a common independent k-set. ■



It can be helpful to restrict the range of the minimization.

8.2.49. Corollary. The maximum size of a common independent set in matroids M_1, M_2 on E is the minimum of $r_1(X_1) + r_2(X_2)$ over sets X_1, X_2 such that $X_1 \cup X_2 = E$ and each X_i is closed in M_i.

Proof: The incorporation property implies that $r_i(\sigma_i(X)) = r_i(X)$. ■

We have proved special cases of the Matroid Intersection Theorem by other means. We proved the König–Egerváry Theorem in various ways, and we proved the Ford–Fulkerson characterization of CSDRs from Menger's Theorem in Theorem 4.2.25. Whenever we have two matroids on the same set, the Matroid Intersection Theorem tells us that there must be a min-max relation for the maximum size of a common independent set, tells us what the result should be, and provides a proof.

8.2.50. Corollary. (König [1931], Egerváry [1931]) In a bipartite graph, the largest matching and smallest vertex cover have equal size.

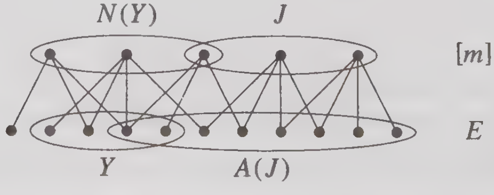
Proof: When M_1 and M_2 are the partition matroids on $E(G)$ induced by the partite sets U_1, U_2 of G, the matchings are the common independent sets. For $X_1, X_2 \subseteq E$, the rank $r_i(X_i)$ counts the vertices of U_i incident to edges in X_i. Hence if $X_1 \cup X_2 = E$, then G has a vertex cover of size $r_1(X_1) + r_2(X_2)$, using vertices of U_i to cover X_i. Conversely, if $T_1 \cup T_2$ is a vertex cover with $T_i \subseteq U_i$, let X_i be the set of edges incident to T_i; we have $X_1 \cup X_2 = E$ with X_i closed in M_i and $r_1(X_1) + r_2(X_2) = |T_1| + |T_2|$. We conclude that

$$\alpha'(G) = \max\{|I|: I \in \mathbf{I}_1 \cap \mathbf{I}_2\} = \min\{r_1(X_1) + r_2(X_2)\} = \beta(G). \quad \blacksquare$$

The next corollary uses the rank function for transversal matroids.

8.2.51. Example. *Transversal matroids* (see Example 8.2.13). Suppose that $A_1 \cup \cdots \cup A_m = E$, and let G be the corresponding incidence graph with partite sets E and $[m]$. Consider $X \subseteq E$. If $|N(Y)| < |Y|$ for some $Y \subseteq X$, then Y forces at least $|Y| - |N(Y)|$ unsaturated elements in X. Hall's Condition applied to X yields $r(X) = \min\{|X| - (|Y| - |N(Y)|): Y \subseteq X\}$ (Exercise 51).

We obtain another expression for $r(X)$ (see Ore [1955]). Let $A(J) = \bigcup_{i \in J} A_i$; in terms of the graph, $A(J) = N(J)$. By applying Hall's Condition to $[m]$ instead of E, we can write the maximum size of a matching as $r(M) = \min\{m - (|J| - |A(J)|): J \subseteq [m]\}$. To determine the maximum number of elements in $X \subseteq E$ that can be matched, we discard the elements of $E - X$, obtaining $r(X) = \min_{J \subseteq [m]}\{|A(J) \cap X| - |J| + m\}$.



The first formula for $r(X)$ uses neighborhoods of subsets of E; the second uses neighborhoods of subsets of $[m]$. Exercise 53 shows directly that the second rank formula is the rank function of a matroid, without relying on results from bipartite matching. Further material on transversals appears in Mirsky [1971] and in Lovász–Plummer [1986]. ■

8.2.52. Corollary. (Ford–Fulkerson [1958]) Families $\mathbf{A} = \{A_1, \ldots, A_m\}$ and $\mathbf{B} = \{B_1, \ldots, B_m\}$ have a common system of distinct representatives (CSDR) if and only if, for each $I, J \subseteq [m]$,

$$\Big|\big(\bigcup_{i\in I} A_i\big) \cap \big(\bigcup_{j\in J} B_j\big)\Big| \geq |I| + |J| - m.$$

Proof: A common partial SDR is a common independent set in the two transversal matroids M_1, M_2 induced on E by $\mathbf{A}$ and $\mathbf{B}$. To determine when there is a complete CSDR, we need only restate the condition $r_1(X) + r_2(\overline{X}) \geq m$ to find the appropriate condition on the set systems.

The rank formulas from Example 8.2.51 yield

$$r_1(X) + r_2(\overline{X}) = \min_{I\subseteq[m]}\{|A(I)\cap X| - |I| + m\} + \min_{J\subseteq[m]}\{\big|B(J)\cap \overline{X}\big| - |J| + m\}.$$

Hence $r_1(X) + r_2(\overline{X}) \geq m$ for all X if and only if

$$|A(I)\cap X| + \big|B(J)\cap \overline{X}\big| \geq |I| + |J| - m \text{ for all } X \subseteq E \text{ and } I, J \subseteq [m].$$

Given I, J, consider the contribution of an element of E to the left side. Each element of $A(I)\cap B(J)$ counts once whether it belongs to X or $\overline{X}$. Elements of $A(I) - B(J)$ count if and only if they belong to X, and elements of $B(J) - A(I)$ count if and only if they belong to $\overline{X}$. Hence the left side is minimized for I, J when $A(I) - B(J) \subseteq \overline{X}$ and $B(J) - A(I) \subseteq X$. In this case the left side equals $|A(I) \cap B(J)|$, which yields the Ford–Fulkerson condition. ∎

The augmenting path approach to maximum bipartite matching generalizes to matroid intersection. The algorithm yields a common independent set I of maximum size and a set X such that $r_1(X) + r_2(\overline{X}) = |I|$ (see Lawler [1976], Edmonds [1979], Faigle [1987]). Finding a maximum common independent set in three matroids is NP-complete (??s –).

MATROID UNION

The intersection of two matroids is seldom a matroid, but a natural concept of matroid union does always yield a matroid. Together with a useful min-max relation for the rank function, this is the content of the Matroid Union Theorem. The Matroid Intersection and Union Theorems are equivalent; they can be derived from each other. Welsh [1976] proves the Matroid Union Theorem first; here we obtain it from the Matroid Intersection Theorem.

8.2.53. Definition. The **union** $M_1 \cup \cdots \cup M_k$ of hereditary systems $M_1, \ldots, M_k$ on E is the hereditary system M on E defined by $\mathbf{I}_M = \{I_1 \cup \cdots \cup I_k \colon I_i \in \mathbf{I}_i\}$. The **direct sum** $M_1 \oplus \cdots \oplus M_k$ of hereditary systems $M_1, \ldots, M_k$ on disjoint sets $E_1, \ldots, E_k$ is the hereditary system M on $E_1 \cup \cdots \cup E_k$ defined by $\mathbf{I}_M = \{I_1 \cup \cdots \cup I_k : I_i \in \mathbf{I}_i\}$.

The direct sum $M_1 \oplus \cdots \oplus M_k$ on $E_1, \ldots, E_k$ can be expressed as the union of $M_1', \ldots, M_k'$ on $E' = E_1 \cup \cdots \cup E_k$ by letting M_i' be a copy of M_i with the additional elements of $E' - E_i$ added as loops. When each M_i is a uniform matroid, the direct sum is a **generalized partition matroid**. Here $E_1, \ldots, E_k$ partition E, there are positive integers $r_1, \ldots, r_k$, and $X \in \mathbf{I}$ if $|X \cap E_i| \le r_i$. The partition matroids defined earlier arise when all $r_i = 1$.

8.2.54. Proposition. Given matroids $M_1, \ldots, M_k$ on disjoint sets $E_1, \ldots, E_k$, the direct sum $M = M_1 \oplus \cdots \oplus M_k$ is a matroid.

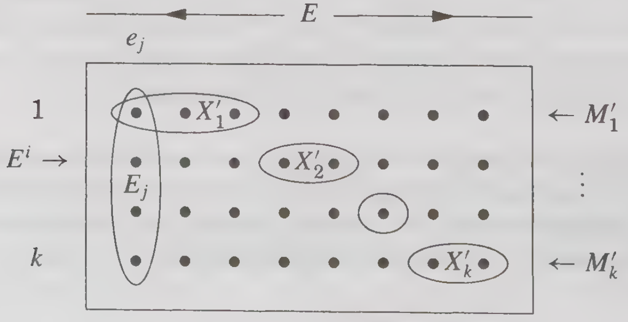
Proof: Since the $E_1, \ldots, E_k$ are pairwise disjoint, the intersection of any $I \in \mathbf{I}$ with each E_i is independent in M_i. If $I_1, I_2 \in \mathbf{I}$ with $|I_2| > |I_1|$, then $|I_2 \cap E_i| > |I_1 \cap E_i|$ for some i. Since both sets are independent in M_i, we can augment $I_1 \cap E_i$ from $I_2 \cap E_i$ and therefore I_1 from I_2. Hence $M_1 \oplus \cdots \oplus M_k$ satisfies the augmentation property. ■

Using a direct sum, we prove that the union of matroids is always a matroid, and we compute the rank function.

8.2.55. Theorem. (Matroid Union Theorem—Edmonds–Fulkerson [1965], Nash-Williams [1966]) If $M_1, \ldots, M_k$ are matroids on E with rank functions $r_1, \ldots, r_k$, then the union $M = M_1 \cup \cdots \cup M_k$ is a matroid with rank function $r(X) = \min_{Y \subseteq X}(|X - Y| + \sum r_i(Y))$.

Proof: (following Schrijver [to appear]). After proving the formula for the rank function, we will verify the submodularity property to prove that M is a matroid. First we reduce the computation of the rank function to the computation of $r(E)$. In the restriction of the hereditary system M to the set X, we have $\mathbf{I}_{M|X} = \{Y \subseteq X : Y \in \mathbf{I}_M\}$ and $r_{M|X}(Y) = r_M(Y)$ for $Y \subseteq X$. Thus $M|X = \cup_i (M_i|X)$, and applying the formula for the rank of the full union to $M|X$ yields $r_M(X)$.

Consider a k by $|E|$ grid of elements E' in which the jth column E_j consists of k copies of the element $e_j \in E$. We define two matroids N_1, N_2 on E' such that the maximum size of a set independent in both N_1 and N_2 equals the maximum size of a set independent in M. We then compute $r_M(E)$ by applying the Matroid Intersection Theorem to N_1 and N_2. Let M_i' be a copy of M_i defined on the elements E^i of row i in E'. Let N_1 be the direct sum matroid $M_1' \oplus \cdots \oplus M_k'$, and let N_2 be the partition matroid induced on E' by the column partition $\{E_j\}$.



Each set $X \in \mathbf{I}_M$ has a decomposition as a disjoint union of subsets $X_i \in \mathbf{I}_i$, because $\mathbf{I}_i$ is a hereditary family. Given a decomposition $\{X_i\}$ of $X \in \mathbf{I}_M$, let X_i' be the copy of X_i in E^i. Since $\{X_i\}$ are disjoint, $\cup X_i'$ is independent in N_2, and $X_i \in \mathbf{I}_i$ implies that $\cup X_i'$ is also independent in N_1. From $X \in \mathbf{I}_M$, we have constructed $\cup X_i'$ of size $|X|$ in $\mathbf{I}_{N_1} \cap \mathbf{I}_{N_2}$. Conversely, any $X' \in \mathbf{I}_{N_1} \cap \mathbf{I}_{N_2}$ corresponds to a decomposition of a set in $\mathbf{I}_M$ of size $|X'|$ when the sets $X' \cap E^i$ are transferred back to E, because N_2 forbids multiple copies of elements.

Hence $r(E) = \max\{|I| : I \in \mathbf{I}_{N_1} \cap \mathbf{I}_{N_2}\}$. To compute this, let the rank functions of N_1, N_2 be q_1, q_2, and let r_i' be the rank function of the copy M_i' of M_i on E^i. We have $q_1(X') = \sum r_i'(X' \cap E^i)$, and $q_2(X')$ is the number of elements of E that have copies in X'. The Matroid Intersection Theorem yields $r(E) = \min_{X' \subseteq E'}\{q_1(X') + q_2(E' - X')\}$.

By Corollary 8.2.49, the minimum is achieved by a set X' such that $E' - X'$ is closed in N_2. The closed sets in the partition matroid N_2 are the sets that contain all or none of the copies of each element—the unions of full columns of E'. Given X' with $E' - X'$ closed in N_2, let $Y \subseteq E$ be the set of elements whose copies comprise X'. Then $q_2(E' - X') = |E - Y|$, and X' contains all copies of the elements of Y, so $q_1(X') = \sum r_i'(X' \cap E^i) = \sum r_i(Y)$. We conclude that $r(E) = \min_{Y \subseteq E}\{|E - Y| + \sum r_i(Y)\}$.

To show that M is a matroid, we verify submodularity for r. Given $X, Y \subseteq E$, the formula for r yields $U \subseteq X$ and $V \subseteq Y$ such that

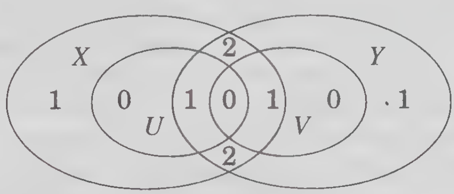
$$r(X) = |X - U| + \sum r_i(U); \qquad r(Y) = |Y - V| + \sum r_i(V).$$

Since $U \cap V \subseteq X \cap Y$ and $U \cup V \subseteq X \cup Y$, we also have

$$r(X \cap Y) \le |(X \cap Y) - (U \cap V)| + \sum r_i(U \cap V);$$

$$r(X \cup Y) \le |(X \cup Y) - (U \cup V)| + \sum r_i(U \cup V).$$

After applying the submodularity of each r_i and the diagram below, these inequalities yield $r(X \cap Y) + r(X \cup Y) \le r(X) + r(Y)$. ■



$$|(X \cap Y) - (U \cap V)| + |(X \cup Y) - (U \cup V)| = |X - U| + |Y - V|$$

In applying the Matroid Intersection Theorem, we needed N_1 to be a matroid, which required $\{M_i\}$ to be matroids. Hence this rank formula does not apply for unions of arbitrary hereditary systems.

The Matroid Union Theorem yields short proofs of min-max relations for packing and covering problems. In each formula below, the optimal subset is closed, since switching from X to $\sigma(X)$ improves the numerator without changing the denominator. The graph corollaries originally had difficult ad hoc proofs.

8.2.56. Corollary. (Matroid Covering Theorem—Edmonds [1965b]) In a loopless matroid M on E, the minimum number of independent sets whose union is E is $\max_{X\subseteq E} \left\lceil \frac{|X|}{r(X)} \right\rceil$.

Proof: Let $M_1, \ldots, M_k$ be copies of M on E. The set E is the union of k independent sets in M if and only if E is independent in $M' = M_1 \cup \cdots \cup M_k$. By the Matroid Union Theorem, $r'(E) \geq |E|$ is equivalent to $|E| - |Y| + \sum r_i(Y) \geq |E|$ for all $Y \subseteq E$. Since $r_i(Y) = r(Y)$ for all i, we conclude that E is the union of k independent sets if and only if $kr(Y) \geq |Y|$ for all $Y \subseteq E$. ■

8.2.57. Corollary. (Nash-Williams [1964]) The minimum number of forests needed to cover the edges of a graph G (its **arboricity**) is $\max_{H\subseteq G} \left\lceil \frac{e(H)}{n(H)-1} \right\rceil$.

Proof: (Edmonds [1965b]) This follows immediately by applying Corollary 8.2.56 to $M(G)$. The best lower bound arises from a connected induced subgraph H (corresponding to a closed set in $M(G)$). ■

8.2.58. Corollary. (Matroid Packing Theorem—Edmonds [1965c]) Given a matroid M on E, the maximum number of pairwise disjoint bases equals $\min_{X:\, r(X)<r(E)} \left\lfloor \frac{|E|-|X|}{r(E)-r(X)} \right\rfloor$.

Proof: The set E contains k disjoint bases if and only if $r'(E) \geq kr(E)$ in the union M' of k matroids $M_1, \ldots, M_k$ that are copies of M on E. By the Matroid Union Theorem, this requires $|E| - |Y| + \sum r_i(Y) \geq kr(E)$ for all $Y \subseteq E$. Since $r_i(Y) = r(Y)$ for all i, we conclude that k disjoint bases exist if and only if $|E| - |Y| \geq k(r(e) - r(Y))$ for all $Y \subseteq E$. ■

8.2.59. Corollary. (Nash-Williams [1961], Tutte [1961a]) A graph G has k pairwise edge-disjoint spanning trees if and only if, for every vertex partition P, there are at least $k(|P| - 1)$ edges with endpoints in different sets of P.

Proof: (Edmonds [1965c]) We may assume that G is connected. By applying Corollary 8.2.58 to $M(G)$, we must determine when $|E| - |X| \geq k(r(E) - r(X))$ for each closed set X. The closed sets correspond to partitions of $V(G)$ into vertex sets inducing connected subgraphs. For each such partition $V_1, \ldots, V_p$, the corresponding closed set X is $\bigcup E(G[V_i])$ with rank $n - p$. Since $|E| - |X|$ counts the edges between sets of the partition and $r(E) - r(X) \geq p - 1$, the graph has k disjoint spanning trees if and only if the condition holds. ■

EXERCISES

8.2.1. (−) Show that the stable sets of a graph need not be the independent sets of a matroid by finding vertex-weighted graphs where the ratio between the maximum weight of a stable set and the weight of a stable set found greedily is arbitrarily large.

8.2.2. (−) Characterize the graphs whose stable sets form the family of independent sets of a matroid on the set of vertices.

8.2.3. (−) Show that every partition matroid is a transversal matroid.

8.2.4. Modify the greedy algorithm to obtain (with proof) an algorithm for finding the maximum-weighted independent set in a matroid with arbitrary real weights (not necessarily nonnegative) on the elements.

8.2.5. Characterize the graphs whose matchings form the family of independent sets of a matroid on the set of edges.

8.2.6. (!) Determine which uniform matroids are graphic. Characterize the graphs whose cycle matroids are uniform matroids.

8.2.7. (!) Determine which partition matroids are graphic. Characterize the graphs whose cycle matroids are partition matroids.

8.2.8. Using only linear dependence, prove that vectorial matroids satisfy the induced circuit property: adding an element to a linearly independent set of vectors creates at most one minimal dependent set.

8.2.9. Describe the circuits of a transversal matroid M in terms of the corresponding bipartite graph G. Using only properties of bipartite graphs, prove that M satisfies the weak elimination property.

8.2.10. Let $M(G)$ be the cycle matroid of G. Let $k(X)$ be the number of components of the spanning subgraph G_X with edge set X; so $r(X) = n - k(X)$. Let U and V be the sets of components in G_X and G_Y, respectively. Let H be the U, V-bigraph with $u \leftrightarrow v$ when the components corresponding to u and v intersect.

a) Count the vertices and components of H in terms of the numbers $k(X)$, $k(Y)$, and $k(X \cap Y)$. Prove that $k(X \cup Y) \geq e(H)$.

b) Use part (a) to prove the submodularity property for $M(G)$ without using other properties of matroids. (Aigner [1979])

8.2.11. Use the König–Egerváry Theorem to prove directly that the rank function of a transversal matroid is submodular.

8.2.12. Let D be a digraph with distinguished source s and sink t. Let $E = V(D) - \{s, t\}$. For $X \subseteq E$, let $r(X)$ be the number of edges from $s \cup X$ to $\overline{X} \cup t$. Prove that r is submodular.

8.2.13. (−) For an element x in a hereditary system, prove that the following properties are equivalent and characterize loops.

a) $r(x) = 0$.

b) $x \in \sigma(\emptyset)$.

c) x is a circuit.

d) x belongs to no base.

e) Every set containing x is dependent.

f) x belongs to the span of every $X \subseteq E$.

8.2.14. (−) Prove equivalence of the following characterizations of parallel elements, assuming that $x \neq y$ and neither is a loop.

a) $r(x, y) = 1$.

b) $\{x, y\} \in \mathbf{C}$.

c) $x \in \sigma(y)$, $y \in \sigma(x)$, $r(x) = r(y) = 1$.

Furthermore, show that if x, y are parallel and $x \in \sigma(X)$, then $y \in \sigma(X)$.

8.2.15. (−) Suppose that $r(X) = r(X \cap Y)$ for some $X, Y \subseteq E$ in a matroid on E. Prove that $r(X \cup Y) = r(Y)$. Does the converse hold?

8.2.16. Let M be a hereditary system with nonnegative weights on E. Prove directly that if M satisfies the base exchange property (B), then the greedy algorithm always generates a maximum-weighted base.

8.2.17. *Alternative matroid axiomatics.* Let M be a hereditary system. Prove the following implications directly for M.

a) (−) Submodularity (R) implies weak absorption (A).

b) Strong absorption (A′) implies submodularity (R) (without using uniformity). (Hint: Use induction on $|X \triangle Y|$.)

c) Base exchange (B) implies uniqueness of induced circuits (J).

d) (−) Uniqueness of induced circuits (J) implies weak elimination (C).

e) Uniqueness of induced circuits (J) implies augmentation (I). (Hint: Use J and induction on $|I_1 - I_2|$ to obtain the augmentation.)

8.2.18. Prove that a hereditary system is a matroid if and only if it satisfies the "ultra-weak" augmentation property: If $I_1, I_2 \in \mathbf{I}$ with $|I_2| > |I_1|$ and $|I_1 - I_2| = 1$, then $I_1 + e \in \mathbf{I}$ for some $e \in I_2 - I_1$. (Chappell [1994a])

8.2.19. (−) Let M be a matroid on E, and fix $A \subseteq E$. Obtain $\mathbf{I}'$ from $\mathbf{I}$ by deleting the sets that intersect A. Prove that $\mathbf{I}'$ is the family of independent sets of a matroid on E.

8.2.20. For a matroid on E with $e \notin B \in \mathbf{B}$, let $C(e, B)$ be the unique circuit in $B + e$.

a) For $e \notin B$, prove that $B - f + e$ is a base if and only if f belongs to $C(e, B)$.

b) For $e \in C \in \mathbf{C}$, prove that $C = C(e, B)$ for some base B.

8.2.21. (−) Let B_1, B_2 be bases of a matroid such that $|B_1 \triangle B_2| = 2$. Prove that there is a unique circuit C such that $B_1 \triangle B_2 \subseteq C \subseteq B_1 \cup B_2$.

8.2.22. (−) Let B_1, B_2 be bases in a matroid M. Given $X_1 \subseteq B_1$, prove that there exists $X_2 \subseteq B_2$ such that $(B_1 - X_1) \cup X_2$ and $(B_2 - X_2) \cup X_1$ are both bases of M. (Greene [1973])

8.2.23. (!) Let B_1, B_2 be distinct bases of a matroid M.

a) Let G be a B_1, B_2-bigraph with $e \in B_1$ adjacent to $f \in B_2$ when $B_2 + e - f \in \mathbf{B}$. Prove that G has a perfect matching.

b) Conclude from part (a) that there exists a bijection $\pi\colon B_1 \to B_2$ such that for each $e \in B_1$, the set $B_2 - \pi(e) + e$ is a base of M.

8.2.24. (!) Let B_1, B_2 be distinct bases of a matroid M.

a) Prove that for each $e \in B_1$, there is $f \in B_2$ such that $B_1 - e + f$ and $B_2 - f + e$ are bases. (Hint: Use the incorporation property. Note: This generalizes Exercise 2.1.34.)

b) Use the cycle matroid $M(K_4)$ to show that there may be no bijection $\pi\colon B_1 \to B_2$ such that e and $f = \pi(e)$ satisfy part (a) for all $e \in B_1$.

8.2.25. (−) A collection of $|E| - r(E)$ circuits of a matroid on E form a **fundamental set of circuits** if it is possible to order the elements $e_1, \ldots, e_n$ in such a way that C_i contains $e_{r(E)+i}$ but no higher-indexed element. Prove that every matroid has a fundamental set of circuits. (Whitney [1935])

8.2.26. (−) Given k distinct circuits $\{C_i\}$ with none contained in the union of the others, and given a set X with $|X| < k$, prove that $\bigcup_{i=1}^{k} C_i - X$ contains a circuit. (Welsh [1976])

8.2.27. (+) For a hereditary system, prove directly that the weak elimination property implies the strong elimination property, using induction on $|C_1 \cup C_2|$. (Lehman [1964])

8.2.28. (!) *Min-max relation for weighted independent set.* Let M be a matroid on E, with each $e \in E$ having nonnegative integer weight $w(e)$. Let $\mathbf{A}$ be the set of chains $X_1 \subseteq X_2 \subseteq \cdots$ such that each $e \in E$ appears in at least $w(e)$ sets in the chain (sets may repeat in the chain). Use the greedy algorithm to prove that

$$\max_{I \in \mathbf{I}} \sum_{e \in I} w(e) = \min_{\{X_i\} \in \mathbf{A}} \sum_i r(X_i).$$

8.2.29. (−) Let r and σ be the rank function and span function of a matroid. Prove that $r(X) = \min\{|Y| : Y \subseteq X, \sigma(Y) = \sigma(X)\}$.

8.2.30. Prove that a matroid of rank r has at least 2^r closed sets. (Lazarson [1957])

8.2.31. Prove that a matroid is simple if and only if 1) no element appears in every hyperplane, and 2) from every distinct pair of elements some hyperplane contains exactly one. Prove that these conditions also suffice for a family of sets to be the collection of hyperplanes of a simple matroid.

8.2.32. Prove that in a matroid, a set is a hypobase if and only if it is a hyperplane.

8.2.33. Use the weak elimination property to characterize when a family of sets is the family of hyperplanes of some matroid.

8.2.34. Prove that the closed sets of a matroid are the complements of the unions of cocircuits.

8.2.35. Let X be a closed set in a matroid M.

a) Let Y be a closed set contained in X such that $r(Y) = r(X) - 1$. Prove that M has a hyperplane H such that $Y = X \cap H$. (Hint: Given a maximal independent subset Z of Y, augment it by $e \in X$ and then to a base B, and let $H = \sigma(B - e)$.)

b) Prove that X is the intersection of $r(M) - r(X)$ distinct hyperplanes.

8.2.36. Prove the following properties of closed sets in a matroid.

a) The intersection of two closed sets is a closed set.

b) The span of a set is the intersection of all closed sets containing it. (Comment: Hence $\sigma(X)$ is the unique minimal closed set containing X.)

c) The union of two closed sets need not be a closed set.

8.2.37. Prove that $M.X$ has no loops if and only if $\overline{X}$ is closed.

8.2.38. (!) *Bases and cocircuits in matroids.*

a) Prove that when e belongs to a base B in a matroid M, there is exactly one cocircuit of M disjoint from $B - e$, and it contains e.

b) Use part (a) to prove that if C is a circuit of a matroid M and x, y are distinct elements of C, then there is a cocircuit $C^* \in \mathbf{C}^*$ with $C^* \cap C = \{x, y\}$. (Minty [1966])

c) Explain why part (b) is trivial for cycle matroids.

8.2.39. (−) Show that the dual of a simple matroid (no loops or parallel elements) need not be simple. Determine whether a set can be both a circuit and a cocircuit in a matroid.

8.2.40. (!) Use matroid duality to prove Euler's Formula for connected plane graphs.

8.2.41. Prove that any minor of a matroid obtained by restricting and then contracting can also be obtained by contracting and then restricting. In particular, if M is a matroid on E and $Y \subseteq X \subseteq E$, prove that $(M|X).Y = (M.\overline{X-Y})|Y$ and $(M.X)|Y = (M|\overline{X-Y}).Y$.

8.2.42. (!) Use duality and matroid restriction to prove that $r_{M.F}(X) = r_M(X \cup F) - r_M(F)$. Also derive the formula directly by proving that X is independent in $M.F$ if and only if adding X to $\overline{F}$ increases the rank by $|X|$.

8.2.43. Prove that the cycle matroid $M(G)$ is the column matroid over $\mathbb{Z}_2$ of the vertex-edge incidence matrix of G. (Hence every graphic matroid is binary.)

8.2.44. Tutte [1958] proved that a matroid if and only if it has no $U_{2,4}$-minor.

a) Prove that the matrix $\binom{1\,0\,1\,1}{0\,1\,1\,2}$ represents $U_{2,4}$ over $\mathbb{Z}_3$.

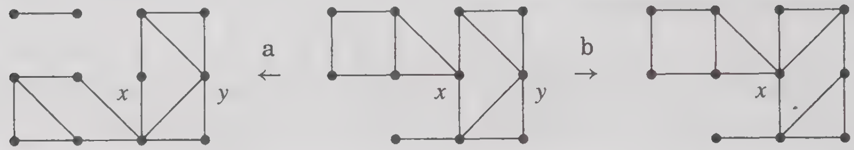
b) Prove that $U_{2,4}$ has no representation over $\mathbb{Z}_2$.

8.2.45. Prove that the three operations below preserve the cycle matroid of G.

a) Decompose G into its blocks $B_1, \ldots, B_k$, and reassemble them to form another graph G' with blocks $B_1, \ldots, B_k$.

b) In a block B of G that has a two-vertex cut $\{x, y\}$, interchange the neighbors of x and y in one of the components of $B - \{x, y\}$.

c) Add or delete isolated vertices.



(Comment: Whitney's 2-Isomorphism Theorem [1933b] states that G and H have the same cycle matroid if and only if some sequence of these operations turns G into H. Thus every 3-connected planar graph has only one dual graph, meaning essentially only one planar embedding. See also Kelmans [1980])

8.2.46. Construct a graph without isolated vertices that is an abstract dual of the graph below but is not a geometric dual of this graph. (Hint: Consider the operations of Exercise 8.2.45.) (Woodall, in Welsh [1976], p91–92)

8.2.47. The **matroid basis graph** is the graph having a vertex for each base of a matroid, with bases adjacent when their symmetric difference has size 2. Prove that every matroid basis graph has a spanning cycle, and interpret the result for graphic matroids and for uniform matroids. (Hint: Use contraction and restriction inductively to establish a spanning cycle through any edge.) (Holzmann–Harary [1972], Kung [1986, p72])

8.2.48. Use weak duality of linear programming to prove the weak duality property for matroid intersection: $|I| \le r_1(X) + r_2(\overline{X})$ for any $I \in \mathbf{I}_1 \cap \mathbf{I}_2$ and $X \subseteq E$. (Hint: Consider the discussion of dual pairs of linear programs in Remark 8.1.7.)

8.2.49. Let M_1, M_2 be two matroids on E.

a) Prove that the minimum size of a set in E that is spanning in both M_1 and M_2 is $\max_{X \subseteq E}(r_1(E) - r_1(X) + r_2(E) - r_2(\overline{X}))$.

b) Apply part (a) to prove that in a bipartite graph with no isolated vertices the minimum number of edges needed to cover all the vertices equals the maximum number of vertices with no edges among them. (König's "other" theorem)

c) From part (a), prove that the maximum size of a common independent set plus the minimum size of a common spanning set equals $r_1(E)+r_2(E)$. In particular, conclude Gallai's Theorem for bipartite graphs: in a bipartite graph with no isolated vertices, the maximum size of a matching plus the minimum number of edges needed to cover the vertices equals the number of vertices.

8.2.50. Use the Matroid Intersection Theorem to prove that in every acyclic orientation of G the vertices can be covered with at most $\alpha(G)$ pairwise-disjoint paths. (Chappell [1994b]) (Comment: This is the special case of Theorem 8.4.33 for acyclic digraphs.)

8.2.51. (−) Let M be the transversal matroid on $E = \cup A_i$ induced by sets $A_1, \ldots, A_m$. Use Hall's Theorem for matchings in bipartite graphs to derive the rank function as $r(X) = \min_{Y \subseteq X}\{|X| - (|Y| - |N(Y)|)\}$.

8.2.52. Let G be an $E, [m]$-bigraph without isolated vertices. For $X \subseteq E$, let $r(X) = \min\{|N(J) \cap X| - |J| + m \colon J \subseteq [m]\}$. Prove that the following are equivalent for X.

A) Hall's Condition holds ($|N(S)| \ge |S|$ for all $S \subseteq X$).

B) $r(X) \ge |X|$.

C) X is saturated by some matching in G.

(Hint: The proof of B $\Rightarrow$ C uses paths from unsaturated vertices that alternate between edges outside and within a specified matching.)

8.2.53. (!) Let G be an $E, [m]$-bigraph without isolated vertices. For $X \subseteq E$ and $J \subseteq [m]$, let $g(X, J) = |N(J) \cap X| - |J|$, and let $r(X) = \min\{g(X, J) + m \colon J \subseteq [m]\}$. Say that J is *X-optimal* if $r(X) = g(X, J) + m$.

a) Prove that $r(\emptyset) = 0$ and that $r(X) \le r(X + e) \le r(X) + 1$.

b) Prove that r satisfies the weak absorption property.

8.2.54. Prove that restrictions and unions of transversal matroids are transversal matroids, but that contractions and duals of transversal matroids need not be.

8.2.55. *Gammoids.* Let D be a digraph, and let F, E be subsets of $V(D)$. The **gammoid** on E induced by D, F is the hereditary system given by $\mathbf{I} = \{X \subseteq E \colon$ there exist $|X|$ pairwise disjoint paths from F to $X\}$; equivalently, $r(X)$ is the maximum number of pairwise disjoint F, X-paths.

a) Verify that every transversal matroid is a gammoid.

b) (+) Prove that every gammoid is a matroid. (Hint: Use Menger's Theorem to verify the submodularity property. Verifying the augmentation property is also possible but somewhat longer.) (Mason [1972])

8.2.56. *Strict gammoids.* Let D be a directed graph, let F, E be subsets of the vertices of D, and let M be the gammoid on E induced by D, F (Exercise 8.2.55). When E consists of all vertices of D, the gammoid is a **strict gammoid**. Prove that a matroid is a strict gammoid if and only if it is the dual of a transversal matroid. (Hint: Use a natural correspondence between directed graphs on n vertices and bipartite graphs on $2n$ vertices.) (Ingleton–Piff [1973])

8.2.57. (−) Since the union of two matroids is a matroid, there should be a dual operation yielding its dual. Given matroids M_1, M_2 with spanning sets $\mathbf{S}_1, \mathbf{S}_1$, let $M_1 \wedge M_2$ be the hereditary system whose spanning sets are $\{X_1 \cap X_2 \colon X_1 \in \mathbf{S}_1, X_2 \in \mathbf{S}_2\}$. Prove that $M_1 \wedge M_2$ is the matroid $(M_1^* \cup M_2^*)^*$.

8.2.58. *Generalized transversal matroids.*

a) Let M be a matroid on E, and let $\mathbf{A} = \{A_1, \ldots, A_m\}$ be a set system on E. Let M' be the hereditary system on $[m]$ whose independent sets are the subsets of $\mathbf{A}$ having transversals that belong to $\mathbf{I}_M$. Prove that M' is a matroid with rank function $r'(X) = \min_{Y \subseteq X}\{|X - Y| + r(A(Y))\}$.

b) Let E, F be finite sets, and let f be a function from E to F. For $X \subseteq E$, let $f(X)$ be the set of images of elements of X. Let M be a matroid on E. Let M' be the hereditary system on F defined by $\mathbf{I}_{M'} = \{f(X) \colon X \in \mathbf{I}_M\}$. Prove that M' is a matroid. Prove also that $r'(X) = \min_{Y \subseteq X}\{|X - Y| + r(f^{-1}(Y))\}$ when f is surjective.

8.2.59. Apply matroid sum and Exercise 8.2.58 to prove the Matroid Union Theorem.

8.2.60. (!) Prove that the maximum size of a common independent set in matroids M_1 and M_2 on E is $r_{M_1 \cup M_2^*}(E) - r_{M_2^*}(E)$. Use this to prove the Matroid Intersection Theorem by applying the Matroid Union Theorem to $M_1 \cup M_2^*$. (Comment: Thus these two theorems are equivalent.)

8.2.61. Let G be an n-vertex weighted graph, and let $E_1, \ldots, E_{n-1}$ be a partition of $E(G)$ into $n - 1$ sets. Is there a polynomial-time algorithm to compute a spanning tree of minimum weight among those that have exactly one edge in each subset E_i?

8.2.62. (!) Use the characterization of graphs having k pairwise edge-disjoint spanning trees (Corollary 8.2.59) to prove that every $2k$-edge-connected graph has k pairwise edge-disjoint spanning trees. Exhibit for each k a $2k$-edge-connected graph that does not have $k+1$ pairwise edge-disjoint spanning trees. (Nash-Williams [1961])

8.2.63. Given matroids $M_1, \dots, M_k$ on E, the **Matroid Partition Problem** is the problem of deciding whether an input set $X \subseteq E$ partitions into sets $I_1, \dots, I_k$ with $I_i \in \mathbf{I}_i$.

a) Use the Matroid Union Theorem to show that X is partitionable if and only if $|X - Y| + \sum r_i(Y) \geq |X|$ for all $Y \subseteq X$, and that all maximal partitionable sets are maximum partitionable sets.

b) Let M' be the union of k copies of a matroid M on E, and let X be a maximum partitionable set. Prove that there are disjoint sets $F_1, \dots, F_k \subseteq X$ such that $\{F_i\} \subseteq \mathbf{I}$ and $\overline{X} \subseteq \sigma(F_1) = \cdots = \sigma(F_k)$.

8.3. Ramsey Theory

"Ramsey theory" refers to the study of partitions of large structures. Typical results state that a special substructure must occur in some class of the partition. Motzkin described this by saying that "Complete disorder is impossible". The objects we consider are merely sets and numbers, and the techniques are little more than induction.

Ramsey's Theorem generalizes the pigeonhole principle, which itself concerns partitions of sets. We study applications of the pigeonhole principle, prove Ramsey's Theorem, and then focus on Ramsey-type questions for graphs. Finally, we discuss Sperner's Lemma about labelings of triangulations; like Ramsey's Theorem, it guarantees a special substructure.

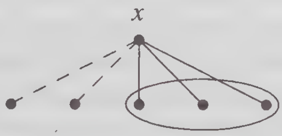
THE PIGEONHOLE PRINCIPLE REVISITED

The pigeonhole principle (Lemma A.57) states that if m objects are partitioned into n classes, then some class has at least $\lceil m/n \rceil$ objects (and some class has at most $\lfloor m/n \rfloor$ objects). This is a discrete version of the statement that every set of numbers contains a number at least as large as the average (and one at least as small). The concept is simple, but the applications can be quite subtle. The difficulty is how to define a partitioning problem relevant to the desired application. We illustrate this with four examples.

8.3.1. Proposition. Among six persons it is possible to find three mutual acquaintances or three mutual non-acquaintances.

Proof: (Exercise 1.1.29). In the language of graph theory, we are asked to show that for every simple graph G with six vertices, there is a triangle in G or in $\overline{G}$. The degrees of vertex x in G and $\overline{G}$ sum to 5, so the pigeonhole principle implies that one of them is at least 3.

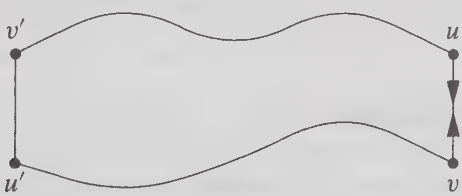
By symmetry, we may that assume $d_G(x) \geq 3$. If two neighbors of x are adjacent, then they form a triangle in G with x; otherwise, three neighbors of x form a triangle in $\overline{G}$. ■



8.3.2. Theorem. (Graham–Entringer–Székely [1994]) If T is a spanning tree of the k-dimensional cube Q_k, then there is an edge of Q_k outside T whose addition to T creates a cycle of length at least $2k$.

Proof: For each vertex v of Q_k, expressed as a binary k-tuple, there is a complementary vertex v' that differs from v in each position. There is a unique v, v'-path in T; orient its first edge toward v'. Since $n(Q_k) = e(T) + 1$, doing this for each vertex orients some edge twice, by the pigeonhole principle.

Since this edge uv receives an orientation from u and from v, we have v on the u, u'-path and u on the v, v'-path in T. Hence the u, v'-path and the v, u'-path in T are disjoint. Each has length at least $k - 1$, since the distance in Q_k between a vertex and its complement is k. Finally, $u \leftrightarrow v$ in Q_k implies also $u' \leftrightarrow v'$, which completes a cycle of length at least $2k$. ■



Theorem 8.3.2 implies that every spanning tree of Q_k has diameter at least $2k - 1$ (Graham–Harary [1992]).

8.3.3. Theorem. (Erdős–Szekeres [1935]) Every list of more than n^2 distinct numbers has a monotone sublist of length more than n.

Proof: Let $a = a_1, \ldots, a_{n^2+1}$ be the list. Assign position k the label (x_k, y_k), where x_k is the length of a longest increasing sublist ending at a_k, and y_k is the length of a longest decreasing sublist ending at a_k. If a has no monotone sublist of length $n+1$, then x_k and y_k never exceed n, and there are only n^2 possible labels.

Since the list has length $n^2 + 1$, the pigeonhole principle now implies that two labels must be the same. This is impossible when the elements of a are distinct. When $i < j$ and $a_i < a_j$, we can append a_j to the longest increasing sequence ending at a_i. When $i < j$ and $a_i > a_j$, we can append a_j to the longest decreasing sequence ending at a_i. (See Exercise 5.1.43 for a generalization.) ■

a:	7	4	1	8	5	2	9	6	3	0
x, y:	1, 1	1, 2	1, 3	2, 1	2, 2	2, 3	3, 1	3, 2	3, 3	4, 1

8.3.4. Theorem. (Graham–Kleitman [1973]) In every labeling of $E(K_n)$ using distinct integers, there is a trail of length at least $n-1$ along which the labels strictly increase.

Proof: We assign each vertex a weight equal to the length of the longest increasing trail ending there. If we can show that these n weights sum to at least $n(n-1)$, then the pigeonhole principle guarantees a vertex with a large enough weight. The problem is how to compute the weights and their sum.

We grow the graph from the trivial graph by adding the edges in order, updating the weights and their sum at each step. The vertex weights begin at 0. If the next edge joins two vertices whose weights were both i, then their weights both become $i+1$. If it joins two vertices of weights i and j with $i<j$, then their weights become $j+1$ and j.

In either case, each time an edge is added, the sum of the weights of the vertices increases by at least 2. Therefore, when the construction is finished, the sum of the vertex weights is at least $n(n-1)$. ■

Finally, we note that the thresholds in the classes may differ.

8.3.5. Theorem. If $\sum p_i - k + 1$ objects are partitioned into k classes with quotas $\{p_i\}$, then some class must meet its quota.

Proof: If not, then at most $\sum(p_i - 1)$ objects can be accommodated. ■

RAMSEY'S THEOREM

The pigeonhole principle guarantees a class with many objects when we partition objects into classes. The famous theorem of Ramsey [1930] makes a similar statement about partitioning the r-element subsets of objects into classes. Roughly put, Ramsey's Theorem says that whenever we partition the r-sets in a sufficiently large set S into k classes, there is a p-subset of S whose r-sets all lie in the same class.

A partition is a separation of a set into subsets, and the set we want to partition consists of subsets of another set, so for clarity we use the language of coloring instead of the language of partitioning. Recall that a k-**coloring** of a set is a partition of it into k classes. A class or its label is a **color**. Typically we use $[k]$ as the set of colors, in which case a k-coloring of X can be viewed as a function $f: X \to [k]$.

8.3.6. Definition. Let $\binom{S}{r}$ denote the set of r-element subsets (r-**sets**) of a set S. A set $T \subseteq S$ is **homogeneous** under a coloring of $\binom{S}{r}$ if all r-sets in T receive the same color; it is i-**homogeneous** if that color is i.

Let r and $p_1, \ldots, p_k$ be positive integers. If there is an integer N such that every k-coloring of $\binom{[N]}{r}$ yields an i-homogeneous set of size p_i for some i, then the smallest such integer is the **Ramsey number** $R(p_1, \ldots, p_k; r)$.

Ramsey's Theorem states that such an integer exists for every choice of r and $p_1, \dots, p_k$ (the latter are called **thresholds** or **quotas**). When the quotas all equal p, the theorem states that every k-coloring of the r-sets of a sufficiently large set has a p-set whose r-sets receive the same color. A thorough study of Ramsey's Theorem and other partitioning theorems appears in Graham–Rothschild–Spencer [1980, 1990].

Before proving the theorem, we consider the case $r = k = 2$, which is easy to describe in terms of edge-coloring of graphs. The proof for this case has the same structure as for the general case.

When $r = 2$, a k-partition of $\binom{S}{r}$ is merely a k-edge-coloring of the complete graph with vertex set S (not a proper edge-coloring). When $k = 2$, the time-honored tradition in Ramsey theory is that color 1 is "red" and color 2 is "blue".

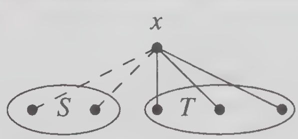
By Proposition 8.3.1, $R(3, 3; 2) \le 6$; we extend the argument to prove that

$$R(p_1, p_2; 2) \le R(p_1 - 1, p_2; 2) + R(p_1, p_2 - 1; 2).$$

Assuming that $R(p_1 - 1, p_2; 2)$ and $R(p_1, p_2 - 1; 2)$ exist, let N be their sum. Proving the bound for $R(p_1, p_2; 2)$ means showing that every red/blue-coloring of the edges of a complete graph with N vertices yields a p_1-set of vertices within which all edges are red or a p_2-set of vertices within which all edges are blue.

Consider a red/blue-coloring of K_N, and choose a vertex x. Let $s = R(p_1 - 1, p_2; 2)$ and $t = R(p_1, p_2 - 1; 2)$; there are $s + t - 1$ vertices other than x. Theorem 8.3.5 implies that x has at least s incident red edges or at least t incident blue edges.

By symmetry, we may assume that x has at least N incident red edges. By the definition of s, the complete subgraph induced by the neighbors of x along these edges has a blue p_2-clique or a red $p_1 - 1$-clique. The latter would combine with x to form a red p_1-clique. In either case, we obtain an i-homogeneous set of size p_i for some i. We postpone discussion of the resulting bound on $R(p_1, p_2; 2)$.



$|S| \ge R(p_1, p_2 - 1; 2)$ or $|T| \ge R(p_1 - 1, p_2; 2)$

8.3.7. Theorem. (Ramsey [1930]) Given positive integers r and $p_1, \dots, p_k$, there exists an integer N such that every k-coloring of $\binom{[N]}{r}$ yields an i-homogeneous set of size p_i for some i.

Proof: The proof is a "double" induction. We use induction on r, but the proof of the induction step itself uses induction on $\sum p_i$.

Basis step: $r = 1$. By Theorem 8.3.5, $R(p_1, \dots, p_k; 1)$ exists.

Induction step: $r > 1$. We assume that the claim in the theorem statement holds for k-colorings of the $r - 1$-subsets of a set, no matter what the thresholds are. We prove the same statement for k-colorings of the r-subsets of a set by induction on the sum of the quotas, $\sum p_i$.

Basis step: some quota p_i is less than r. In this case, a set of p_i objects contains no r-sets, so vacuously its r-sets all have color i. Hence $R(p_1, \ldots, p_k; r) = \min\{p_1, \ldots, p_k\}$ when $\min\{p_1, \ldots, p_k\} < r$.

For clarity, we state the induction step only for $k = 2$; the argument for general k is similar (Exercise 17). Write (p, q) for (p_1, p_2). Let

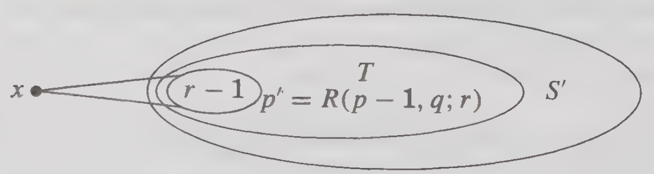
$$p' = R(p-1, q; r), \quad q' = R(p, q-1; r), \quad N = 1 + R(p', q'; r-1).$$

By the induction hypothesis of the inner induction, p' and q' exist. By the induction hypothesis of the outer induction, N also exists. Note that p' and q' may be very large; this is why we need the double induction.

Let S be a set of N elements, and choose $x \in S$. Consider a 2-coloring f of $\binom{S}{r}$. With colors (red,blue), we need to show that f has a red-homogeneous p-set or a blue-homogeneous q-set.

We use f to induce a 2-coloring f' of the $r-1$-sets of $S' = S - x$. This is the reason for our choice of $|S'|$ as a Ramsey number for $r-1$-sets. Define f' by assigning color i to an $(r-1)$-set in S' if its union with x has color i under f. Since $|S'| = R(p', q'; r-1)$, the induction hypothesis implies that some color meets its quota (p' or q') under f' (when $r = 2$, this step was the invocation of the pigeonhole principle). By symmetry, we may assume that the red quota is met. Let T be a p'-element subset of S' whose $r-1$-sets are red under f'.

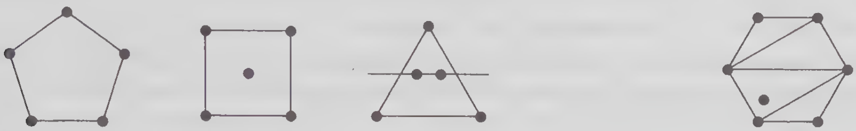
We return to the original coloring f on the r-sets in T. Since $|T| = p' = R(p-1, q; r)$, under f there is a red-homogeneous $p-1$-set or a blue-homogeneous q-set in T. If there is a blue-homogeneous q-set, then we are done. If there is a red-homogeneous $p-1$-set P, then consider $P \cup \{x\}$. From the definition of T, the $(r-1)$-sets of P are all red under f', which means their unions with x are red under f. Hence $P \cup \{x\}$ is a red-homogeneous p-set under f. ∎



Like the pigeonhole principle, Ramsey's Theorem has subtle and fascinating applications. Ramsey's Theorem typically gives an elegant existence proof but a horribly large bound.

8.3.8. Theorem. (Erdős–Szekeres [1935]) Given an integer m, there exists a (least) integer $N(m)$ such that every set of at least $N(m)$ points in the plane with no three collinear contains an m-subset forming a convex m-gon.

Proof: We need two facts. *(1) Among five points in the plane, four determine a convex quadrilateral* (if no three are collinear). Construct the convex hull of the five points. If it is a pentagon or a quadrilateral, then the result follows immediately. If it is a triangle, then the other two points lie inside. By the pigeonhole principle(!), two of the vertices of the triangle are on one side of the line through the two inside points. These two vertices together with the two points inside form a convex quadrilateral, as illustrated below.



In a convex m-gon, any four corners determine a convex quadrilateral. We need the converse: *(2) If every 4-subset of m points in the plane forms a convex quadrilateral, then the m points form a convex m-gon.* If the claim fails, then the convex hull of the m points consists of t points, for some $t < m$. The remaining points lie inside the t-gon. When we triangulate the t-gon, as illustrated on the right above, a point inside lies in one of the triangles. With the vertices of that triangle, it forms a 4-set that does not determine a convex quadrilateral.

To prove the theorem, let $N = R(m, 5; 4)$. Given N points in a plane with no three on a line, color each 4-set by convexity: red if it determines a convex quadrilateral, blue if it does not. By fact (1), there cannot be five points whose 4-subsets are all blue. By Ramsey's Theorem, this means there are m points whose 4-subsets are all red. By fact (2), they form a convex m-gon. Hence $N(m)$ exists and is at most $R(m, 5; 4)$. ∎

The bound $R(m, 5; 4)$ is very loose. It is exact for $m = 4$, where fact (1) implies that $N(4) = 5 = R(4, 5; 4)$. In contrast, $N(5) = 9$ (Exercise 10), but $R(5, 5; 4)$ is enormous. Erdős and Szekeres conjectured that $N(m) = 2^{m-2} + 1$ and proved that $2^{m-2} \le N(m) \le \binom{2m-4}{m-2} + 1$.

Another application concerns search strategies for numbers stored in tables. From a set U, a subset of size n is stored in a table of size n according to some rule for storing n-sets. Yao [1981] used Ramsey's Theorem to prove that when U is large, the strategy minimizing the worst-case number of probes required to test whether some element of U is in the table is to store the chosen set in sorted order and test membership by binary search. (For small U, this strategy is not best!) The value that Ramsey's Theorem yields for "large" is probably much larger than needed.

RAMSEY NUMBERS

Ramsey's Theorem defines the Ramsey numbers $R(p_1, \dots, p_k; r)$. No exact formula is known, and few Ramsey numbers have been computed. To prove that $R(p_1, \dots, p_k; r) = N$, we must exhibit a k-coloring of the r-sets among $N-1$ points that meets no quota (or show that one exists without constructing it), *and* we must show that *every* coloring on N points meets some quota.

In principle, we could use a computer to examine all k-colorings of $\binom{[n]}{r}$ for successive n until we find the first N such that every such coloring meets a quota p_i for some i. Even for 2-color Ramsey numbers, $2^{\binom{n}{2}}$ rapidly becomes too large to contemplate. Erdős joked that if an alien being threatened to destroy us unless we told it the exact value of $R(5, 5)$, then we should set all the computers in the world to work on an exhaustive solution. If we were asked for $R(6, 6)$, then his advice was to try to destroy the alien.

When $r = 2$, we abbreviate the notation $R(p_1, \ldots, p_k; r)$ to $R(p_1, \ldots, p_k)$. When $p = p_1 = \cdots = p_k$, we abbreviate it to $R_k(p; r)$. For $r > 2$, little is known other than $R(4, 4; 3) = 13$ (McKay–Radziszowski [1991]). Even for $r = 2$, only one Ramsey number is known exactly when $k > 2$, which is $R(3, 3, 3) = 17$. The table below contains the known values of $R(p, q)$ and the best known upper and lower bounds for several other values as of July 1999. Several of these bounds have improved slightly since the first edition of this book. The current bounds are maintained in Radziszowski [1995], which is periodically updated.

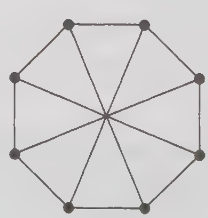
	3	4	5	6	7	8	9
3	6	9	14	18	23	28	36
4		18	25	35/41	49/61	55/84	69/115
5			43/49	58/87	80/143	95/216	116/316
6				102/165	109/298	122/495	153/780

The computations of $R(3, 9)$ (Grinstead–Roberts [1982]), $R(3, 8)$ (McKay–Zhang [1992]), and $R(4, 5)$ (McKay–Radziszowski [1995]) are recent; the others are much older (due primarily to Greenwood–Gleason [1955], Kalbfleisch [1967], and Graver–Yackel [1968]).

We prove only the first two of these results (see Exercise 16 for $R(3, 5)$). When $r = k = 2$, we simplify terminology by using two colors called "in" and "out". Ramsey's Theorem for this case then becomes: "There exists a minimum integer $R(p, q)$ such that every graph on $R(p, q)$ vertices has a clique of size p or an independent set of size q".

8.3.9. Example. $R(3, 3) = 6$. We showed earlier that $R(3, 3) \leq 6$. Since the 5-cycle has no triangle and no independent 3-set, $R(3, 3) \geq 6$. ■

8.3.10. Example. $R(3, 4) = 9$. The graph below has no K_3 and no $\overline{K}_4$, since four independent vertices on an 8-cycle include pairs of opposite vertices on the cycle. Hence $R(3, 4) \geq 9$.



Given a vertex x in a graph G, we can add x to two adjacent neighbors to form a triangle or add x to an independent 3-set of nonneighbors to form an independent 4-set. Since $R(2, 4) = 4$ and $R(3, 3) = 6$, we conclude that if x has four neighbors or has six nonneighbors, then G has a triangle or an independent 4-set. Avoiding both possibilities limits x to at most three neighbors and at most five nonneighbors, which yields $n(G) \leq 9$. If this occurs for a 9-vertex graph, then *every* vertex has exactly three neighbors. Since the degree-sum formula forbids 3-regular graphs of order 9, we obtain $R(3, 4) = 9$. ■

The proof of Ramsey's Theorem yields a (very large) recursive upper bound on $R(p, q; r)$. Graham–Rothschild–Spencer [1980, 1990] explains how large.

8.3.11. Theorem. $R(p, q) \leq R(p-1, q) + R(p, q-1)$. If both summands on the right are even, then the inequality is strict.

Proof: If a vertex in an arbitrary graph has $R(p-1, q)$ neighbors or $R(p, q-1)$ nonneighbors, then the graph has a p-clique or an independent q-set. With $R(p-1, q) + R(p, q-1)$ points altogether in the graph, the pigeonhole principle guarantees that one of these possibilities occurs. Equality in the bound requires a regular graph with $R(p-1, q) + R(p, q-1) - 1$ vertices. If both summands are even, this requires a regular graph of odd degree on an odd number of vertices, which is impossible. ■

Since $R(p, 2) = R(2, p) = p$, Theorem 8.3.11 yields $R(p, q) \leq \binom{p+q-2}{p-1}$ (Exercise 15). The lack of exact answers has led to study of asymptotics. For fixed q and large p, $R(p, q) \leq cp^{q-1} \log\log p / \log p$ (Graver–Yackel [1968], Chung–Grinstead [1983]). For $q = 3$, the answer is known within a constant factor:

$$c' p^2 / \log p \leq R(p, 3) \leq cp^2 / \log p.$$

The upper bound is due to Ajtai–Komlós–Szemerédi [1980]; the lower to Kim [1995]. All these bounds use probabilistic methods (Section 8.5).

Ramsey numbers for equal quotas are called **diagonal Ramsey numbers**. Asymptotically, the upper bound of $\binom{2p-2}{p-1}$ for $R(p, p)$ is $c4^p/\sqrt{p}$. Exercise 14 presents a constructive lower bound that is polynomial in p. The best known constructive lower bound grows faster than every polynomial in p but slower than every exponential in p (Frankl–Wilson [1981], Exercise 29).

An exponential lower bound can be proved by counting methods. It yields

$$\sqrt{2} \leq \liminf R(p, p)^{1/p} \leq \limsup R(p, p)^{1/p} \leq 4.$$

Determination of this limit (and whether it exists) is the foremost open problem about Ramsey numbers.

8.3.12. Theorem. (Erdős [1947]). $R(p, p) > (e\sqrt{2})^{-1} p 2^{p/2}(1 + o(1))$.

Proof: Consider the graphs with vertex set $[n]$. Each possible p-clique occurs in $2^{\binom{n}{2}-\binom{p}{2}}$ of these $2^{\binom{n}{2}}$ graphs. Similarly, each p-set occurs as an independent set in $2^{\binom{n}{2}-\binom{p}{2}}$ of these graphs. Discarding this amount for each possible p-clique and each possible independent p-set leaves a lower bound on the number of graphs having no p-clique or independent p-set.

Since there are $\binom{n}{p}$ ways to choose p vertices, the inequality $2\binom{n}{p}2^{-\binom{p}{2}} < 1$ thus implies $R(p, p) > n$. Rough approximations yield $\binom{n}{p}2^{1-\binom{p}{2}} < 1$ whenever $n < 2^{p/2}$. More careful approximations (using Stirling's formula to approximate the factorials) lead to the result claimed. ■

GRAPH RAMSEY THEORY

Ramsey's Theorem for $r = 2$ says that k-coloring the edges of a large enough complete graph forces a monochromatic complete subgraph. A monochromatic p-clique contains a monochromatic copy of every p-vertex graph. Perhaps monochromatic copies of graphs with fewer edges can be forced by coloring a smaller graph than needed to force K_p. For example, 2-coloring the edges of K_3 always yields a monochromatic P_3, although six points are needed to force a monochromatic triangle. This suggests many Ramsey number questions, some easier to answer than the questions for cliques.

8.3.13. Definition. Given simple graphs $G_1, \ldots, G_k$, the **(graph) Ramsey number** $R(G_1, \ldots, G_k)$ is the smallest integer n such that every k-coloring of $E(K_n)$ contains a copy of G_i in color i for some i. When $G_i = G$ for all i, we write $R_k(G)$ for $R(G_1, \ldots, G_k)$.

Burr [1983] determined $R(G, G)$, called the "Ramsey number of G", for all 113 graphs with at most six edges and no isolated vertices. Nice formulas are known for $R(G_1, G_2)$ in some cases. Again our two colors are red and blue.

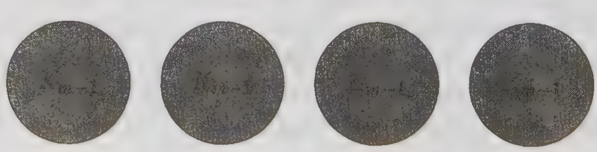
8.3.14. Theorem. (Chvátal [1977]) If T is an m-vertex tree, then $R(T, K_n) = (m-1)(n-1)+1$.

Proof: For the lower bound, color $K_{(m-1)(n-1)}$ by letting the red graph be $(n-1)K_{m-1}$. With red components of order $m-1$, there is no red m-vertex tree. The blue edges form an $n-1$-partite graph and hence cannot contain K_n.

The proof of the upper bound uses induction on each parameter, focusing on the neighbors of one vertex. Our presentation uses induction on n, invoking a property of trees that we proved in Chapter 2 by induction on m. The basis step is $n = 1$; no edges are needed to obtain K_1.

Given a 2-coloring of $E(K_{(m-1)(n-1)+1})$, consider a vertex x. If x has more than $(m-1)(n-2)$ neighbors along blue edges, then the induction hypothesis yields a red T or a blue K_{n-1} among them. This yields a red T or a blue K_n (with x) in the full coloring.

Otherwise, every vertex has at most $(m-1)(n-2)$ incident blue edges and thus at least $m-1$ incident red edges. This yields a red T, because every graph with minimum degree at least $m-1$ contains T (Proposition 2.1.8). ■



Whenever the largest component of G has m vertices and $\chi(H) = n$, the construction in Theorem 8.3.14 yields $R(G, H) \geq (m-1)(n-1)+1$ (Chvátal–Harary [1972]). Burr and Erdős [1983] conjectured that $R(G, K_n) = (m-1)(n-1)+1$ when m is sufficiently large relative to $n(H)$ and $\max_{F \subseteq G} \frac{e(F)}{n(F)}$. Although this holds (Burr [1981]) when G has many vertices of degree 2 and in some other cases, Brandt [2000] showed that for every nonbipartite graph H (such as K_n) and every $h \in \mathbb{R}$, there is a threshold d_0 such that $R(G, H) > hn(G)$ for almost every d-regular graph G with $d > d_0$.

In the upper bound for Theorem 8.3.14, it is crucial that the color classes in H are single vertices. When this fails, the lower bound can be very weak. When $G = H = mK_3$, for example, the Chvátal–Harary result yields $R(G, H) \geq (3-1)(3-1)+1 = 5$, but the correct value is $5m$. Here the coloring for the lower bound is surprisingly asymmetric, considering the symmetry of the inputs.

8.3.15. Theorem. (Burr–Erdős–Spencer [1975]) $R(mK_3, mK_3) = 5m$ for $m \geq 3$.

Proof: Let the red graph be $K_{3m-1} + K_{1,2m-1}$, as shown below. Every triangle in this graph uses three vertices from the $3m-1$-clique, but the clique does not have enough vertices to make m disjoint triangles. The complementary blue graph is $(K_{2m-1} + K_1) \vee \overline{K}_{3m-1}$. Every blue triangle has at least 2 vertices in the copy of K_{2m-1}, so there cannot be m disjoint blue triangles.

For the upper bound, we use induction on m. Basis step: $m = 2$. This requires a case analysis that is fairly short if phrased carefully (Exercise 26).

Induction step: $m \geq 3$. Since $5m > R(3,3) = 6$, we know that every 2-coloring contains a monochromatic triangle. Discarding vertices of triangles as we find them, we can continue to find monochromatic triangles while at least six vertices remain. Since $5m - 3m \geq 6$ for $m \geq 3$, we find m disjoint monochromatic triangles. If these all have the same color, then we are done.

Otherwise, we have at least one triangle in each color. Let abc be a red triangle, and let def be a blue triangle disjoint from it. Of the nine edges between them, we may assume by symmetry that at least five are red. Some pair of these must have a common endpoint in def.

Now we have a red triangle and a blue triangle with a common vertex; together they have five vertices. Since $m > 2$, the induction hypothesis for the coloring on the remaining $5m-5$ vertices yields $(m-1)K_3$ in one color. We add the appropriately colored triangle from the five special vertices. ■

Readers worried about the omission of the basis step in Theorem 8.3.15 may consider coloring K_{11}. Avoiding $2K_3$ forces a bowtie (monochromatic triangles with a common vertex) as above, but then we find another monochromatic triangle among the remaining six points. This completes a proof that $R(mK_3, mK_3) \leq 5m+1$. Related results appear in Exercises 27–28.

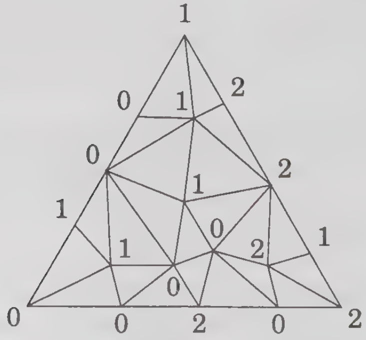
We mention one remarkable result. The Ramsey number of an arbitrary graph may be exponential in the number of vertices, such as for K_n. Chvátal, Rödl, Szemerédi, and Trotter [1983] proved that for the class of graphs with maximum degree d, the Ramsey number grows at most linearly in the number of vertices! In other words, $R(G, G) \le cn(G)$, where c is a constant depending only on d. Of course, the constant is a fast-growing function of d, but it does not depend on $n(G)$. The proof uses the Szemerédi Regularity Lemma [1978], itself a difficult result with many applications.

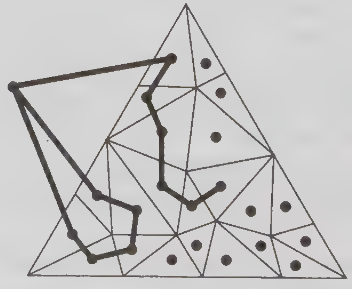
SPERNER'S LEMMA AND BANDWIDTH

Although Sperner's Lemma is not generally considered part of Ramsey theory, we include this material in this section because Sperner's Lemma has the flavor of Ramsey theory: every labeling of a triangulation that satisfies certain boundary conditions contains a piece with a special labeling (one element from *each* class). Like Ramsey's Theorem, Sperner's Lemma uses very simple ideas but has subtle applications; Ramsey's Theorem relies on the pigeonhole principle and induction, while Sperner's Lemma uses only a parity argument (and induction for a generalization to higher dimensions).

8.3.16. Definition. A **simplicial subdivision** of a large triangle T is a partition of T into triangular **cells** such that every intersection of two cells is a common edge or corner. We call the corners of cells **nodes**. A **proper labeling** of a simplicial subdivison of T assigns labels from $\{0, 1, 2\}$ to the nodes, avoiding label i on the ith edge of T, for $i \in \{0, 1, 2\}$. A **completely labeled cell** is a cell having all three labels on its corners.

In a proper labeling, each label appears at one corner of T, and label i avoids the edge of T joining the corners not labeled i. The figure below illustrates a simplicial subdivision and the graph we will obtain from it to prove that it has a completely labeled cell.





8.3.17. Theorem. (Sperner's Lemma [1928]) Every properly labeled simplicial subdivision has a completely labeled cell.

Proof: We prove the stronger result that there are an odd number of completely labeled cells. We seek such a cell by beginning outside T and entering a cell by crossing an edge with labels 0 and 1. If we reach a cell whose third label is 2, we are finished. If not, then the third label is 0 or 1, and the cell has another 0,1-edge. By crossing it, we enter a new cell and can continue looking for a cell with the third label.

This suggests defining a graph G encoding the possible steps. We include a vertex for each cell plus one vertex for the outside region. Two vertices of G are adjacent if those regions share a boundary edge with endpoints labeled 0 and 1. The graph on the right above results from the proper labeling on the left.

A completely labeled cell becomes a vertex of degree 1 in G. A cell with no 0 or no 1 becomes a vertex of degree 0. The remaining cells have corners labeled 0, 0, 1 or 0, 1, 1 and become vertices of degree 2. Hence the desired cells become vertices of degree 1 in G, and these are the only cells that become vertices of odd degree. We have transformed the original problem into the problem of showing that G has such a vertex of degree 1.

The vertex v for the outside region also has odd degree. As we travel from the 0-corner to the 1-corner along the edge of T that avoids label 2, we cross an edge of G involving v every time we switch from a 0 to a 1 or back again. Since we start with 0 and end with 1, we switch an odd number of times. Hence v has odd degree. Since the number of vertices of odd degree in every graph is even, the number of vertices other than v having odd degree is odd, so there are an odd number of completely labeled cells. ■

8.3.18. Application. *Brouwer Fixed-Point Theorem.* Brouwer's Theorem (for two dimensions) can be interpreted as saying that a continuous mapping from a triangular region T to itself must have a fixed point. Suppose that the corners of T are the points (vectors) v_0, v_1, v_2. Just as we can express a point on a segment uniquely as a weighted average of its endpoints, so we can express each $v \in T$ uniquely as a weighted average of the corners: $v = a_0v_0 + a_1v_1 + a_2v_2$, where $\sum a_i = 1$ and each $a_i \geq 0$ (Exercise 37). We can specify v by its vector of coefficients $a = (a_0, a_1, a_2)$.

Define sets S_0, S_1, S_2 from the mapping f by placing $a \in S_i$ if $a'_i \leq a_i$, where $f(a) = a'$. Since the coefficients of each point sum to 1, every point in T belongs to some S_i, and a point belongs to all three sets if and only if it is a fixed point for f. We want to show that the three sets have a common point.

Given a simplicial subdivision of T, for each node a choose a label i such that $a \in S_i$. Points on the edge of T opposite v_i have ith coordinate 0. Their ith coordinate cannot decrease under f, so we can choose a label other than i for each point on that edge. The resulting labeling is proper, and Sperner's Lemma guarantees a completely labeled cell. Repeating the process using triangulations with successively smaller cells yields a sequence of successively smaller completely labeled triangles. Let the jth triangle have corners x_j, y_j, z_j with labels 0,1,2, respectively. In each S_i, we obtain an infinite sequence of points.

The remaining details are topological; we merely suggest the steps. Since f is continuous, each S_i is closed and bounded. Every infinite sequence of points in a closed and bounded set has a convergent subsequence. Hence $\{x_1, x_2, \ldots\}$ has a convergent subsequence; let x_{i_k} be its kth entry. Because the distance from x_{i_k} to y_{i_k} and z_{i_k} approaches 0, these subsequences also converge to the same point. Since S_0, S_1, S_2 are closed and bounded, this limit point belongs to all three of them and is a fixed point of f. ■

We also apply Sperner's Lemma to solve a problem on the "triangular grid".

8.3.19. Definition. When the vertices of G are numbered with distinct integers, the **dilation** is the maximum difference between integers assigned to adjacent vertices. The **bandwidth** $B(G)$ of a graph G is the minimum dilation of a numbering of G.

Dilation is always minimized when there are no gaps in the numbering, but it can be convenient to allow gaps (Exercise 42). The name "bandwidth" comes from matrix theory; the optimal numbering describes a permutation of the rows and columns of the adjacency matrix so that the 1's appear only in diagonal bands close to the main diagonal; arranging the matrix in this order can speed up computation of the inverse. Another motivation is to minimize the delay between adjacent vertices when the vertices must be processed in a linear order. Computation of bandwidth is NP-hard even for trees with maximum degree 3 (Garey–Graham–Johnson–Knuth [1978]).

We present two fundamental lower bounds on bandwidth.

8.3.20. Lemma. $B(G) \geq \max_{H \subseteq G} \frac{n(H)-1}{\operatorname{diam} H}$.

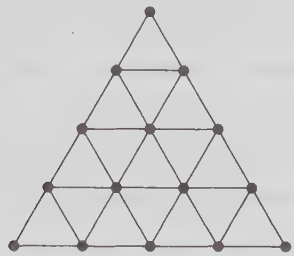
Proof: Every numbering of G contains a numbering of each subgraph of G. On every subgraph H, two numbers differing by at least $n(H) - 1$ are used. By the pigeonhole principle, some edge on a path between the two corresponding vertices has dilation at least $n(H) - 1$ divided by the distance between them. ■

8.3.21. Lemma. (Harper [1966]) $B(G) \geq \max_k \min\{|\partial S| \colon |S| = k\}$, where ∂S denotes the subset of vertices in a set $S \subseteq V(G)$ that have at least one neighbor outside S.

Proof: For every value of k, some set S of k vertices must be the first k vertices in the optimal numbering of G. The bandwidth of G must be at least $|\partial S|$, because the vertex among ∂S that has the least label has an edge of dilation at least $|\partial S|$ to its neighborhood above S. ■

Chung [1988] named the first bound the **local density** bound. The computation of Harper's bound is usually difficult. For the cube Q_k, the value is $\sum_{i=0}^{n-1} \binom{i}{\lfloor i/2 \rfloor}$. For the grid $P_m \square P_n$, the value of Harper's lower bound is $\min\{m, n\}$, which can be achieved (Exercise 43).

8.3.22. Example. *The triangular grid.* The triangular grid T_l consists of vertices (i, j, k) such that i, j, k are nonnegative integers summing to l, with two vertices adjacent if the total of the absolute differences in corresponding coordinates is 2. Below we show T_4. Numbering the vertices by rows produces an upper bound of $l+1$ for $B(T_l)$. This is optimal, but the local density bound is only about $l/2$, and Harper's bound is about $l/\sqrt{2}$. Sperner's Lemma can be used to prove that $l+1$ is optimal. ■



Let G be the graph formed by a simplicial subdivision of a triangle. The outer boundary of G is a cycle, the bounded regions are triangles, and the cycle is partitioned into three paths by the corners of the large triangle. We say that a **connector** is a vertex set inducing a connected subgraph that contains a vertex of each boundary path.

8.3.23. Lemma. (Hochberg–McDiarmid–Saks [1995]) Let T be a simplicial subdivision in which each vertex is assigned red or blue. Let R, B be the subgraphs induced by the red and by the blue vertices, respectively. For each such coloring, exactly one of R, B contains a connector.

Proof: For each vertex v, consider the vertices reachable from v using vertices with the same color as v. If the three sides are not all reachable, label v with the smallest index of a side not reachable from v. For the vertices on the ith side, the label i does not appear. If there is no connector, then each node has a label, and this is a proper labeling of T.

By Sperner's Lemma, there is a completely labeled cell. Since the cell has three corners and we only used two colors R, B, two of the corners of this cell have the same color. Since they are adjacent, they can reach the same set of vertices in their color. Hence the least side unreachable from them cannot be different. This contradiction means that we could not have constructed the specified labeling. Hence there is a vertex from which every side is reachable.

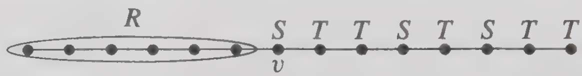
If one color has a connector, it partitions the remaining vertices into sets such that from each set at least one side is unreachable. Hence there cannot be connectors in both colors. ■

8.3.24. Theorem. (Hochberg–McDiarmid–Saks [1995]) Let G be a graph that triangulates a region bounded by a cycle C partitioned into three paths. If k is the minimum over $v \in V(G)$ of the sum of the distances from v to each of the three paths, then $B(G) \geq k+1$.

Proof: Let f be a numbering of G. Let t be the maximum index such that the subgraph induced by the vertices numbered $1, \ldots, t$ does not have a component meeting all three paths. Let R be this vertex set, let S be the set of vertices outside R having neighbors in R, and let T be the remaining vertices.

By construction, the vertex v with $f(v) = t + 1$ belongs to S. Since $R \cup \{v\}$ contains a connector, $R \cup S$ contains a connector, and T does not. Since there is no edge between R and T and R contains no connector, $R \cup T$ contains no connector. Now Lemma 8.3.23 implies that S contains a connector. The set S equals $\partial(S \cup T)$ for the terminal segment $S \cup T$ in the numbering. Therefore, the numbering has difference at least $|S|$ on some edge from S to R.

A connector contains walks from each of its vertices to each of the three boundary paths. By hypothesis, the sum of the lengths of these walks from any fixed vertex is at least k. There exists a vertex in S for which these walks in S are disjoint paths. Hence $|S| \geq k + 1$. ■



8.3.25. Corollary. The triangular grid T_l has bandwidth $l + 1$.

Proof: For each vertex (i, j, k) in T_l, the distances to the three sides are i, j, k, respectively, so the sum of the distances is l. By Theorem 8.3.24, the bandwidth is at least $l + 1$, which we have observed is achievable. ■

EXERCISES

8.3.1. (−) Each of two concentric discs has 20 radial sections of equal size. For each disc, 10 sections are painted red and 10 blue, in some arrangement. Prove that the two discs can be aligned so that at least 10 sections on the inner disc match colors with the corresponding sections on the outer disc.

8.3.2. For $n \in \mathbb{N}$, let S be a set of $n + 1$ elements in $\{1, \ldots, 2n\}$. Prove that S has two elements with greatest common factor 1 and has two elements such that one divides the other. For each conclusion, exhibit a subset of size n where it does not hold; hence these conclusions are best possible.

8.3.3. Use partial sums and the pigeonhole principle to prove the following statements.

a) Every set of n integers contains a nonempty subset whose sum is divisible by n. (Also exhibit a collection of $n - 1$ integers with no such subset.)

b) Given $x \in \mathbb{R}$, prove that at least one of $\{x, 2x, \ldots, (n-1)x\}$ differs by at most $1/n$ from an integer.

8.3.4. (!) A private club has 90 rooms and 100 members. Keys are given to the members so that any 90 members have access to the rooms in the sense that each of these 90 members will have a key to a different room. (They do not share their keys.) Prove that at least 990 keys are needed and that 990 suffice.

8.3.5. Let T be a tree. Use the technique of Theorem 8.3.2 to prove that the center of T consists of one vertex or two adjacent vertices (this proves Theorem 2.1.13 again). (Jordan [1869], Graham–Entringer–Székely [1994])

8.3.6. Prove that every set of $2^m + 1$ integer lattice points in $\mathbb{R}^m$ contains a pair of points whose centroid (mean vector) is also an integer lattice point.

8.3.7. Prove that every 2-coloring of the integer lattice points in $\mathbb{R}^m$ has a collection of n points with the same color whose centroid (mean vector) is an integer lattice point also having that color. (Hint: Ramsey's Theorem is not needed; there is a short proof using only the pigeonhole principle.) (Bòna [1990])

8.3.8. Let S be a collection of $n+1$ positive integers summing to k. For $k \le 2n+1$, prove that S has a subset with sum i for each $i \in [k]$. For each n, exhibit a collection for which this fails when $k = 2n+2$.

8.3.9. For even n, construct an ordering of $E(K_n)$ so that the maximum length of an increasing trail is $n-1$. (Comment: This proves that Theorem 8.3.4 is best possible when n is even. It also is best possible when n is odd and at least 9, but the construction is much more difficult.) (Graham–Kleitman [1973])

8.3.10. Let S be a set of nine points in the plane (no three collinear). Prove that S contains the vertex set of a convex 5-gon. Exhibit a set of eight points without this property.

8.3.11. (!) Let S be a set of $R(m, m; 3)$ points in the plane no three of which are collinear. Prove that S contains m points that form a convex m-gon. (Tarsi)

8.3.12. Recall that a digraph is *simple* if no two edges have the same ordered pair of endpoints. A **monotone tournament** is a tournament in which the orientation of the edges always agrees with the order of the indices on the vertices or always disagrees with that order. A **complete loopless digraph** has one copy of each ordered pair of distinct vertices as an edge. Given m, prove that if N is sufficiently large, then every simple loopless digraph with vertex set $[N]$ has an independent set of order m or a monotone tournament of order m or a complete loopless digraph of order m.

8.3.13. (!) *Schur's Theorem.* (Schur [1916])

a) Given $k > 0$, prove that there exists a least integer s_k such that every k-coloring of the integers $1, \dots, s_k$ yields a monochromatic x, y, z (not necessarily distinct) such that $x + y = z$. (Hint: Apply Ramsey's Theorem for $r = 2$.)

b) Prove constructively that $s_k \ge 3s_{k-1} - 1$ and hence that $s_k \ge (3^k - 1)/2$.

8.3.14. (!) The **composition** or **lexicographic product** of two simple graphs G and H is the simple graph $G[H]$ whose vertex set is $V(G) \times V(H)$, with edges given by $(u, v) \leftrightarrow (u', v')$ if and only if (1) uu' is an edge of G, or (2) $u = u'$ and vv' is an edge of H.

a) Prove that $\alpha(G[H]) = \alpha(G)\alpha(H)$.

b) Prove that the complement of $G[H]$ is $\overline{G}[\overline{H}]$.

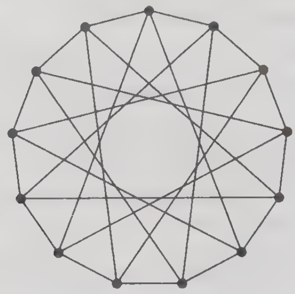
c) Use parts (a) and (b) to prove by construction that

$$R(pq+1, pq+1) - 1 \ge [R(p+1, p+1) - 1] \times [R(q+1, q+1) - 1].$$

d) Deduce that $R(2^n + 1, 2^n + 1) \ge 5^n + 1$ for $n \ge 0$ and compare this lower bound to the nonconstructive lower bound for $R(k, k)$. (Abbott [1972])

8.3.15. (−) Verify that $R(p, 2) = R(2, p) = p$. Use this and Theorem 8.3.11 to prove that $R(p, q) \le \binom{p+q-2}{p-1}$.

8.3.16. (−) Use the graph below to prove that $R(3, 5) = 14$.



8.3.17. *Ramsey numbers for $r = 2$ and multiple colors.*

a) Let $p = (p_1, \dots, p_k)$, and let q_i be obtained from p by subtracting 1 from p_i but leaving the other coordinates unchanged. Prove that $R(p) \le \sum_{i=1}^{k} R(q_i) - k + 2$.

b) Prove that $R(p_1 + 1, \dots, p_k + 1) \le \frac{(p_1+\dots+p_k)!}{p_1!\cdots p_k!}$.

8.3.18. Let $r_k = R_k(3; 2)$ (this is the value of n such that k-coloring $E(K_n)$ forces a monochromatic triangle).

a) Show that $r_k \le k(r_{k-1} - 1) + 2$.

b) Use part (a) to show that $r_k \le \lfloor k!e \rfloor + 1$, so that $r_3 \le 17$. (Comment: $r_3 = 17$, but the lower bound requires a clever 3-coloring of K_{16} that arises from the finite field $GF(2^4)$).

8.3.19. Prove that $R_k(p; r + 1) \le r + k^M$, where $M = \binom{R_k(p;r)}{r}$.

8.3.20. (+) *Off-diagonal Ramsey numbers.*

a) Prove that $R(k, l) > n$ if $\binom{n}{k} p^{\binom{k}{2}} + \binom{n}{l}(1 - p)^{\binom{l}{2}} < 1$ for some $p \in (0, 1)$. Prove that $R(k, l) > n - \binom{n}{k} p^{\binom{k}{2}} - \binom{n}{l}(1 - p)^{\binom{l}{2}}$ for all $n \in \mathbb{N}$ and $p \in (0, 1)$.

b) Use part (a) to prove $R(3, k) > k^{3/2+o(1)}$. What lower bound on $R(3, k)$ can be obtained from the first part of (a)? (Spencer [1977])

c) Use part (a) to obtain a lower bound for $R_k(q)$.

8.3.21. (!) Determine the Ramsey number $R(K_{1,m}, K_{1,n})$. (Hint: The answer depends on whether m and n are even or odd.)

8.3.22. (!) Let T be a tree with m vertices. Given that $m - 1$ divides $n - 1$, determine the Ramsey number $R(T, K_{1,n})$. (Burr [1974])

8.3.23. If $p > (m - 1)(n - 1)$, prove that every 2-coloring of $E(K_p)$ in which the red graph is transitively orientable contains a red m-clique or a blue n-clique, and prove that this is best possible. (Brozinsky–Nishiura) (Hint: Use perfect graphs.)

8.3.24. Show that $R(T, K_{n_1}, \dots, K_{n_k}) = (m - 1)(R(n_1, \dots, n_k) - 1) + 1$ when T is a tree with m vertices. (Burr)

8.3.25. Prove that $R(C_4, C_4) = 6$. (Comment: There are many proofs.)

8.3.26. Prove that $R(2K_3, 2K_3) = 10$. (Hint: Reduce to the case of a bowtie with triangles of both colors plus monochromatic 5-cycles; then use symmetry.)

8.3.27. (!) Prove that $R(mK_2, mK_2) = 3m - 1$.

8.3.28. (!) For $1 \le i \le k$, let G_i be a graph on p_i vertices, and fix a multiplicity m_i. Prove that $R(m_1G_1, \dots, m_kG_k) \le \sum(m_i - 1)p_i + R(G_1, \dots, G_k)$.

8.3.29. Frankl and Wilson [1981] explicitly constructed graphs with n vertices that have no clique or independent set with size exceeding $2^{c\sqrt{\log n \log\log n}}$, where c is a particular constant. Prove that this gives a lower bound for $R(p, p)$ that grows faster than every polynomial in p but slower than every exponential in p.

8.3.30. (!) For every simple graph G, determine $R(P_3, G)$ as a function only of the number of vertices of G and the maximum size of a matching in $\overline{G}$.

8.3.31. (!) Let r and s be natural numbers with $r + s \not\equiv 0 \bmod 4$. Prove that every 2-coloring of $E(K_{r,s})$ has a monochromatic connected graph with at least $\lceil r/2 \rceil + \lceil s/2 \rceil$ vertices. Conclude that every 3-coloring of $E(K_{r+s})$ contains a monochromatic connected subgraph with more than $(r + s)/2$ vertices. Show that this fails when 4 divides $r + s$.

8.3.32. *Forcing 4-cycles.*

a) Prove that if $\sum_{v \in V(G)} \binom{d(v)}{2} > \binom{n(G)}{2}$, then G contains a 4-cycle.

b) Prove that if $e(G) > \frac{n(G)}{4}(1 + \sqrt{4n(G) - 3})$, then G contains a 4-cycle.

c) Prove that $R_k(C_4) \le k^2 + k + 2$. (Chung–Graham [1975])

8.3.33. (!) Bondy [1971a] proved that $x \not\leftrightarrow y$ implies $d(x) + d(y) \ge n(G)$, then $G = K_{t,t}$ or G has a cycle of each length from 3 to n. Use this to prove that $R(C_m, K_{1,n}) = \max\{m, 2n + 1\}$, except possibly if m is even and at most $2n$. (Lawrence [1973])

8.3.34. (!) Prove that every 2-coloring of $E(K_n)$ has a Hamiltonian cycle that is monochromatic or consists of two monochromatic paths. (Hint: Use induction on n.) (Lovász [1979, p85, p482 - attributed to H. Raynaud])

8.3.35. (+) Let f be a 2-coloring of $E(K_n)$, and suppose that $k \ge 3$. Prove the following:

a) If f has a monochromatic C_{2k+1}, then f also has a monochromatic C_{2k}.

b) If f has a monochromatic C_{2k}, then f also has a monochromatic C_{2k-1} or $2K_k$.

c) If $m \ge 5$, then $R(C_m, C_m) \le 2m - 1$ (see Exercise 8.3.25 for $m = 4$). (Hint: Use parts (a) and (b) and the result of Erdős–Gallai [1959] (Theorem 8.4.35) that $e(G) > (m - 1)(n(G) - 1)/2$ forces a cycle of length at least m in G. There remains one difficult case).

8.3.36. The **Ramsey multiplicity** of G is the minimum number of monochromatic copies of G in a 2-coloring of the edges of a clique on $R(G, G)$ vertices. Show that the Ramsey multiplicity of K_3 is 2.

8.3.37. Prove that each point in a triangular region has a unique expression as a convex combination of the vertices of the triangle (convex combinations are linear combinations where the coefficients are nonnegative and sum to 1).

8.3.38. *Sperner's Lemma in higher dimensions.* A **k-dimensional simplex** consists of the convex combinations of $k + 1$ points in $\mathbb{R}^k$ not lying in a hyperplane. A **simplicial subdivision** expresses a k-dimensional simplex as a union of k-dimensional simplices (cells) such that any two cells intersect in the simplex determined by their common corners. A **completely labeled** cell has $\{0, \dots, k\}$ at its corners.

State a general definition of "proper labeling" so that every proper labeling of a simplicial subdivision of a k-simplex contains a completely labeled cell. Prove this theorem. (Hint: The proof of Sperner's Lemma in two dimensions (Theorem 8.3.17) is an instance of the induction step for a proof by induction on k.)

8.3.39. (–) Compute the bandwidths of P_n, K_n, and C_n.

8.3.40. Compute the bandwidth of $K_{n_1,\ldots,n_k}$. (Eitner [1979])

8.3.41. (!) Prove that every tree with k leaves is the union of $\lceil k/2 \rceil$ pairwise intersecting paths (Exercise 2.1.37). Use this to prove that the bandwidth of a tree with k leaves is at most $\lceil k/2 \rceil$. (Ando–Kaneko–Gervacio [1996])

8.3.42. (+) Let G be a caterpillar (Definition 2.2.17), and let m be an integer such that $\left\lceil \frac{n(H)-1}{\text{diam } H} \right\rceil \le m$ for all $H \subseteq G$. Prove that $B(G) \le m$. (Hint: Prove that G has a numbering f in which $f(v)$ is a multiple of m whenever v is on the spine and $|f(u) - f(v)| \le m$ for all $u \leftrightarrow v$.) (Sysło–Zak [1982], Miller [1981])

8.3.43. *Bandwidth of grids.*

a) Compute the local density lower bound for the bandwidth of $P_m \square P_n$.

b) Let S be a k-set of vertices in $P_n \square P_n$ with a_i vertices in the ith row and b_j vertices in the jth column. Prove that $|\partial T| \le |\partial S|$ if T is the set consisting of the first a_i vertices in the ith row for each i.

c) Prove that $|\partial S|$ is minimized over k-sets in $V(P_n \square P_n)$ by some S such that $a_1 \ge \cdots \ge a_n$ and $b_1 \ge \cdots \ge b_n$. Conclude that Harper's lower bound for $B(P_n \square P_n)$ is n.

d) Conclude that $B(P_m \square P_n) = \min\{m, n\}$. (Chvátalová [1975])

8.3.44. (+) Let G be a simple graph with order n and bandwidth b.

a) For $e \in \overline{G}$, prove that $B(G + e) \le 2b$.

b) Prove that if $n \ge 6b$, then $B(G + e)$ can be as large as $2b$.

(Comment: The maximum of $B(G+e)$ is $b+1$ if $n \le 3b+4$ and is $\lceil (n-1)/3 \rceil$ if $3b+5 \le n \le 6b-2$.) (Wang–West–Yao [1995])

8.4. More Extremal Problems

Extremal graph theory is a huge area. In Section 1.3 we described the distinction between optimization problems (find an extremal structure in the input graph) and extremal problems (find an extremal instance over a class of graphs), and we have studied both types of problems throughout this book. In this section we study the latter type. The archetypal example is the Turán problem: find the maximum number of edges in a graph not containing H as a subgraph. We list one additional example from each chapter.

Objective	*Class of graphs*	*Answer*	*Reference*
max $e(G)$	n vertices and k components	$\binom{n-k+1}{2}$	Exercise 1.3.40
max girth	diameter k and not a tree	$2k+1$	Exercise 2.1.61
max $\beta(G)$	$\alpha'(G) \le k$	$2k$	Exercise 3.3.10
min $\alpha(G)$	$\kappa(G) = k$ and diameter d	$\lceil (d+1)/2 \rceil$	Exercise 4.2.22
max $\chi(G)$	$2K_2$-free and $\omega(G) = k$	$\binom{k+1}{2}$	Exercise 5.2.11
max $\chi(G)$	outerplanar	3	Exercise 6.3.12
max $e(G)$	$n(G) = n$ and non-Hamiltonian	$\binom{n-1}{2}+1$	Exercise 7.2.26
max $n(G)$	$\omega(G) < p$ and $\alpha(G) < q$	$R(p,q)-1$	Section 8.3

With such enormous variety of extremal problems, we can only hope in this section to exhibit a small sample of interesting results.

ENCODINGS OF GRAPHS

We first consider parameters related to three types of graph encoding. Each model of encoding involves assigning vectors to vertices, and the parameter is the minimum length of vectors that suffice. We study the maximum of this parameter over n-vertex graphs. The parameters are intersection number, product dimension, and squashed-cube dimension.

8.4.1. Definition. An **intersection representation** of **length** t assigns each vertex a 0,1-vector of length t such that $u \leftrightarrow v$ if and only if their vectors have a 1 in a common position. Equivalently, it assigns each $x \in V(G)$ a set $S_x \subseteq [t]$ such that $u \leftrightarrow v$ if and only if $S_u \cap S_v \neq \emptyset$. The **intersection number** $\theta'(G)$ is the minimum length of an intersection representation of G.

The elements of $[t]$ in a representation correspond to complete subgraphs that cover $E(G)$. This motivates our use of θ' for intersection number: $\theta(G)$ is the minimum number of cliques needed to cover $V(G)$.

8.4.2. Proposition. (Erdős–Goodman–Pósa [1966]) The intersection number equals the minimum number of complete subgraphs needed to cover $E(G)$.

Proof: We define a natural correspondence between representations of length t and coverings of $E(G)$ by t complete subgraphs. Each $i \in [t]$ generates a clique $\{v \in V(G): i \in S_v\}$. The resulting complete subgraphs cover $E(G)$, since $u \leftrightarrow v$ if and only if $S_u \cap S_v \neq \emptyset$.

Conversely, if complete subgraphs $Q_1, \dots, Q_t$ cover $E(G)$, then assigning $\{i: v \in V(Q_i)\}$ to each vertex v yields an intersection representation. ■

Hence $\theta'(G) = e(G)$ if G is triangle-free, and $\theta'(K_{\lfloor n/2 \rfloor, \lceil n/2 \rceil}) = \lfloor n^2/4 \rfloor$. In fact, this is the unique n-vertex graph maximizing $\theta'(G)$. Exercise 1 suggests a direct proof of the bound; here we present a stronger result.

Let **F** be a family of graphs. For an input graph G, the **F-decomposition** problem is to decompose G into the minimum number of graphs in **F**. When **F** is not closed under taking subgraphs, **F**-decomposition may require more subgraphs than **F-covering**. For example, we can cover the kite with two complete subgraphs, but three complete subgraphs are needed to decompose it.

Proving $\theta'(G) \leq \lfloor n^2/4 \rfloor$ for n-vertex graphs means showing that every n-vertex graph can be covered with $\lfloor n^2/4 \rfloor$ complete subgraphs; we prove the stronger result that there is always a decomposition using at most this many complete subgraphs. In fact, we can find such a decomposition greedily.

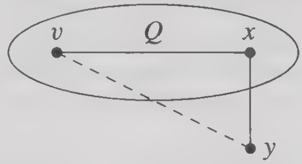
8.4.3. Theorem. (McGuinness [1994]) Every greedy clique decomposition of an n-vertex graph uses at most $\lfloor n^2/4 \rfloor$ cliques.

Proof: We use induction on n. The claim is obvious for $n \leq 2$; consider $n > 2$. Let $\mathbf{Q} = Q_1, \dots, Q_m$ be a greedy decomposition of G, meaning that each Q_i is a maximal complete subgraph in $G - \cup_{j<i} E(Q_j)$. Note that deleting Q_j from the list $\mathbf{Q}$ leaves a greedy decomposition of $G - E(Q_j)$.

If each Q_i has at least three edges, then $m < n^2/6$, so we may assume that some Q_j is an edge xy. Let R consist of the elements of $\mathbf{Q} - \{Q_j\}$ that are incident to x, and let S consist of those incident to y. The set $\mathbf{Q}' = \mathbf{Q} - (R \cup S \cup \{Q_j\})$ is a greedy decomposition of a subgraph of $G - x - y$. By the induction hypothesis, $|\mathbf{Q}'| \le (n-2)^2/4$. Hence it suffices to prove that $|R| + |S| \le n - 2$.

We prove this by choosing distinct vertices in $V(G) - \{x, y\}$ from the vertex sets of the elements of $R \cup S$. Since each edge is deleted exactly once, each $v \notin \{x, y\}$ appears once in R if $v \in N(x)$ and once in S if $v \in N(y)$. Consider $Q \in R$. If Q uses a vertex $v \notin N(y)$, then we choose such a v for Q. If $V(Q) \subseteq N(y)$, then we choose for Q a vertex $v \in Q$ such that vy belongs to the earliest element of $\mathbf{Q}$ that contains both y and some vertex of Q. Call this element Q'; note that Q' is the only element of S containing v. Since Q and xy are maximal when chosen, Q' precedes both of these in $\mathbf{Q}$. For elements of S, choose vertices by reversing the roles of x and y.

We have shown that if v belongs to some $Q \in R$ and to some $Q' \in S$, and v is chosen for one of them, then the one for which it is chosen occurs after the other one in the list $\mathbf{Q}$. Hence no vertex is chosen twice. We conclude that $|R| + |S| \le n - 2$ and $m \le n^2/4$. ■



Both Chung [1981] and Győri–Kostochka [1979] strengthened the decomposition bound, proving that every n-vertex graph has a decomposition into complete subgraphs whose orders sum to at most $\lfloor n(G)^2/2 \rfloor$.

Now we consider the second encoding model.

8.4.4. Definition. A **product representation** of **length** t assigns the vertices distinct vectors of length t so that $u \leftrightarrow v$ if and only if their vectors differ in every position. The **product dimension** pdim G is the minimum length of such a representation of G.

By devoting one coordinate to each $e \in E(\overline{G})$, in which the vertices of e have value 0 and other vertices have distinct positive values, we obtain pdim $G \le e(\overline{G})$ (if G is not a complete graph).

8.4.5. Example. Every complete graph has product dimension 1. For $\overline{K}_n$, each pair of vertices must agree in some coordinate, but we cannot assign two vertices the same vector. Hence two coordinates are needed, and assigning $(0, j)$ to v_j for each j suffices.

For $K_1 + K_{n-1}$, the vectors for the clique must differ in each coordinate. The vector for the isolated vertex must agree with each of the others somewhere, but it cannot agree with more than one in a single coordinate. Hence at least $n-1$ coordinates are needed. This suffices, by using $(1, 2, \ldots, n-1)$ for the isolated vertex and $(i, i, \ldots, i)$ for the ith vertex of the clique. ■

Again we can describe the parameter using complete graphs.

8.4.6. Definition. An **equivalence** on G is a spanning subgraph of G whose components are complete graphs.

8.4.7. Proposition. The product dimension of G is the minimum number of equivalences $E_1, \ldots, E_t$ such that $\bigcup E_i = \overline{G}$ and $\bigcap E_i = \emptyset$.

Proof: Again there is a natural bijection. Given a product representation, the ith coordinate generates E_i, with a component for each value used in the ith coordinate. Every nonadjacent pair agrees in some coordinate, so every edge of $\overline{G}$ is covered.

Conversely, given $E_1, \ldots, E_t$, each component of E_i becomes a fixed value in the ith coordinate of a representation. The requirement $\cap E_i = \emptyset$ is the requirement of using distinct vectors in the product representation. ■

8.4.8. Lemma. If $\chi'(\overline{G}) > 1$, then $\text{pdim}\, G \le \chi'(\overline{G})$, with equality if $\overline{G}$ is triangle-free.

Proof: Every matching is a disjoint union of complete graphs and becomes an equivalence by the addition of isolated vertices; hence $\chi'(\overline{G})$ equivalences cover $\overline{G}$. If $\chi'(\overline{G}) > 1$, then these equivalences have no common edge.

If $\overline{G}$ is triangle-free, then every equivalence used in a cover of $\overline{G}$ is a matching plus isolated edges, and thus $\chi'(\overline{G}) \le \text{pdim}\, G$. ■

8.4.9. Corollary. For $n \ge 3$, the maximum product dimension of an n-vertex graph is $n-1$.

Proof: Let G be an n-vertex graph. By Lemma 8.4.8 and Vizing's Theorem (Theorem 7.1.10), $\text{pdim}\, G \le \chi'(\overline{G}) \le \Delta(\overline{G}) + 1 \le n$. Furthermore, the bound improves to $n-1$ unless $\Delta(\overline{G}) = n-1$. Let S be the set of vertices of degree $n-1$ in $\overline{G}$; we may assume that $|S| = k \ge 1$.

By Lemma 8.4.8 and Vizing's Theorem again, $\text{pdim}\,(G - S) \le n - k$. By duplicating coordinates if needed, we obtain a product representation of $G - S$ of length $n-k$. Let x^i be the vector assigned to v_i in this representation.

Each vertex of S is isolated in G. We now assign to each $v \in S$ the vector whose ith coordinate, for $1 \le i \le n-k$, is the ith coordinate of x^i. If $k = 1$, then this completes a representation of G with length $n-1$. If $k > 1$, then we have assigned the same vector to all of S; we add one coordinate using distinct values to complete a representation of length $n-k+1$, which is less than $n-1$.

Since $\text{pdim}\,(K_1 + K_{n-1}) = n-1$ (Example 8.4.5), the bound is sharp. ■

Lovász–Nešetřil–Pultr [1980] characterized the n-vertex graphs with product dimension $n-1$ (Exercise 4). They also proved a general lower bound using a dimension argument in linear algebra.

8.4.10. Theorem. (Lovász–Nešetřil–Pultr [1980]) Let $u_1, \ldots, u_r$ and $v_1, \ldots, v_r$ be two lists of vertices (not necessarily distinct) in a graph G. If $u_i \leftrightarrow v_j$ for $i = j$ and $u_i \not\leftrightarrow v_j$ for $i < j$, then pdim $G \geq \lceil \lg r \rceil$.

Proof: Let G have a representation of length d. Let $x^1, \ldots, x^r$ and $y^1, \ldots, y^r$ be the vectors for $u_1, \ldots, u_r$ and $v_1, \ldots, v_r$, respectively. The vectors x^i and y^i differ in every coordinate, but x^i and y^j agree in some coordinate if $i \neq j$. Hence $\prod_{k=1}^{d}(x_k^i - y_k^j)$ is nonzero if and only if $i = j$.

We use this product property to construct r linearly independent vectors in $\mathbb{R}^{2^d}$; this proves that $r \leq 2^d$ and hence that pdim $G \geq \lceil \lg r \rceil$. Expansion of $\prod_{k=1}^{d}(w_k - z_k)$ for $w, z \in \mathbb{R}^d$ yields the sum $\sum_{S \subseteq [d]} \prod_{i \in S} w_i \prod_{j \in \overline{S}}(-z_j)$. To relate r to 2^d, we view this as a dot product in $\mathbb{R}^{2^d}$, with coordinates indexed by the subsets of $[d]$. For each $w \in \mathbb{R}^d$, define two vectors in $\mathbb{R}^{2^d}$ by setting $\overline{w}_S = \prod_{i \in S} w_i$ and $\hat{w}_S = \prod_{i \notin S}(-w_i)$ for the coordinate $S \subseteq [d]$. With this definition, the dot product $\overline{w} \cdot \hat{z}$ equals $\prod_{k=1}^{d}(w_k - z_k)$. The conditions on the x's and y's thus imply that $\overline{x}^i \cdot \hat{y}^j$ is nonzero if and only if $i = j$.

We claim that $\overline{x}^1, \ldots, \overline{x}^r$ are independent. Consider a linear dependence $\sum_{i=1}^{r} c_i \overline{x}^i = \mathbf{0}$. Taking the dot product of $\hat{y}^r$ with both sides kills all terms below $i = r$, yielding $c_r \overline{x}^r \cdot \hat{y}^r = 0$. Since $\overline{x}^r \cdot \hat{y}^r \neq 0$, we have $c_r = 0$. We can now apply the same argument using $\hat{y}^{r-1}$. Knowing that $c_r = 0$ yields $c_{r-1}\overline{x}^{r-1} \cdot \hat{y}^{r-1} = 0$. Successively decreasing the index yields $c_j = 0$ for all j. We conclude that $\overline{x}^1, \ldots, \overline{x}^r$ are independent, which requires $2^d \geq r$. ∎

8.4.11. Example. *Matchings:* pdim $(n/2)K_2 = \lceil \lg n \rceil$. Given k coordinates, the graph encoded by using all 2^k binary k-tuples as codes is $2^{k-1}K_2$, since only with its complement does a vector disagree in each position. If n is not a power of 2, then we can discard complementary pairs to obtain a construction. The lower bound follows from Theorem 8.4.10, using each vertex in each list (for example, set $u_i = v_{n+1-i}$). ∎

In our third encoding model, we want to recover more detailed information: distance between vertices. This arises from an addressing problem in communication networks. Each message should travel a shortest path to its destination. Without central control, a vertex receiving a message must determine where to send it using only the name of the destination. If the vectors for two vertices yield the distance between them in G, then a vertex can compare the destination vector with the vectors for its neighbors and send the message to a neighbor closest to the destination.

For a connected graph G, we want to assign vectors to vertices such that the distance between vertices is the number of positions where their vectors differ. This is an **isometric** or "**distance-preserving**" embedding of G into $H = K_{n_1} \square \cdots \square K_{n_t}$, meaning a mapping $f\colon V(G) \to V(H)$ such that $d_G(u, v) = d_H(f(u), f(v))$. However, many connected graphs have no isometric embedding in a cartesian product of cliques; C_{2k+1} for $k \geq 2$ is an example (Exercise 11).

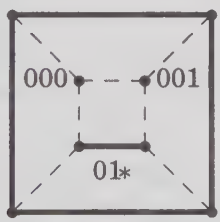
Hence we introduce a "don't care" symbol $*$. Let $S = \{0, 1, *\}$, and define a symmetric function d by $d(0, 1) = 1$ and $d(0, *) = 0 = d(1, *)$. Let S^N denote the set of N-tuples (vectors) with entries in S, and for $a, b \in S^N$ let $d_S(a, b) = \sum d(a_i, b_i)$. For each graph G, we obtain for some N an encoding $f\colon V(G) \to S^N$ so that $d_G(u, v) = d_S(f(u), f(v))$ for all $u, v \in V(G)$.

Each $a \in S^N$ corresponds to a subcube of Q_N, the N-dimensional cube; the dimension of the subcube is the number of $*$s in a. For $a, b \in S^N$, the minimum distance between vertices of the corresponding subcubes is $d_S(a, b)$. The vectors assigned to distinct vertices correspond to disjoint subcubes, else their distance would be 0. If we contract the edges of each assigned subcube, we obtain a "squashed cube" H. The distance-preserving map $f\colon V(G) \to S^N$ is an isometric embedding of G in H.

8.4.12. Definition. A **squashed-cube embedding** of **length** N is a map $f\colon V(G) \to S^N$ such that $d_G(u, v) = d_S(f(u), f(v))$. The **squashed-cube dimension** qdim G is the minimum length of such an embedding of G.

8.4.13. Example. The vectors 000, 001, 01$*$, and 1$**$ form a squashed-cube embedding of K_4 with length 3. Two adjacent vertices of the 3-cube remain unchanged, an edge adjacent to both collapses, and the entire opposite face collapses. The resulting graph is K_4. The image subcubes appear below in bold. The construction generalizes to embed K_n in a squashed $n-1$-dimensional cube.

The path P_n embeds isometrically in Q_{n-1} without squashings, using $00\cdots00$, $10\cdots00$, $11\cdots00, \ldots, 11\cdots10$, $11\cdots11$. No shorter embedding exists, because the distance between the endpoints of P_n is $n-1$, and each coordinate contributes at most 1 to the distance between vectors. ■



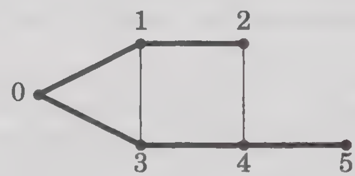
8.4.14. Proposition. For a graph G, qdim $(G) \le \sum_{i<j} d_G(v_i, v_j)$.

Proof: For each pair i, j with $i < j$, we dedicate a block of $d_G(v_i, v_j)$ coordinates. Set these coordinates to 0 for v_i, to 1 for v_j, and to $*$ for other vertices. Given two vertices, the only coordinates where neither contains $*$ are the coordinates dedicated to the pair, so $d_G(v_i, v_j) = d_S(f(v_i), f(v_j))$. ■

Using an eigenvalue technique (Exercise 8.6.14), Graham and Pollak [1971, 1973] proved a general lower bound on qdim (G) that yields qdim $K_n = n-1$. Hence K_n and P_n both have squashed-cube dimension $n-1$; Graham and Pollak conjectured that qdim $G \le n-1$ for every n-vertex connected graph. Graham offered \$100 for a proof, and Winkler found an encoding scheme to prove this "Squashed-Cube Conjecture".

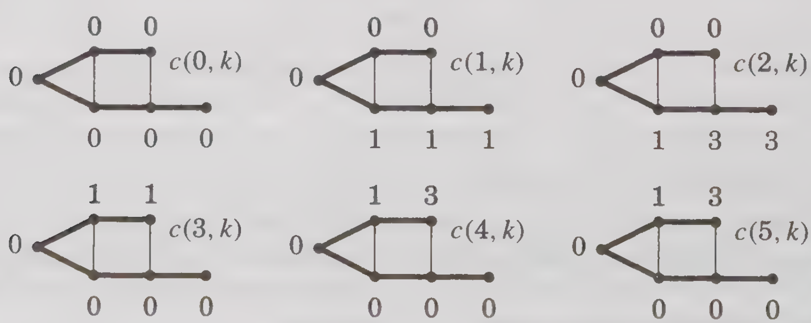
Winkler's proof generates an explicit $n-1$-dimensional squashed-cube encoding for each connected n-vertex graph G. We begin by indexing the vertices; choose v_0 arbitrarily. Next, find a spanning tree T such that $d_T(v, v_0) = d_G(v, v_0)$ for all $v \in V(G)$ (T can be generated by a breadth-first search from v_0). Now, number the vertices by a *depth*-first search in T. In other words, having chosen the indexing for $v_0, \dots, v_i$, let v_{i+1} be an unvisited child of v_i in T, if one exists; otherwise backtrack toward the root until a vertex with such a child is found. The resulting indices increase along every path from v_0 in T.

8.4.15. Example. *Depth-first numbering of a breadth-first spanning tree.* Below, the bold edges belong to T and the solid edges to $G - T$. We will use this example to illustrate several steps in the proof. ■



We henceforth fix T and this ordering and refer to vertices by their index in this ordering. Let P_i be the vertex set of the $i, 0$-path in T, let i' be the parent of i in T (the next vertex on the path from i to 0), and let $i \wedge j = \max(P_i \cap P_j)$ be the vertex at which the $i, 0$-path and $j, 0$-path meet. Given a depth-first numbering of a breadth-first tree T in G, let $c(i, j) = d_T(i, j) - d_G(i, j)$ be the **discrepancy** of two vertices i, j.

8.4.16. Example. In the graph marked $c(i, k)$ below, we record at each vertex k the discrepancy $c(i, k)$ for the tree T in Example 8.4.15. ■



8.4.17. Lemma. (Winkler [1983]). Discrepancy has the following properties.

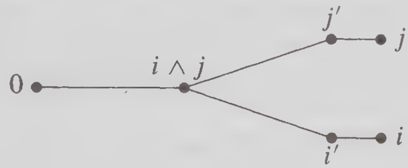
a) $c(i, j) = c(j, i) \geq 0$.

b) If $i \in P_j$, then $c(i, j) = 0$.

c) If neither $i \in P_j$ nor $j \in P_i$, then $c(i, j') \leq c(i, j) \leq c(i, j') + 2$.

Proof: (a) Distance in graphs is symmetric, and the shortest i, j-path in G is no longer than the path between them in T. (b) The preservation of distances to

v_0 implies that the i, j-path in T is a shortest i, j-path in G. (c) Since j' belongs to the i, j-path in T, we have $d_T(i, j) - d_T(i, j') = 1$. Since $jj' \in E(G)$, we have $|d_G(i, j) - d_G(i, j')| \le 1$. Thus $c(i, j) - c(i, j')$ is 0, 1, or 2. ■



With this notion of discrepancy, we can give an overview of how Winkler's encoding works. We use a search tree because it gives us $n - 1$ natural coordinates. Distance in the tree is an "approximation" to distance in the graph; it needs to be adjusted (reduced) by the discrepancy. Winkler's encoding puts a 1 in coordinate k for exactly one of vertices i and j for exactly $d_T(i, j)$ values of k. We want the other code to have a 0 in exactly $d_G(i, j)$ of these coordinates, so we perform the adjustment by having $*$ in exactly $c(i, j)$ of the coordinates where one code has a 1. The problem is to design the encoding to achieve this simultaneously for all pairs of vertices.

8.4.18. Theorem. (Winkler [1983]) Every connected n-vertex graph G has squashed-cube dimension at most $n - 1$.

Proof: Choose a tree T and numbering $0, \ldots, n-1$ as described above. We define an encoding $f(i) = (f_1(i), \ldots, f_{n-1}(i))$ and verify that $d_G(i, j) = d_S(a_i, a_j)$. The encoding is

$$f_k(i) = \begin{cases} 1 & \text{if } k \in P_i \\ * & \text{if } c(i,k) - c(i,k') = 2 \\ * & \text{if } c(i,k) - c(i,k') = 1 \text{ and } i < k \text{ and } c(i,k) \text{ is even} \\ * & \text{if } c(i,k) - c(i,k') = 1 \text{ and } i > k \text{ and } c(i,k) \text{ is odd} \\ 0 & \text{otherwise} \end{cases}$$

(The vectors in the encoding for Example 8.4.16 are $f(0) = 00000$, $f(1) = 10000$, $f(2) = 110{*}0$, $f(3) = {*}0100$, $f(4) = {*}{*}110$, and $f(5) = {*}{*}111$.)

To prove that $d_S(f(i), f(j)) = d_G(i, j)$, we count the coordinates where one of $f(i), f(j)$ has a 1 and the other has a 0. Such coordinates k belong to $P_i \cup P_j$, where all the 1's are located. By symmetry, we may assume that $i < j$. Hence $j \notin P_i$, and we consider the two cases $i \in P_j$ and $i \notin P_j$.

If $i \in P_j$, then $d_G(i, j) = d_T(i, j) = |P_j - P_i|$, and $f_k(i) = f_k(j) = 1$ if and only if $k \in P_i$. The coordinates where exactly one of $f(i), f(j)$ has a 1 all lie in $P_j - P_i$. For $k \in P_j - P_i$, we have $f_k(i) = 0$, and thus $d_G(i, j) = d_S(f(i), f(j))$.

If $i \notin P_j$, then exactly one of $\{f_i(k), f_j(k)\}$ equals 1 precisely when $k \in (P_j - P_i) \cup (P_i - P_j)$. We need to prove that the other vector has $*$ in exactly $c(i, j)$ of these coordinates. This will yield

$$d_S(f(i), f(j)) = |P_j - P_i| + |P_i - P_j| - c(i, j) = d_T(i, j) - c(i, j) = d_G(i, j).$$

In Example 8.4.16, $(P_5 - P_2) \cup (P_2 - P_5)$ is all five coordinates; since $f(2)$ and $f(5)$ together have $*$ in three of these coordinates, we have $d_S(f(2), f(5)) = d_G(2, 5) = 2$, as desired.

To locate the $*$'s in these positions, consider the change in discrepancies as we bring either of i, j to the point where P_i, P_j meet. Consider two lists:

$$0 = c(i, i \wedge j) \leq \cdots \leq c(i, j') \leq c(i, j)$$
$$0 = c(i \wedge j, j) \leq \cdots \leq c(i', j) \leq c(i, j).$$

We will obtain one $*$ in $f(i)$ for each even m with $0 < m \leq c(i, j)$ and one $*$ in $f(j)$ for each odd m with $0 < m \leq c(i, j)$.

For even m with $0 < m \leq c(i, j)$, let j_m be the unique vertex such that $c(i, j_m) \geq m$ and $c(i, j_m') < m$. Even when the value m is not in the first list, j_m is well-defined. Because c changes by at most 2 with each step, the values of j_m are distinct. Furthermore, the depth-first ordering guarantees $i < k$ for all $k \in P_j - P_i$. Thus $f_k(i) = *$ for $k \in P_j - P_i$ if and only if $k = j_m$ for some even m. In Example 8.4.16, for $(i, j) = (2, 5)$ we have $j_2 = 4$ and $f_4(2) = *$.

Similarly, for odd m with $0 < m \leq c(i, j)$, let i_m be the unique vertex such that $c(i_m, j) \geq m$ and $c(i_m', j) < m$. As before, the values of i_m are distinct and well-defined. The depth-first ordering guarantees $j > k$ for all $k \in P_i - P_j$, so $a_j(k) = *$ for $k \in P_i - P_j$ if and only if $k = i_m$ for some odd m. In Example 8.4.16, for $(i, j) = (2, 5)$ we have $i_1 = 1$, $i_3 = 2$, and $f_1(j) = f_3(j) = *$.

Thus, we have counted the $*$'s in $P_i - P_j \cup P_j - P_i$. Their number is the number of even integers between 1 and $c(i, j)$ plus the number of odd integers between 1 and $c(i, j)$, which together equals $c(i, j)$. ∎

BRANCHINGS AND GOSSIP

We have studied the problem of finding the maximum number of pairwise edge-disjoint spanning trees in a graph; this equals the maximum k such that for every vertex partition P, there are at least $k(|P| - 1)$ edges crossing between sets of P (Corollary 8.2.59). Here we consider an analogous problem for digraphs that is related to Menger's Theorem (Exercise 14). Menger's Theorem is a min-max theorem that focuses on vertex pairs. We examine "connectedness" from a single vertex to the rest of the digraph.

8.4.19. Definition. An **r-branching** in a digraph is a rooted tree "branching out" from r. Vertex r has indegree 0, all other vertices have indegree 1, and all other vertices are reachable from r. Let $\kappa'(r; G)$ denote the minimum number of edges whose deletion makes some vertex unreachable from r.

Deleting the edges entering a set $X \subseteq V(G) - \{r\}$ makes each vertex of X unreachable from r. On the other hand, a minimal set whose deletion makes some vertex unreachable includes all edges leaving the set of reached vertices. Hence $\kappa'(r; G)$ equals the minimum, over nonempty $X \subseteq V(G) - \{r\}$, of the number of edges entering X.

In a set of pairwise edge-disjoint r-branchings, each must use at least one edge entering X. Thus there are at most $\kappa'(r;G)$ pairwise edge-disjoint r-branchings in G. Edmonds proved that this bound is achievable. Our discussion allows multiple edges.

8.4.20. Theorem. (Edmonds' Branching Theorem [1973]) *For a vertex r in a digraph G, the maximum number of pairwise edge-disjoint r-branchings in G is $\kappa'(r;G)$.*

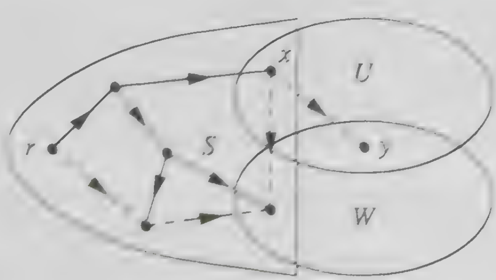
Proof: (Lovász [1976]) Let V be the vertex set of G. The upper bound holds since each subset of $V - r$ is entered by at least one edge in every r-branching. We prove the existence of $\kappa'(r;G)$ edge-disjoint r-branchings by induction on $k = \kappa'(r;G)$. For $k = 1$, a breadth-first search suffices to grow an r-branching, since every vertex is reachable. For $k > 1$, we seek an r-branching T such that $\kappa'(r;G - E(T)) = k - 1$; the induction hypothesis then supplies $k - 1$ additional r-branchings.

A *partial* r-branching is an r-branching of an induced subgraph of G. Let T be a partial r-branching of maximum order such that $\kappa'(r;G - E(T)) \ge k - 1$. The vertex r itself is such a branching, with $E(T) = \emptyset$. Let $S = V(T)$. If $S = V$, then we are done, so we may assume that $S \ne V$.

For $X \subseteq V - r$, let e_X denote the number of edges in $G - E(T)$ that enter X. If $e_X \ge k$ for every $X \subseteq V - r$ that intersects $V - S$, then we can extend T by adding any edge from S to $V - S$. Hence we can choose a smallest set $U \subseteq V - r$ that intersects $V - S$ and is entered by exactly $k - 1$ edges. (In the illustration, T consists of the solid edges.)

Because $\kappa'(r;G) = k$ and we have deleted no edge entering $U - S$, we still have $e_{U-S} \ge k$. However, $e_U = k - 1$, so there must be an edge xy from $S \cap U$ to $U - S$. We claim that xy can be added to enlarge T, contradicting the maximality of T. We need only verify that at least $k - 1$ edges still enter each $W \subseteq V - r$ when we delete xy from $G - E(T)$. This holds trivially unless $x \in V - W$ and $y \in W$. It suffices to prove that $e_W \ge k$ for such a W.

The quantity $e_W + e_U$ counts edges entering W and entering U. Except for the edges between $U - W$ and $W - U$, these enter $W \cup U$, and those entering $W \cap U$ are counted twice. Thus $e_W + e_U \ge e_{W \cup U} + e_{W \cap U}$. We have $e_{W \cup U} \ge k - 1$ by the defining property of T, $e_U = k - 1$ by construction, and $e_{W \cap U} \ge k$ by $x \in U - W$ and the minimality of U. Hence $e_W \ge k - 1 - (k - 1) + k = k$, as desired. ■



Lovász's proof can be converted to an algorithm for finding the maximum number of pairwise disjoint r-branchings; Tarjan [1974/75] gave another algorithm. We might call $\kappa'(r; G)$ the **local-global edge-connectivity**. Theorem 8.4.20 has several equivalent forms:

8.4.21. Corollary. If G is a directed graph, r is a vertex of G, and $k \geq 0$, then the following statements are equivalent.

A) G has k pairwise edge-disjoint r-branchings.

B) $\kappa'(r; G) \geq k$; equivalently, $|[\overline{X}, X]| \geq k$ for all $X \subseteq V(G) - \{r\}$.

C) For each $s \neq r$ there exist k pairwise edge-disjoint r, s-paths.

D) There exist k pairwise edge-disjoint spanning trees of the underlying (undirected) graph that for each $s \neq r$ contain among them exactly k edges of the digraph G entering s.

Proof: $A \Leftrightarrow B$ is Edmonds' Theorem, $B \Leftrightarrow C$ is Menger's Theorem, and $A \Rightarrow D$ is immediate. For $D \Rightarrow B$, assume that the trees exist and consider $U \subseteq V - r$. Each spanning tree has at most $|U| - 1$ edges within U, so the trees together have at most $k(|U|-1)$ edges within U. By hypothesis, the edges of the digraph G corresponding to these trees contain exactly $k\,|U|$ edges with heads in U, so at least k edges enter U. ■

Schrijver observed that Edmonds' Branching Theorem can also be proved using matroid union and matroid intersection. Discard the edges entering the root r. Let M_1 be the union of k copies of the cycle matroid on the underlying undirected graph. Let M_2 be the matroid in which a set of edges is independent if and only if no $k + 1$ of them have the same head (this is the direct sum of uniform matroids of rank k). There exist k disjoint r-branchings if and only if these two matroids have a common independent set of size $k(n(G) - 1)$.

Pairwise edge-disjoint r-branchings provide a fault-tolerant static protocol for message transmissions from r; alternative trees are available. We next consider a static protocol for transmissions from each vertex to every other. Each transmission is two-way, but they are performed in a specified order.

The resulting question is the **gossip problem**. Consider n gossips, each having a tidbit of information. Being gossips, each wants to know all the information, and when two communicate they tell each other everything they know. How many telephone calls are needed to transmit all the information? Several solutions were published in the early 1970s.

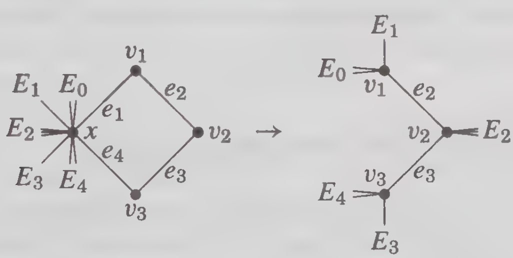
Succeeding with $2n - 3$ calls is easy: everyone calls x, and then x calls everyone back, saving one call by combining the last call in and first call out. When $n \geq 4$, $2n - 4$ calls suffice: first the others call in to a set S of four people, then S shares the information in two successive pairings, and finally the others receive calls back from S, using a total of $(n - 4) + 4 + (n - 4) = 2n - 4$ calls. Using a graph model, we show that this is optimal.

8.4.22. Definition. An **ordered graph** is a graph with an ordering of the edges (multiple edges allowed). An **increasing path** is a path via successively later edges. A **gossip scheme** is an ordered graph having an increasing path from each vertex to every other vertex. A gossip scheme **satisfies NOHO** ("No One Hears his or her Own information") if it has no increasing x, y-path plus a later edge between x and y.

8.4.23. Theorem. For $n \geq 4$, the minimum number of edges in a gossip scheme on n vertices is $2n - 4$.

Proof: (Baker–Shostak [1972]). We freely use "calls" in place of "edges" to emphasize the ordering and the possibility of repeated edges. The scheme described above uses $2n - 4$ calls, and case analysis shows that it is optimal for $n = 4$. This provides the basis for a proof by induction on n. For $n > 4$, we may assume that every gossip scheme with $n-1$ vertices uses at least $2n-6$ calls. If $2n - 4$ is not optimal for n vertices, then we can add calls to the optimal scheme (if necessary) to obtain an n-vertex gossip scheme G with exactly $2n - 5$ calls.

Claim 1. G satisfies NOHO. Otherwise, G has an increasing path from x to v_k along edges $e_1, \ldots, e_k$ followed by a call $e_{k+1} = v_k x$. Delete e_1 and e_{k+1}. Partition the other calls involving x into $k + 2$ sets: E_0 consists of those before e_1, E_i for $1 \leq i \leq k$ consists of those between e_i and e_{i+1}, and E_{k+1} consists of those after e_{k+1}. In each edge $e \in E_i$, replace x by v_1, v_i, or v_k in the cases $i = 0$, $1 \leq i \leq k$, or $i = k + 1$, respectively (see illustration). Now $E(G) - \{e_1, e_{k+1}\}$ is a gossip scheme on $V(G) - \{x\}$, because every increasing path through x is replaced by an increasing path that consists of the same edges and perhaps additional edges from $\{e_i\}$. The scheme has $2(n-1)-5$ edges, which contradicts the induction hypothesis.



Claim 2. $d(x) - 3$ calls are useless to x, and hence $\delta(G) \geq 3$. Let $O(x)$ be the set of calls on which some vertex is reached for the first time by an increasing path "Out" from x; these calls form a tree. The tree $I(x)$ of edges useful "In" to x is $O(x)$ for the reverse order on $E(G)$. We show that $O(x) \cap I(x)$ is the set of edges incident to x. If an increasing x, y-path reaches $y \in N(x)$ before the edge xy, then x violates NOHO; hence $xy \in O(x)$. Similarly, $xy \in I(x)$. Conversely, if $O(x) \cap I(x)$ contains an edge e not incident to x, then an increasing path from x containing e and an increasing path to x containing e combine to violate NOHO for x. Hence $|O(x) \cap I(x)| = d(x)$. The edges "useless to x" are those not in $O(x) \cup I(x)$. We have

$$\left|\overline{O(x) \cup I(x)}\right| = 2n - 5 - (n-1) - (n-1) + d(x) = d(x) - 3.$$

Since this counts a set of edges, $\delta(G) \geq 3$.

Claim 3. The subgraph obtained by deleting the first call and the last call made by each vertex has at least five components and has no isolated vertex. Let xy be the first call involving x. If the first call involving y is yz with $z \neq x$, then by definition it occurs before xy, and these two calls do not communicate from x to z. After yz and xy, an increasing x, z-path violates NOHO at z. Hence the set F of first calls is a matching, and there are $n/2$ of them. Similarly, the set L of last calls is a matching of size $n/2$. The graph $G - F - L$ has $n - 5$ edges and hence at least five components, by Proposition 1.2.11. It has no isolated vertex, since $\delta(G) \geq 3$.

The contradiction. Since $e(G) = 2n - 5 < 2n$, some vertex x has degree at most 3. Let C_1, C_2, C_3 be the components of $G - F - L$ containing x, its first neighbor, and its last neighbor, respectively (its middle neighbor is in C_1). Edges of $G - F - L$ can belong to $O(x)$ only via paths that start with the first or middle edge involving x and avoid $F \cup L$, so they belong to C_1 or C_2. Similarly, edges of $G - F - L$ belonging to $I(x)$ appear only in C_1 or C_3. The edges of the remaining components, of which there are at least two, are useless to x ($G - F - L$ has no isolated vertex), but Claim 3 allows only $d(x) - 3 = 0$ edges useless to x. ■

In practical applications, we might wish to minimize the total length of the messages or the total time (assuming that each vertex participates in at most one call per time unit). We can also restrict the pairs that are allowed to call each other. Gossiping can be completed in $2n - 4$ if and only if the graph of allowable calls is connected and has a 4-cycle (Bumby [1981], Kleitman–Shearer [1980]). Other variations consider digraphs (Exercises 15–16), fault-tolerance, conference calls, etc.

LIST COLORING AND CHOOSABILITY

List coloring is a more general version of the vertex coloring problem. We still pick a single color for each vertex, but the set of colors available at each vertex may be restricted. This model was introduced independently in Vizing [1976] and Erdős–Rubin–Taylor [1979].

8.4.24. Definition. For each vertex v in a graph G, let $L(v)$ denote a list of colors available at v. A **list coloring** or **choice function** is a proper coloring f such that $f(v) \in L(v)$ for all v. A graph G is k-**choosable** or **list** k-**colorable** if every assignment of k-element lists to the vertices permits a proper list coloring. The **list chromatic number**, **choice number**, or **choosability** $\chi_l(G)$ is the minimum k such that G is k-choosable.

Since the lists could be identical, $\chi_l(G) \geq \chi(G)$. If the lists have size at least $1 + \Delta(G)$, then coloring the vertices in succession leaves an available color at each vertex. This argument is analogous to the greedy coloring algorithm and proves that $\chi_l(G) \leq 1 + \Delta(G)$ (see Exercise 22 for other analogues with

$\chi(G)$). It is not possible, however, to place an upper bound on $\chi_l(G)$ in terms of $\chi(G)$; there are bipartite graphs with arbitrarily large list chromatic number.

8.4.25. Proposition. (Erdős–Rubin–Taylor [1979]) If $m = \binom{2k-1}{k}$, then $K_{m,m}$ is not k-choosable.

Proof: Let X, Y be the bipartition of $G = K_{m,m}$. Assign the distinct k-subsets of $[2k-1]$ as the lists for the vertices of X, and do the same for Y. Consider a choice function f. If f uses fewer than k distinct choices in X, then there is a k-set $S \subseteq [2k-1]$ not used, which means that no color was chosen for the vertex of X having S as its list. If f uses at least k colors on vertices of X, then there is a k-set $S \subseteq [2k-1]$ of colors used in X, and no color can be properly chosen for the vertex of Y with list S. ■

List chromatic number is more difficult to compute than chromatic number, because the statements of the upper bound and lower bound both involve universal quantifiers. Determining the 3-choosable complete bipartite graphs was difficult. For $3 \le m \le n$, $K_{m,n}$ is 3-choosable if and only if

$m = 3$ and $n \le 26$ (Erdős–Rubin–Taylor [1979]), or
$m = 4$ and $n \le 20$ (Mahadev–Roberts–Santhanakrishnan [1991]), or
$m = 5$ and $n \le 12$ (Shende–Tesman [1994]), or
$m = 6$ and $n \le 10$ (O'Donnell [1995]).

Alon and Tarsi [1992] used a polynomial associated with a graph to obtain upper bounds on $\chi_l(G)$ (see also Alon [1993]). Fleischner and Stiebitz [1992] used the technique to prove that adding n disjoint triangles to a $3n$-cycle yields a 3-colorable graph; they proved the stronger result that it is 3-choosable.

There is also an edge-coloring variant, where we assign lists to the edges and must choose a proper edge-coloring.

8.4.26. Definition. Let $L(e)$ denote the list of colors available for e. A **list edge-coloring** is a proper edge-coloring f with $f(e)$ chosen from $L(e)$ for each e. The **edge-choosability** $\chi'_l(G)$ is the minimum k such that every assignment of lists of size k yields a proper list edge-coloring. Equivalently, $\chi'_l(G) = \chi_l(L(G))$, where $L(G)$ is the line graph of G.

The argument for $\chi'(G) \le 2\Delta(G) - 1$ also yields $\chi'_l(G) \le 2\Delta(G) - 1$ (Exercise 22) and thus $\chi'_l(G) < 2\chi'(G)$. As in ordinary coloring, the use of line graphs expresses the edge version as a special case of the vertex version, and it behaves much better. Even so, the conjectured bound for edge-choosability is surprising. It was suggested independently by many researchers, including Vizing, Gupta, Albertson, Collins, and Tucker, and it seems to have been published first in Bollobás–Harris [1985] (see also Bollobás [1986]).

8.4.27. Conjecture. (List Coloring Conjecture) $\chi'_l(G) = \chi'(G)$ for all G. ■

For simple graphs, this conjecture and Vizing's Theorem (Theorem 7.1.10) would yield $\chi'_l(G) \le \Delta(G) + 1$. Bollobás and Harris [1985] proved that $\chi'_l(G) <$

$c\Delta(G)$ when $c > 11/6$ for sufficiently large $\Delta(G)$. This and later improvements used probabilistic methods. Kahn [1996] proved the conjecture asymptotically: $\chi'_l(G) \le (1 + o(1))\Delta(G)$. Häggkvist and Janssen [1997] sharpened the error term: $\chi'_l(G) \le d + O(d^{2/3}\sqrt{\log d})$ when $d = \Delta(G)$. Molloy and Reed [1999] further sharpened (and generalized) the bound.

The special case of the List Coloring Conjecture for $G = K_{n,n}$ was posed by Dinitz in 1979. (Janssen [1993] proved it for $K_{n,n-1}$.) The Dinitz Conjecture became popular in its matrix formulation: If each position of an n by n grid contains a set of size n, then it is possible to choose one element from each set so that the elements chosen in each row are distinct and the elements chosen in each column are distinct.

Galvin [1995] proved the List Coloring Conjecture for bipartite graphs, which includes the Dinitz Conjecture (see also Slivnik [1996]). Here we prove only the Dinitz Conjecture, using the Stable Matching Problem (Section 3.2).

8.4.28. Definition. A **kernel** of a digraph is an independent set S having a successor of every vertex outside S. A digraph is **kernel-perfect** if every induced subdigraph has a kernel. Given a function $f\colon V(G) \to \mathbb{N}$, the graph G is f**-choosable** if a proper coloring can be chosen from the lists at the vertices whenever $|L(x)| = f(x)$ for each x.

We used the concept of "kernel" in Application 1.4.14 (digraphs without odd cycles, for example, have kernels). An f-choosable graph is k-choosable for $k = \max f(x)$, since adding colors to a list cannot make the choice more difficult.

8.4.29. Lemma. (Bondy–Boppana–Siegel) If D is a kernel-perfect orientation of G and $f(x) = 1 + d_D^+(x)$ for all $x \in V(G)$, then G is f-choosable.

Proof: We use induction on $n(G)$; the statement is trivial for $n(G) = 1$. For $n(G) > 1$, consider an assignment of lists, with the list $L(x)$ having size $f(x)$. Choose a color c appearing in some list. Let $U = \{v\colon c \in L(v)\}$. Let S be the kernel of the induced subdigraph $D[U]$. Assign color c to all of S, which is permissible since S is independent.

Delete c from $L(v)$ for each $v \in U - S$. Delete additional colors arbitrarily from other lists to reduce $L(x)$ for each $x \in V(D) - S$ to size $f'(x)$, where $f'(x) = 1 + d_{D-S}^+(x)$. Since each vertex not in S has a successor in S, we have $f'(x) < f(x)$ for $x \in V(D) - S$, which accommodates the deletion of c from the lists. By the induction hypothesis, D' is $f'(x)$-choosable, so we can complete a list coloring for G by adding a list coloring of D' to the use of c on S. ■

8.4.30. Theorem. (Galvin [1995]) $\chi'_l(K_{n,n}) = n$.

Proof: Since $\chi'_l(G) = \chi_l(L(G))$, it suffices by Lemma 8.4.29 to prove that $L(K_{n,n})$ has a kernel-perfect orientation with each vertex having indegree and outdegree $n-1$. The graph $L(K_{n,n})$ is the cartesian product $K_n \,\square\, K_n$ (Exercise 7.1.8); placed in an n by n grid, vertices are adjacent if and only if they are in the same row or in the same column.

Assign labels $1, 2, \ldots, n$ so that vertex (r, s) has label $r+s-1 \bmod n$. Define an orientation D of $K_n \square K_n$ by directing edges from vertex (r, s) with label i to the vertices in column s with lower labels and the vertices in row r with higher labels. Since i is higher than $i-1$ other labels, (r, s) has $i-1$ successors in its column and $n-i$ successors in its row. Hence $d^+(r, s) = d^-(r, s) = n-1$.

We prove that D is kernel-perfect. Given $U \subseteq V(D)$, we obtain a kernel for the subdigraph $D[U]$ by solving a stable matching problem. When $(r, b) \in U$ and $(r, s) \to (r, b)$ in D, we want r to prefer b to s. Thus for row r, the preferences among columns begin with $\{s\colon (r, s) \in U\}$ in decreasing order of vertex labels, followed by any order among $\{s\colon (r, s) \notin U\}$. Similarly, for column s, the preferences among rows begin with $\{r\colon (r, s) \in U\}$ in increasing order of vertex labels, followed by any order among $\{r\colon (r, s) \notin U\}$.

The Gale–Shapley Proposal Algorithm (Algorithm 3.2.17) yields a stable matching M for these preferences. Viewing the matched pairs in M as positions in the grid, let $S = M \cap U$. Because M is a matching, S has no two positions in the same row or column; hence S is an independent set in D. We show that each $x \in U - S$ has a successor in S.

Let i be the label of position $x = (r, s) \in U - S$. Since $S = M \cap U$, we have $x \notin M$. Thus M has a position $y = (r, b)$ with some label j and a position $z = (a, s)$ with some label k. Because M is stable, we cannot have both r preferring s to b and s preferring r to a. From this statement we deduce by the steps below that x has y or z as a successor in S. ■

$$\begin{array}{c|cc|} & b & s \\ \hline a & & z:k \\ & & \\ r & y:j & x:i \\ \hline \end{array}$$

$$\begin{array}{ll} \text{not} & \big[(r \text{ prefers } s \text{ to } b) \;\text{ and }\; (s \text{ prefers } r \text{ to } a)\big] \\ \text{not} & \big[(y \notin U \;\text{ or }\; i > j) \;\text{ and }\; (z \notin U \;\text{ or }\; i < k)\big] \\ & (y \in U \;\text{ and }\; i < j) \;\text{ or }\; (z \in U \;\text{ and }\; i > k) \\ & \qquad (x \to y \in S) \;\text{ or }\; (x \to z \in S) \end{array}$$

8.4.31. Remark. The List Coloring Conjecture relates to another conjecture. A **total coloring** of G assigns a color to each vertex and to each edge so that colored objects have different colors when they are adjacent or incident. The Total Coloring Conjecture (Behzad [1965]) states that every simple graph G has a total coloring with at most $\Delta(G)+2$ colors. Rosenfeld [1971] and Behzad [1971] provide results on special classes. The List Coloring Conjecture would yield an upper bound of $\Delta(G)+3$, since every graph G has a total coloring with at most $\chi_l'(G)+2$ colors (Exercise 25). ■

The List Coloring Conjecture has been studied for planar graphs. Ellingham and Goddyn [1996] proved that every k-regular k-edge-colorable planar graph is k-edge-choosable (using the Four Color Theorem).

The discussion of planar graphs brings us back to list coloring of vertices. Although planar graphs have chromatic number at most 4, Vizing [1976] and

Erdős–Rubin–Taylor [1979] conjectured that the maximum choice number on this class is 5. Voigt [1993] constructed a non-4-choosable planar graph with 238 vertices; Mirzakhani [1996] (Exercise 26) reduced this to 63 vertices (both examples generalize to infinite families). In fact, there are 3-colorable planar graphs that are not 4-choosable (Gutner [1996], Voigt–Wirth [1997]).

Thomassen [1994b] proved the upper bound (and also [1995] that planar graphs of girth 5 are 3-choosable). Often in inductive proofs for planar graphs, the vertices on the unbounded face ("external vertices") play a special role.

8.4.32. Theorem. (Thomassen [1994b]) Planar graphs are 5-choosable.

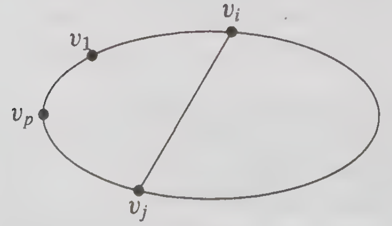
Proof: Adding edges cannot reduce the list chromatic number, so we may restrict our attention to plane graphs where the outer face is a cycle and every bounded face is a triangle. By induction on $n(G)$, we prove the stronger result that a coloring can be chosen even when two adjacent external vertices have distinct lists of size 1 and the other external vertices have lists of size 3. For the basis step ($n = 3$), a color remains available for the third vertex.

Now consider $n > 3$. Let v_p, v_1 be the vertices with fixed colors on the external cycle C. Let $v_1, \ldots, v_p$ be $V(C)$ in clockwise order.

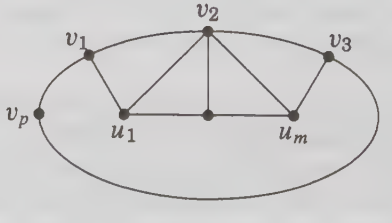
Case 1: C has a chord $v_i v_j$ with $1 \le i \le j-2 \le p-2$. We apply the induction hypothesis to the graph consisting of the cycle $v_1, \ldots, v_i, v_j, \ldots, v_p$ and its interior. This selects a proper coloring in which v_i, v_j receive some fixed colors. Next we apply the induction hypothesis to the graph consisting of the cycle $v_i, v_{i+1}, \ldots, v_j$ and its interior to complete the list coloring of G.

Case 2: C has no chord. Let $v_1, u_1, \ldots, u_m, v_3$ be the neighbors of v_2 in order ($3 = p$ is possible). Because bounded faces are triangles, G contains the path P with vertices $v_1, u_1, \ldots, u_m, v_3$. Since C is chordless, $u_1, \ldots, u_m$ are internal vertices, and the outer face of $G' = G - v_2$ is bounded by a cycle C' in which P replaces v_1, v_2, v_3.

Let c be the color assigned to v_1. Since $|L(v_2)| \ge 3$, we may choose distinct colors $x, y \in L(v_2) - \{c\}$. We reserve x, y for possible use on v_2 by forbidding x, y from $u_1, \ldots, u_m$. Since $|L(u_i)| \ge 5$, we have $|L(u_i) - \{x, y\}| \ge 3$. Hence we can apply the induction hypothesis to G', with $u_1, \ldots, u_m$ having lists of size at least 3 and other vertices having the same lists as in G. In the resulting coloring, v_1 and $u_1, \ldots, u_m$ have colors outside $\{x, y\}$. We extend this coloring to G by choosing for v_2 a color in $\{x, y\}$ that does not appear on v_3 in the coloring of G'. ■



Case 1



Case 2

PARTITIONS USING PATHS AND CYCLES

We have considered the **F-decomposition** problem: partitioning $E(G)$ into the minimum number of subgraphs in a family **F**. This has been studied for many families **F**, such as cliques (Theorem 8.4.3), bipartite graphs (Exercise 3), complete bipartite graphs (Theorem 8.6.20), stars (vertex cover number—Section 3.1), and forests (arboricity—Corollary 8.2.57). Before considering extremal problems for decomposition of graphs into paths and cycles, we discuss an easier problem: covering the vertices of a digraph using the fewest paths.

Comparability graphs are those having transitive orientations; a digraph is **transitive** if $x \to y$ and $y \to z$ imply $x \to z$. The vertices of a path in a transitive digraph induce a tournament. Comparability graphs are perfect (Proposition 5.3.25), meaning that a transitive digraph D in which the largest tournament has ω vertices can be properly ω-colored. By the Perfect Graph Theorem (Theorem 8.1.6), we also know that $V(D)$ can be covered using $\alpha(D)$ tournaments in D, where $\alpha(D)$ is the maximum size of an independent set.

Letting paths be "chains" and independent sets be "antichains", this becomes Dilworth's Theorem for transitive loopless digraphs: The maximum size of an antichain equals the minimum number of chains needed to partition $V(D)$. In addition to following from the Perfect Graph Theorem, Dilworth's Theorem is equivalent to the König–Egerváry Theorem (Exercise 27), and a generalization of it follows from the Matroid Intersection Theorem (Exercise 8.2.50). Here we present a further generalization that has a short and self-contained proof.

8.4.33. Theorem. (Gallai–Milgram [1960]) The vertices of a digraph D can be covered using at most $\alpha(D)$ pairwise disjoint paths.

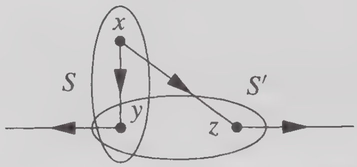
Proof: Since $V(D)$ can be covered using n disjoint paths of length 0, it suffices to prove a stronger claim: If **C** is a set of pairwise disjoint paths covering $V(D)$, and S is the set of sources (initial vertices) of these paths, then $V(D)$ can be covered using at most $\alpha(D)$ pairwise disjoint paths with sources in S. The proof is by induction on $n(D)$, with a trivial basis step for $n(D) = 1$. The added statement about the sources helps the induction step work.

Suppose that $n > 1$ and that **C** is a covering of $V(D)$ by k paths with source set S. The claim holds unless $|\mathbf{C}| = k > \alpha(D)$, in which case we construct a cover using fewer paths, all with sources in S. Since $k > \alpha$, there exists an edge xy with $x, y \in S$. Let A and B be the paths in **C** starting with x and y, respectively. We may assume that A has an edge xz, else we could add x to the beginning of B and save one path.

By deleting x from the start of A, we obtain a cover $\mathbf{C}'$ of $V(D - x)$ by k paths having sources in $S' = S - x + z$. Since $\alpha(D - x) \le \alpha(D)$, the induction hypothesis yields a cover $\mathbf{C}''$ of $V(D - x)$ using fewer than k paths, all with sources in S'. All elements of S' belong to S except z.

If z is the source of a path in $\mathbf{C}''$, then we add x at the beginning of that path. If z is not a source but y is, then we add x at the beginning of the path starting with y. If neither y nor z is a source, then at most $|S'| - 2 = k - 2$ paths have been used, and we can add x as a path by itself to obtain the desired cover of $V(D)$ using $k-1$ paths. In all cases, the resulting paths are pairwise disjoint and have sources in S.

By repeating this argument as long as $k > \alpha$, we can reduce the number of paths to α. ∎



We return to the decomposition problem. Gallai conjectured that every n-vertex graph can be decomposed using $\lceil n/2 \rceil$ paths. Equality holds for cliques (Exercise 28). Other graphs have fewer edges, but the lack of connections could require more paths. Hajós conjectured analogously that an n-vertex even graph can be decomposed into $\lfloor n/2 \rfloor$ cycles. Both conjectures remain open, but Lovász proved the optimal bound when both paths and cycles are allowed. The **size** of a decomposition is the number of subgraphs used.

8.4.34. Theorem. (Lovász [1968b]) Every n-vertex graph can be decomposed into $\lfloor n/2 \rfloor$ paths and cycles.

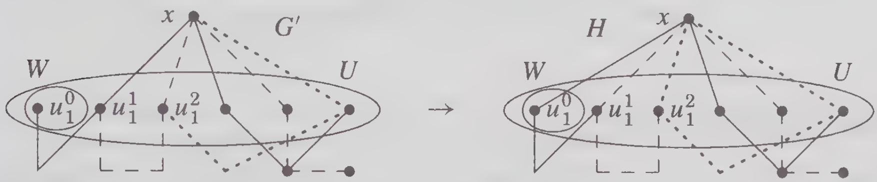
Proof: Let **F** be the family of all paths and cycles, and let $n'(G)$ be the number of non-isolated vertices in a graph G. By induction on $\lambda(G) = 2e(G) - n'(G)$, we prove that G has an **F**-decomposition of size at most $\lfloor n'(G)/2 \rfloor$. Each component of G with more than one edge contributes positively to $\lambda(G)$. Hence $\lambda(G) \geq 0$, with equality only when each nontrivial component is an edge. The claim holds with equality when $\lambda(G) = 0$.

In the induction step, $\lambda(G) > 0$. We consider two cases. **Case 1**: If G has a vertex y of positive even degree, choose $x \in N(y)$, and let $W = \{z \in N(x)\colon d(z)$ is even$\}$. In this case, let $G' = G - \{xz : z \in W\}$. In obtaining G', we lose at least one edge (xy) and we isolate at most one vertex (x), so $\lambda(G') < \lambda(G)$. **Case 2**: If G has no vertex of positive even degree, then $\lambda(G) > 0$ forces $\Delta(G) > 1$. Let x be a vertex of degree at least 3, and form G^+ by introducing a new vertex y to subdivide an edge xx'. Let $W = \{y\}$, and let $G' = G^+ - xy$. Now $e(G') = e(G)$, but $n'(G') > n'(G)$, so $\lambda(G') < \lambda(G)$.

In each case, the induction hypothesis yields an **F**-decomposition **D** of G' with $|\mathbf{D}| \leq \lfloor n'(G')/2 \rfloor$. We convert **D** into an **F**-decomposition of size $|\mathbf{D}|$ for the graph H obtained from G' by adding edges from x to W. In Case 1, $H = G$ and $n'(G') \leq n'(G)$, so this is the desired decomposition. In Case 2, $H = G^+$ and $n'(G') = n'(G^+)$. Since $n'(G)$ is even, $\lfloor n'(G)/2 \rfloor = \lfloor n'(G^+)/2 \rfloor$. In an **F**-decomposition of G^+, the $n'(G)$ vertices of odd degree must all be endpoints of paths; thus the added vertex y of degree 2 cannot be the end of a path. This means that xy and yx' belong to the same subgraph and can be replaced by xx' to obtain the desired decomposition of G.

The two cases now combine; we need only obtain the decomposition of H from **D**. Let $U = N_H(x)$. Every vertex of U has odd degree in G', so for each

$u \in U$ there is a path $P(u)$ in $\mathbf{D}$ with endpoint u. For $u \in W$, we would like to extend $P(u)$ to absorb ux. This cannot be done if $P(u)$ reaches but does not end at x, since then the subgraph $P(u) \cup ux$ is not in $\mathbf{F}$. The idea is to cut the edge $u'x$ on which $P(u)$ reaches x, use the path $P(u) \cup ux - u'x$, and use $u'x$ to extend $P(u')$ instead. This generates a sequence of changes from each $u \in W$. We must show that the sequences terminate and do not conflict with each other.

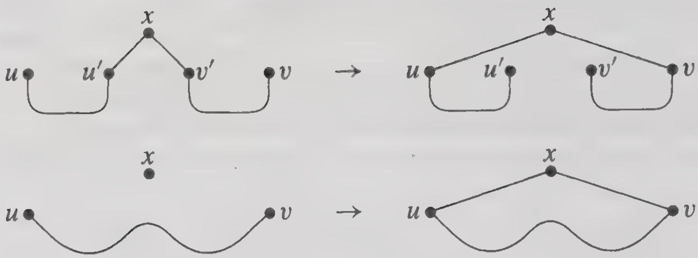


Let $W = w_1, \dots, w_t$. For $w_i \in W$, we form a list $u_i^0, u_i^2, \cdots$ with $u_i^0 = w_i$ and each $u_i^j \in U$. If in the ith list we have chosen a vertex u_i^j, we check whether x is an internal vertex of $P(u_i^j)$. If not, then we stop and do not define u_i^{j+1}. If so, then we set u_i^{j+1} to be the vertex on $P(u_i^j)$ just before x; this is the "u'" suggested above. The path $P(u_i^j)$ for $j \geq 1$ cannot start along the edge $u_i^j x$, because that edge is internal to $P(u_i^{j-1})$. (Our picture of G' shows three successive paths: $P(u_1^0)$ solid, $P(u_1^1)$ dashed, $P(u_1^2)$ dotted.)

We prove next that no vertex of U appears twice in the lists. Since $xu_i^j \in E(G')$ if $j \geq 1$, the vertices of W appear only as initial vertices. Let u_i^j, u_k^l be a repeated vertex with $\min\{j, l\}$ minimal; we have shown that $j, l > 0$. By minimality, $u_i^{j-1} \neq u_k^{l-1}$, and hence the paths $P(u_i^{j-1})$ and $P(u_k^{l-1})$ start at distinct vertices. If $u_i^j = u_k^l$, then the two paths share the edge $u_i^j x$ and must be the same path. This happens from distinct vertices only if u_i^{j-1} and u_k^{l-1} are opposite ends of the path, but then they cannot both visit u_i^j before x. Hence no repetition occurs.

Let $W' = \{u_i^j\}$. If $u = u_i^j$ and u is not the end of its list, let $u' = u_i^{j+1}$. We define an $\mathbf{F}$-decomposition of G consisting of one path or cycle Q' corresponding to each $Q \in \mathbf{D}$. If $Q \neq P(u)$ for some $u \in W'$, let $Q' = Q$. If $Q = P(u)$, let $Q' = Q + ux$ or $Q' = Q + ux - u'x$ depending on whether u is or is not the last vertex in its list. Always Q' is a path, except that Q' is a cycle when Q ends at x (and then u' is not defined). The union of the new paths corresponding to $\{P(u_i^j)\}$ is the same as $\cup P(u_i^j)$, except that the edges $\{xw_i\}$ are absorbed. Since $u \in W'$ appears only once in the lists, the edge ux winds up in only one of the new paths, and $\{Q' \colon Q \in \mathbf{D}\}$ is a decomposition of H. ∎

Note that in this proof Q may be the selected path from each of its endpoints $u, v \in W'$. This is not a problem, because the adjustments to Q made from the two ends do not conflict. The path may visit x (thus defining u' and v') or not, as sketched below.



CIRCUMFERENCE

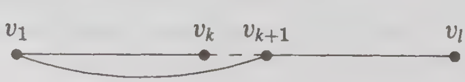
When a sufficient condition for Hamiltonian cycles fails slightly, we might expect that the graph still must have a fairly long cycle. The length of the longest cycle in G is the **circumference** $c(G)$. We first consider the number of edges needed to force a cycle of length at least c in an n-vertex graph. In this section, $P(v, w)$ denotes the v, w-portion of a path P containing v and w. Also P, Q denotes the concatenation of paths P and Q when the last vertex of P is the first vertex of Q.

8.4.35. Theorem. (Erdős–Gallai [1959]) For $m \geq 2$, every simple n-vertex graph with more than $m(n-1)/2$ edges has a cycle of length more than m.

Proof: (Woodall [1972]) We use induction on n for fixed m. When $n = m + 1$, fewer than $(n-1)/2$ edges are missing, so $\delta(G) \geq n/2$ and G is Hamiltonian. Suppose that $n > m+1$ and $c(G) \leq m$. If $d(x) \leq m/2$, then $e(G-x) \geq m(n-2)/2$. Applying the induction hypothesis to $G - x$ yields $c(G - x) > m$. Hence we may assume that $\delta(G) > m/2$. Similarly, we may assume that G is connected.

Among all longest paths in G, choose $P = v_1, \ldots, v_l$ to maximize the degree d of v_1; since G is connected, we have $v_1 \not\leftrightarrow v_l$ (otherwise an edge from $V(P)$ to $V(G) - V(P)$ would yield a longer cycle). Let $W = \{v_i\colon v_1 \leftrightarrow v_{i+1}\}$. All neighbors of v_1 lie on P, so $|W| = d$. For $v_k \in W$, the path $P(v_k, v_1), v_1v_{k+1}, P(v_{k+1}, v_l)$ also has length l; hence $N(v_k) \subseteq V(P)$, and the choice of P yields $d(v_k) \leq d$. Furthermore, no $v_k \in W$ has a neighbor v_j such that $j > m$, because then we could complete the long cycle by adding v_jv_k to $P(v_k, v_1), v_1v_{k+1}, P(v_{k+1}, v_j)$.

By limiting the edges incident to W, we force many edges into $G - W$. Let $Z = \{v_1, \ldots, v_r\}$, where $r = \min\{l, m\}$. For each $v_k \in W$, we have shown that $N(v_k) \subseteq Z$. Hence there are $|[W, Z - W]| + e(G[W])$ edges incident to W. For fixed degree-sum in W, this is maximized when $[W, Z - W]$ is a complete bipartite graph. We further maximize by letting each vertex of W have degree d. The resulting count is $\frac{1}{2}|W|(d+|Z - W|) = dr/2 \leq dm/2$. Therefore, $G - W$ has $n - d$ vertices and more than $m(n-d-1)/2$ edges. By the induction hypothesis $c(G - W) > m$. (If the number of edges forced into $G - W$ is too large to exist, then this case cannot occur, and an earlier case applies.) ■



Most sufficient conditions for Hamiltonian cycles have "long cycle" versions. The long cycle version of Dirac's Theorem says that a 2-connected graph G has a cycle of length at least $\min\{n(G), 2\delta(G)\}$ (Dirac [1952b]). Requiring 2-connectedness eliminates the example $K_1 \vee 2K_\delta$ with circumference $\delta + 1$.

The long cycle version of Ore's Theorem [1960] came much later. It is implicit in Bondy [1971b] and was made explicit in Bermond [1976] and in Linial [1976]. The fundamental argument used in many long cycle results appears in Bondy [1971b]. It strengthens the Ore/Dirac switching argument (Theorem 7.2.8) by considering "gaps".

8.4.36. Lemma. (Bondy [1971b]) If $P = v_1, \ldots, v_l$ is a longest path in a 2-connected graph G, then $c(G) \ge \min\{n(G), d(v_1) + d(v_l)\}$.

Proof: (See also Linial [1976]). Let $m = d(v_1) + d(v_l)$, and suppose that $c(G) < \min\{n(G), m\}$. Since G is connected, an l-cycle would yield a longer path; thus $v_1 \not\leftrightarrow v_l$. If $v_1 \leftrightarrow v_j$ and $v_i \leftrightarrow v_l$ for some $i < j$, then i, j is a *crossover* with *gap* $j - i$. If we add $v_1 v_j$ and $v_l v_i$ to $P(j, l)$ and $P(i, 1)$, we obtain a cycle with length $l - (j - i - 1)$. Hence $l - (j - i - 1) < m$ when i, j is a crossover.

Let $x = v_1$ and $y = v_l$. If P has a crossover, let i, j be one with smallest gap. Thus x and y have no neighbors between v_i and v_j on P. Also $N(y)$ contains no predecessor on P of a neighbor of x, since an l-cycle yields a longer path. Hence $N(y)$ lies in $V(P) - \{y\}$ but avoids $\{v_{i+1}, \ldots, v_{j-2}\}$ and $\{v_{r-1} : v_r \leftrightarrow x\}$. Thus $d(y) \le (l - 1) - (j - 2 - i) - d(x)$. Since $l - (j - i - 1) < m$, we have $d(x) + d(y) < m$, which contradicts the hypothesis. Hence there is no crossover.

With $t_0 = \max\{i : x \leftrightarrow v_i\}$ and $u = \min\{i : y \leftrightarrow v_i\}$, we have proved that $t_0 \le u$. We will construct a cycle containing x and y and all their neighbors. Since the absence of crossovers implies that $|N(x) \cap N(y)| \le 1$, such a cycle has length at least $d(x) + d(y) + 1 > m$.

We iteratively define paths $P_1, P_2, \cdots$. Given t_{i-1}, we choose integers $s_i < t_{i-1} < t_i$ to maximize t_i such that G has a v_{s_i}, v_{t_i}-path P_i internally disjoint from P. Such a path exists because $G - v_{t-1}$ is connected. These paths are disjoint; if P_i shares a vertex with a later path P_j, then we can choose P_i as an s_i, t_j-path, which contradicts the maximality of t_i. Similarly, $s_{i+1} \ge t_{i-1}$, since otherwise P_{i+1} would be chosen instead of P_i.

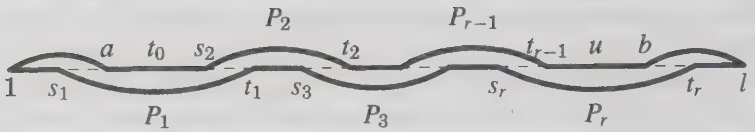
Let r be the smallest index such that $t_r > u$. Set

$$a = \min\{j : x \leftrightarrow v_j \text{ and } j > s_1\}, \qquad b = \max\{j : y \leftrightarrow v_j \text{ and } j < t_r\}.$$

Since $s_1 < t_0$ and $t_r > u$, the indices a, b are well-defined. We use the even-indexed paths P_i to build one x, y-path and the odd-indexed paths to build another x, y-path. When r is odd, the two paths are formed by the following concatenations.

$$x v_a,\ P(a, s_2),\ P_2,\ P(t_2, s_4),\ P_4,\ \ldots,\ P(t_{r-1}, b),\ v_b y$$

$$P(1, s_1),\ P_1,\ P(t_1, s_3),\ P_3,\ P(t_3, s_5),\ \ldots,\ P_r,\ P(t_r, l)$$



When r is even, the path starting with xv_a reaches t_r and ends with $P(t_r, l)$, while the other path reaches v_b and ends with $v_b y$.

We have observed that $s_{i+1} \geq t_{i-1}$. Hence

$$s_1 < a \leq t_0 \leq s_2 < t_1 \leq s_3 < t_2 \cdots < t_{r-1} \leq u \leq b < t_r$$

This implies that the two concatenations described are paths and that their union is a cycle. By the definition of a, we have $N(x) \subseteq P(1, s_1) \cup P(a, t_0)$, and similarly $N(y) \subseteq P(u, b) \cup P(t_r, l)$. With x and y themselves, the cycle thus has length at least $2 + d(x) + d(y) - 1 > m$. ■

Ore proved that G is Hamiltonian if $d(u) + d(v) \geq n(G)$ when $u \not\leftrightarrow v$. Bondy's Lemma implies the long cycle version of this, which strengthens the long cycle version of Dirac's Theorem.

8.4.37. Theorem. (Bondy [1971b], Bermond [1976], Linial [1976]) If G is 2-connected and $d(u) + d(v) \geq s$ for every nonadjacent pair $u, v \in V(G)$, then $c(G) \geq \min\{n(G), s\}$.

Proof: Ore's Theorem guarantees a Hamiltonian cycle if $s \geq n$, so we may assume that $s < n$. Suppose that P is a longest path in G, with endpoints x and y. Since G is connected, the maximality of P implies that $x \not\leftrightarrow y$. Now the condition $d(x) + d(y) \geq s$ allows us to invoke Lemma 8.4.36. ■

Bermond extended this to a "long cycle" combination of Chvátal's condition and Las Vergnas' condition. The technique of edge-switches involving an endpoint of a longest path was used in Theorem 8.4.35. Our statement is slightly weaker than that of Bermond but has a simpler proof.

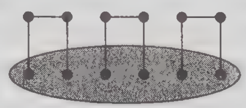
8.4.38. Theorem. (Bermond [1976]) Let G be a 2-connected graph with degree sequence $d_1 \leq \cdots \leq d_n$. If G has no nonadjacent pair x, y with degrees i, j such that $d_i \leq i < c/2$, $d_{j+1} \leq j$, and $i + j < c$, then $c(G) \geq c$.

Proof: Among the longest paths in G, let $P = v_1, \ldots, v_l$ with endpoints $x = v_1$ and $y = v_l$ be chosen to maximize $d(v_1) + d(v_l)$. If $d(x) + d(y) \geq c$, then we apply Bondy's Lemma. If $d(x) + d(y) < c$, then we claim that x, y contradicts the hypotheses. As usual, an l-cycle would yield a longer path (since G is connected), so $x \not\leftrightarrow y$. We may assume that $d(x) \leq d(y)$ and set $i = d(x)$ and $j = d(y)$.

All neighbors of x and y lie in P. If $x \leftrightarrow v_k$, then $P(v_{k-1}, x)$, xv_k, $P(v_k, y)$ is another longest path ending at y; thus $d(v_{k-1}) \leq d(x) = i$, by the choice of P. Since this holds for each of the i neighbors of x, we have $d_i \leq i$. Similarly, the j neighbors of y each have degree at most j. Also $d(x) \leq j$, so $d_{j+1} \leq j$. By hypothesis, $i + j = d(x) + d(y) < c$, which completes the contradiction. ■

G.-H. Fan [1984] strengthened Theorem 8.4.37 by weakening the degree condition and by requiring it only for nonadjacent pairs with common neighbors. T. Feng [1988] used Bondy's Lemma to shorten the proof. The result includes a sufficient condition for Hamiltonian cycles that does not require the closure to be complete.

8.4.39. Example. *A Hamiltonian graph.* For even n, let $G_1 = K_{n/2}$ and $G_2 = (n/4)K_2$, and form G by adding a matching between disjoint copies of G_1 and G_2. The Hamiltonian closure of G is G itself, so our previous sufficient conditions do not apply. Even though G has $n/2$ vertices of degree 2, Fan's Theorem implies that G is Hamiltonian. ■

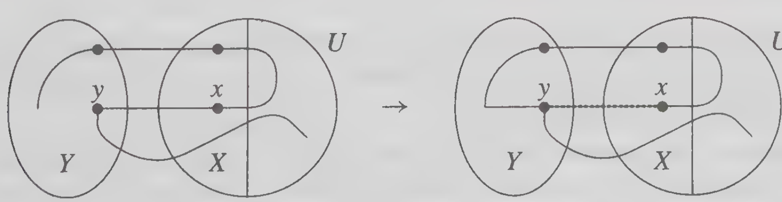


8.4.40. Theorem. (Fan [1984]) If G is 2-connected, and $d_G(u, v) = 2$ implies $\max\{d(u), d(v)\} \geq c/2$, then $c(G) \geq \min\{n(G), c\}$.

Proof: (Feng [1988]) Let $U = \{v \in V(G) \colon d(v) \geq c/2\}$. By Bondy's Lemma, it suffices to find a longest path having both endpoints in U. Among the paths of maximum length, let $P = v_1, \ldots, v_m$ be one that has the maximum number of endpoints in U. If P fails to have both endpoints in U, then we will find a longer path or a path of the same length with more of its endpoints in U. We may assume that $v_1 \notin U$.

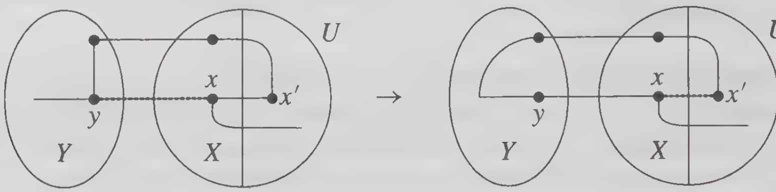
Since $d(v) < c/2$ for all $v \notin U$, the hypothesis on pairs with distance 2 implies that $G - U$ is a disjoint union of complete graphs. Let Y be the one containing v_1. Let X be the set of vertices in U having neighbors in Y. By the hypotheses, vertices of X have neighbors only in $Y \cup U$. Also $|X| \geq 2$, because G is 2-connected.

Let $r = |Y|$. We first show that P begins by visiting all of Y. If P omits some vertex of Y, then we can absorb it before the first exit from Y. If P leaves and returns to Y, then it returns via an edge xy. Because $G[Y]$ is complete, we can replace xy in P with v_1y, obtaining an x, v_m-path having the same length as P but more endpoints in U. Hence we may assume that $Y = \{v_1, \ldots, v_r\}$.



Consider $x \in X - v_{r+1}$. Suppose first that x has a neighbor $y \in Y$ other than the exit vertex v_r of P. If $x \notin V(P)$, then we can instead start with xy, absorb the rest of Y up to v_r, and thus complete an x, v_l-path longer than P. If

$x \in V(P)$, then we let x' be the vertex before x on P. Since $x \neq v_{r+1}$, we have $x' \in U$. We replace $x'x$ in P with yx, obtaining an x', v_l-path with the same length as P but more endpoints in U.



Hence we may assume for $x \in X - v_{r+1}$ that x has no neighbor in Y other than v_r. If $|Y| \geq 2$, this makes v_r a cut-vertex unless v_{r+1} has another neighbor $y \in Y - v_r$. Now we rearrange P to start with $v_r, \ldots, y, v_{r+1}$ instead of $v_1, \ldots, v_r, v_{r+1}$. This puts us in the case just discussed.

The remaining case is $|Y| = 1$ and $N(v_1) = X$. With $x \in X - v_{r+1}$ as before, we append x to the beginning of P or replace $x'x$ with xv_1. ■

Finally, we present one result about digraphs that strengthens Ghouilà-Houri's sufficient condition (Theorem 7.2.22) for Hamiltonian cycles. We consider only loopless digraphs having at most one copy of each ordered pair as an edge; call these **strict** digraphs. For digraphs, we use "u, v nonadjacent" to mean $uv, vu \notin E(G)$. Also, we define $d(v) = d^+(v) + d^-(v)$.

Ghouilà-Houri [1960] actually proved that a digraph G is Hamiltonian if $d(v) \geq n(G)$ for each v; this is stronger than Theorem 7.2.22 as stated. Woodall [1972] proved that it suffices to have $d^+(u) + d^-(v) \geq n(G)$ whenever u, v are nonadjacent. This generalizes Ore's Theorem for undirected graphs (Exercise 33). Meyniel [1973] proved that a strict strong digraph G is Hamiltonian if $d(u)+d(v) \geq 2n(G)-1$ for all nonadjacent pairs u, v. Meyniel's Theorem implies Ghouilà-Houri's Theorem and Woodall's Theorem (Exercise 33).

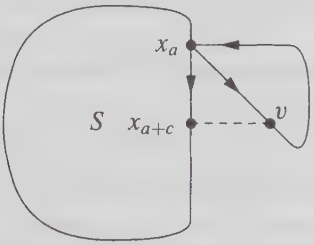
8.4.41. Example. *Meyniel's Theorem is best possible.* Let G consist of two doubly-directed cliques sharing a vertex. The digraph is strongly connected, and the only pairs of nonadjacent vertices consist of one vertex from each clique. If the cliques have order k and order $n + 1 - k$, then the total degrees for any nonadjacent pair are $2k - 2$ and $2n - 2k$, which sum to $2n - 2$. ■

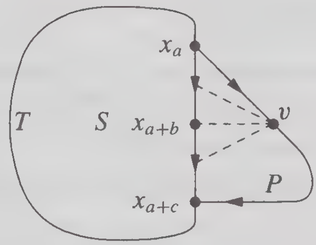
8.4.42. Theorem. (Meyniel [1973]) If G is a strict strongly connected digraph such that $d(u) + d(v) \geq 2n - 1$ whenever u, v are distinct nonadjacent vertices, then G is Hamiltonian.

Proof: (Bondy–Thomassen [1977]). We prove a technical lemma: if $T = v_1, \ldots, v_k$ is a path that cannot absorb the vertex v internally (between two of its vertices), then the number of edges from v to T plus the number of edges from T to v is at most $k+1$. This follows by counting. For $1 \leq i \leq k-1$, only one of the edges $v_i v$ and vv_{i+1} is permitted. Also vv_1 and $v_k v$ are permitted; there is no restriction on absorption at the end.

We use this to prove the following statement: If G is a strict strong non-Hamiltonian digraph, and S is a maximal vertex subset having a spanning cycle $(x_1, \dots, x_m)$ in G, then there exist $v \in \overline{S}$ and integers a, b with $1 \le a \le m$ and $1 \le b < m$ such that (1) $x_a v \in E(G)$, (2) v is not adjacent to any x_{a+i} with $1 \le i \le b$, and (3) $d(v) + d(x_{a+b}) \le 2n - 1 - b$. Since $b \ge 1$, the conclusion of this statement is impossible under the hypothesis of the theorem, which will imply that the only maximal vertex set having a spanning cycle is $V(G)$.

Suppose first that no path leaves S and returns to it. Since G is strong and $S \ne V(G)$, some cycle C of length at least 2 shares exactly one vertex with S. Let this vertex be x_a, and let v be the successor of x_a on C. By the path condition, there is no path between v and $S - \{x_a\}$ in either direction. In particular, each vertex outside $S \cup \{v\}$ is incident to at most two edges also incident to v or v_{a+1}. Furthermore, v is incident to at most two edges also incident to S (the other endpoint must be v_a). Finally, each vertex of $S - v_{a+1}$ is incident to at most two edges also incident to v_{a+1}. Summing the allowed contributions yields $d(v) + d(x_{a+1}) \le 2n - 2$. Hence the desired condition holds with $b = 1$.





Now suppose that some path leaves S and returns to it. Choose such a path P so that the distance c along S from the start of P to the end of P is minimal. Let x_a be the start of P, and let v be its successor on P. The maximality of S implies that $c > 1$. Let T be the portion of S from x_{a+c} to x_a; this has $m - c + 1$ vertices. The maximality of S implies that v cannot be absorbed internally by T. Hence our technical lemma implies that v belongs to at most $m - c + 2$ edges incident to T. The minimality of c makes v nonadjacent to $x_{a+1}, \dots, x_{c-1}$.

Let b be the largest integer in $[c]$ such that G has a path from x_{a+c} to x_a with vertex set $S - \{x_{a+b}, \dots, x_{a+c-1}\}$. Let R be such a path (the path T with $b = 1$ implies that R exists.) Since $P \cup R$ is a cycle, the maximality of S yields $b < c$. By the maximality of b, x_{a+b} is not absorbed internally by R. Hence, by our technical lemma, x_{a+b} belongs to at most $m - c + b + 1$ edges incident to R.

Now we count $d(v) + d(x_{a+b})$. Each vertex outside $S \cup \{v\}$ is incident to at most two edges also incident to $\{v, x_{a+b}\}$, because the minimality of c prevents a path of length 2 between v and x_{a+b} (in either direction) using a vertex not in S. We have observed that v belongs to at most $m - c + 2$ edges incident to S. We have observed that v_{a+b} belongs to at most $m - c + b + 2$ edges incident to R. Finally, x_{a+b} belongs to at most $2(c - b - 1)$ edges incident to $S - R$. Hence $d(v) + d(x_{a+b}) \le 2(n - m - 1) + (m - c + 2) + (m - c + b + 1) + 2(c - b - 1) = 2n - 1 - b$. Again we have obtained the desired condition. ■

EXERCISES

8.4.1. Let $m = \lfloor n^2/4 \rfloor$. Prove that every n-vertex graph has an intersection representation using subsets of $[m]$ such that each element of $[m]$ appears in at most three sets. Equivalently, every n-vertex graph decomposes into at most m edges and triangles.

8.4.2. Prove that the following conditions on a graph G with no isolated vertices are equivalent. (Choudom–Parthasarathy–Ravindra [1975])

A) $\theta'(G) = \alpha(G)$.
B) $\theta'(G \vee G) = (\theta'(G))^2$.
C) $\theta'(G) = \theta(G)$.
D) Every clique in a minimum clique cover of $E(G)$ uses a simplicial vertex of G.

8.4.3. (+) Let $b(G)$ be the minimum number of bipartite graphs needed to partition $E(G)$ (called **biparticity**). Let $a(G)$ denote the minimum number of classes needed to partition $E(G)$ such that every cycle of G contains a non-zero even number of edges from some class. Prove that these parameters both equal $\lceil \lg \chi(G) \rceil$. (Hint: Prove $\lg \chi(G) \le b(G) \le a(G) \le \lceil \lg \chi(G) \rceil$.) (Harary–Hsu–Miller [1977], Alon–Egawa [1985])

8.4.4. Determine all the n-vertex graphs that have product dimension $n - 1$. (Lovász–Nešetřil–Pultr [1980])

8.4.5. Prove that pdim $G \le 2$ if and only if G is the complement of the line graph of a bipartite graph (Lovász–Nešetřil–Pultr [1980])

8.4.6. Given r, compute pdim $(K_r + mK_1)$ for all $m \ge 1$. (Lovász–Nešetřil–Pultr [1980])

8.4.7. (−) Compute the product dimension of the three-dimensional cube.

8.4.8. Obtain upper and lower bounds on the product dimension of the Petersen graph that differ by 1 (the upper bound will most likely be the correct value, but showing that it cannot be improved is tedious).

8.4.9. Let $f(n)$ be the maximum value of pdim $G \cdot$ pdim $\overline{G}$ over all graphs on n vertices. Prove that $\lfloor n^2/4 \rfloor \le f(n) \le (n-1)^2$.

8.4.10. For $n \ge 4$, prove that pdim $P_n = \lceil \lg(n-1) \rceil$. For $n \ge 3$, prove that pdim $C_{2n} = 1 + \lceil \lg(n-1) \rceil$ and $1 + \lceil \lg n \rceil \le$ pdim $C_{2n+1} \le 2 + \lceil \lg n \rceil$. (Lovász–Nešetřil–Pultr [1980]) (Comment: Evans–Fricke–Maneri–McKee–Perkel [1994] showed that pdim $C_{2n+1} = 1 + \lceil \lg n \rceil$ except possibly when n is a power of 2.)

8.4.11. Prove that C_{2k+1} is not isometrically embeddable in any cartesian product of cliques if $k > 1$.

8.4.12. Determine the squashed-cube dimension of C_5.

8.4.13. (+) Determine the squashed-cube dimension of $K_{3,3}$. (Hint: Use symmetry to reduce case analysis.)

8.4.14. (!) Use Edmonds' Branching Theorem (Theorem 8.4.20) to prove the edge version of Menger's Theorem in digraphs: $\lambda'(x, y) = \kappa'(x, y)$. (Hint: Devise an appropriate graph transformation to obtain a short proof.)

8.4.15. (!) The gossip problem is also called the "telephone problem", and the corresponding problem for directed graphs is called the "telegraph problem". As a function of n, determine the minimum number of one-way transmissions among n people so that each person has a transmission path to every other. (Harary–Schwenk [1974])

8.4.16. Let D be a digraph solving the telegraph problem in which each vertex receives information from each other vertex exactly once. Prove that in D at least $n-1$ vertices hear their own information. For each n, construct such a D in which only $n-1$ vertices hear their own information, but for each $x \neq y$ there is exactly one increasing x, y-path. (Seress [1987]).

8.4.17. *The NOHO property.*

a) Let G be a connected graph with $2n-4$ edges having a linear ordering that solves the gossip problem and satisfies NOHO (no increasing cycle). Suppose also that $n(G) > 8$ and that at most two vertices have degree 2. Prove that the graph obtained by deleting the first calls and last calls of vertices in G has 4 components, of which two are isolated vertices and two are caterpillars having the same size. (West [1982a])

b) For every even $n \geq 4$, construct a connected ordered graph with $2n-4$ edges that satisfies the NOHO property. (Hint: Make use of the structural properties proved in part (a) to guide the search.)

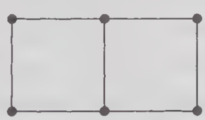
8.4.18. A **NODUP scheme** (NO DUPlicate transmission) is a connected ordered graph that has exactly one increasing path from each vertex to every other.

a) (–) Prove that every NODUP scheme has the NOHO property.

b) Prove that there is no NODUP scheme when $n \in \{6, 10, 14, 18\}$. (Comment: Seress [1986] proved that these are the only even values of n for which NODUP schemes do not exist, constructing them for all other values. For $n = 4k$, West [1982b] constructed NODUP schemes with $9n/4 - 6$ calls, and Seress [1986] proved that these are optimal.)

8.4.19. A vertex in a simple graph G wishes to broadcast information to all other vertices. In each time unit, each vertex that already knows the information can make one call to a neighbor that does not know the information. The time required to broadcast from v is the minimum number of time units in which all vertices can learn the information. Construct an n-vertex graph G with fewer than $2n$ edges such that every vertex of G can broadcast in time at most $1 + \lg n$. (Grigni–Peleg [1991])

8.4.20. (!) Prove that the graph below is not 2-choosable.



8.4.21. Prove that $K_{k,m}$ is k-choosable if and only if $m < k^k$ (Erdős–Rubin–Taylor [1979])

8.4.22. Prove that $\chi_l(G) \leq 1 + \max_{H \subseteq G} \delta(G)$ and that $\chi_l(G) + \chi_l(\overline{G}) \leq n+1$. Prove also that $\chi_l'(G) \leq 2\Delta(G) - 1$.

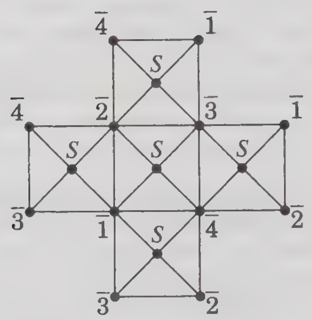
8.4.23. Prove that every chordal graph G is $\chi(G)$-choosable.

8.4.24. Prove that a connected graph G has a proper list coloring from lists such that $|L(v)| \geq d(v)$ for all v if there is strict inequality for at least one vertex.

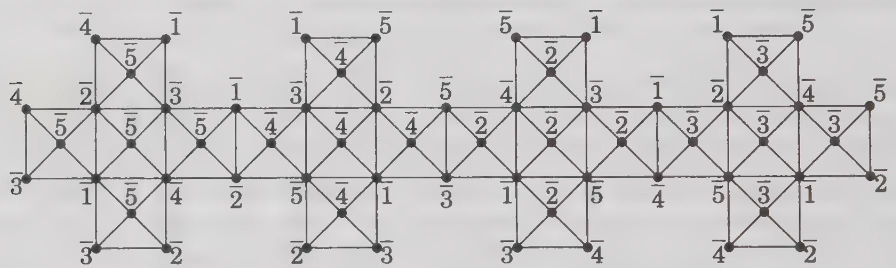
8.4.25. (!) Prove that G has a total coloring (Remark 8.4.31) with at most $\chi_l'(G) + 2$ colors.

8.4.26. (!) *Non-4-choosable planar graph of order 63.*

a) In the list assignments for the graph below, S denotes [4] and $\bar{i}$ denotes $S - \{i\}$. Prove that this graph has no proper coloring chosen from these lists.



b) In the list assignments for the graph G below, $\bar{i}$ denotes $[5] - \{i\}$; each list has size 4. Let G' be the graph obtained from G by adding one vertex with list $\bar{1}$ adjacent to all vertices on the outside face of this drawing of G. Prove that G' has no proper coloring chosen from these lists. (Mirzakhani [1996])



8.4.27. (!) *Equivalence of Dilworth's Theorem and König–Egerváry Theorem.*

a) Given a bipartite graph G, apply Dilworth's Theorem to a transitive orientation of it to obtain the König–Egerváry Theorem.

b) Given a transitive digraph D, let G be the split of D as defined in Definition 1.4.20. Apply the König–Egerváry Theorem to G to obtain Dilworth's Theorem for D.

8.4.28. (!) Prove that K_n decomposes into $\lceil n/2 \rceil$ paths. Prove that K_n decomposes into $\lfloor n/2 \rfloor$ cycles when n is odd.

8.4.29. (!) *Decomposition of K_n into spanning connected subgraphs.*

a) Prove that if K_n decomposes into k spanning connected subgraphs, then $n \geq 2k$.

b) Prove that K_{2k} decomposes into k spanning trees of diameter 3. (Hint: Let the central edges of these trees form a perfect matching.) (Palumbíny [1973])

8.4.30. Prove that every 2-edge-connected 3-regular simple planar graph decomposes into paths of length 3. Prove the same statement for planar triangulations. (Jünger–Reinelt–Pulleyblank [1985])

8.4.31. Prove that Theorem 8.4.35 is best possible when $m - 1$ divides $n - 1$.

8.4.32. Let G be a graph such that $\overline{G}$ is triangle-free and not a forest. Prove that G has a cycle of length at least $n(G)/2$. (Hint: Use Theorem 8.4.37.) (N. Graham)

8.4.33. Use Woodall's Theorem to prove Ore's Theorem, and use Meyniel's Theorem to prove Woodall's Theorem.

8.4.34. Use Meyniel's Theorem to prove that a strict n-vertex digraph has a spanning path if $d(u) + d(v) \geq 2n - 3$ for every pair u, v of distinct nonadjacent vertices.

8.5. Random Graphs

In its simplest form, the probabilistic method is used to prove the existence of desired combinatorial objects without constructing them. An appropriate probability model is defined on a large class of objects. The occurrence of the desired structure is an event. If this event has positive probability, then some object with the desired structure exists. Designing the model and applying probabilistic and asymptotic techniques may involve considerable art.

We discuss these methods in the context of random graphs. The study of random graphs is itself motivated by the modeling of physical properties and by the analysis of algorithms in computer science.

8.5.1. Example. *Melting points.* The behavior of random graphs suggests a mathematical explanation for melting points. Think of a solid as a three-dimensional grid of molecules, with neighboring molecules joined by bonds. For example, consider the graph $P_l \square P_m \square P_n$, with bonds corresponding to edges.

Adding energy excites molecules and breaks bonds. We assume that bonds break at random as we raise the temperature (energy level). Each temperature corresponds to some fraction of bonds broken. While the graph remains largely connected, the material seems solid. Breaking off small pieces doesn't change this, but when all the components are small the global nature of the material changes. Small components of molecules float freely, like a liquid or gas.

Mathematically, there is a threshold for the number of bonds to be broken (in terms of the size of the grid) such that almost every way of breaking somewhat fewer bonds leaves a giant component, and almost every way of breaking somewhat more bonds leaves all components being tiny. Just below the threshold temperature the material will almost certainly be a solid, and just above it the material will almost certainly not be a solid. ■

8.5.2. Example. *Analysis of algorithms.* Worst-case complexity is the maximum running time for an algorithm over all inputs of size n (see Appendix B). For difficult problems, we may seek an algorithm that takes many steps on a few bizarre graphs while running quickly on most graphs. We need a way to describe the usefulness of such algorithms.

The answer is **probabilistic analysis**. We assume a probability distribution on the inputs and study the expected running time with respect to this distribution. Choosing a realistic distribution can be difficult. In practice, we choose a probability distribution that makes the analysis feasible. We cannot define a probability distribution over infinitely many graphs, so we define a distribution on the graphs of each order. This is consistent with viewing the expected running time as a function of the input size. ■

Erdős and Rényi [1959] introduced random graphs. The subject developed rapidly in the 1980s, with books by Bollobás [1985], by Palmer [1985], and by Alon and Spencer [1992] (the last treats broader combinatorial applications of probabilistic methods). The book Janson–Łuczak–Ruciński [2000] emphasizes later developments.

More sophisticated probabilistic techniques than we can present here are now being applied to random graphs. We describe the basic techniques and suggest the flavor of the subject, with no attempt at exhaustive treatment.

EXISTENCE AND EXPECTATION

We begin by showing how probabilistic methods can prove existence statements. Suppose we want to prove that an object with some desired property exists. We define a probability space where occurrence of the desired property is an event A. If A has positive probability, then the desired object exists.

8.5.3. Definition. A discrete **probability space** or **probability model** is a finite or countable set S together with nonnegative weights on the elements that sum to 1. An **event** is a subset of S. The **probability** $P(A)$ of an event A is the sum of the weights of the elements of A. Events A and B are **independent** if $P(A \cap B) = P(A)P(B)$.

Erdős popularized the probabilistic method in 1947 by using it to prove lower bounds on Ramsey numbers (Definition 8.3.6). We phrased this combinatorially in Theorem 8.3.12; here we present the same proof in probabilistic language. It uses the observation that $P(\bigcup_i A_i) \le \sum_i P(A_i)$. Note that **in this section, all graphs are simple**.

8.5.4. Theorem. (Erdős [1947]) If $\binom{n}{p}2^{1-\binom{p}{2}} < 1$, then $R(p, p) > n$.

Proof: It suffices to show that when $\binom{n}{p}2^{1-\binom{p}{2}} < 1$ there is an n-vertex graph G with $\omega(G) < p$ and $\alpha(G) < p$. We define a probability model on graphs with vertex set $[n]$ by letting each edge appear independently with probability .5. If the probability of the event Q = "no p-clique or independent p-set" is positive, then the desired graph exists.

Each possible p-clique occurs with probability $2^{-\binom{p}{2}}$, since obtaining the complete graph requires obtaining all its edges, and they occur independently. Hence the probability of having at least one p-clique is bounded by $\binom{n}{p}2^{-\binom{p}{2}}$. The same bound holds for independent p-sets. Hence the probability of "not Q" is bounded by $\binom{n}{p}2^{1-\binom{p}{2}}$, and the given inequality guarantees that $P(Q) > 0$. ■

8.5.5. Remark. Existence arguments can be used as probabilistic construction algorithms. The probability that a random 1024-vertex graph has a 10-clique or independent 10-set is less than $2^{11}/20!$. If the first random one doesn't work, generate another; the probability of continued failure is the *product* of these small numbers and soon becomes incomprehensibly small. ■

The lower bound in Theorem 8.5.4 is roughly $\sqrt{2}^k$; the inductive upper bound in Theorem 8.3.11 is roughly 4^k. The gap is large. More sophisticated probabilistic methods have achieved only small improvements in the lower bound. Nevertheless, the constructive bounds are much weaker, so this is a triumph for the probabilistic method. The proof is essentially just a counting argument. Many probabilistic arguments with finite sample spaces can be rephrased as weighted counting arguments, but the proofs are simpler in the language of probability.

The introduction of random variables adds considerable power. We assign values to the elements of our probability space.† We have already used the comparison between the average and maximum values of a random variable to prove inequalities.

8.5.6. Definition. A **random variable** is a function assigning a real number to each element of a probability space. We use $X = k$ to denote the event consisting of all elements where the variable X has the value k.

The **expectation** $E(X)$ of a random variable X is the weighted average $\sum_k kP(X = k)$. The **pigeonhole property** of the expectation is the statement that there exists an element of the probability space for which the value of X is as large as (or as small as) $E(X)$.

Applying the pigeonhole property requires a value or bound for $E(X)$. Often the computation applies the **linearity of expectation** to an expression for X in terms of simpler random variables. For our purposes, we generally restrict our attention to probability models on finite sets and sum only finitely many random variables. Analogous results hold in continuous probability spaces.

8.5.7. Lemma. (Linearity property) If X and the finite set $\{X_i\}$ are random variables on the same space and $X = \sum X_i$, then $E(X) = \sum E(X_i)$. Also $E(cX) = cE(X)$ for $c \in \mathbb{R}$.

Proof: In a discrete probability space, each element contributes the same amount to each side of the desired equations. ■

We often apply Lemma 8.5.7 to random variables that count substructures. Such a random variable is a sum of variables indicating whether one of the possible things being counted actually occurs. These **indicator variables** take values in $\{0, 1\}$ (they are also called 0, 1-variables). The expectation of an indicator variable is the probability that it equals 1. These properties facilitate what was perhaps the first use of the probabilistic method.

---

†We consider only discrete probability spaces, but analogous concepts hold for continuous probability spaces.

8.5.8. Theorem. (Szele [1943]) Some n-vertex tournament has at least $n!/2^{n-1}$ Hamiltonian paths.

Proof: Generate tournaments on $[n]$ randomly by choosing $i \to j$ or $j \to i$ with equal probability for each pair $\{i, j\}$. Let X be the number of Hamiltonian paths; X is the sum of $n!$ indicator variables for the possible Hamiltonian paths. Each Hamiltonian path occurs with probability $1/2^{n-1}$, so $E(X) = n!/2^{n-1}$. In some tournament, X is at least as large as the expectation. ∎

This simple bound using expectation gives almost the right answer for the maximum number of Hamiltonian paths in an n-vertex tournament; Alon [1990] proved that it is at most $n!/(2-o(1))^n$. When almost all instances have a value near the extreme, probabilistic arguments are especially effective.

Many inequalities can be interpreted as statements about the expected value of a random variable. This often yields a shorter proof than combinatorial methods. Exercise 3.1.42 requests a combinatorial proof of the next result.

8.5.9. Theorem. (Caro [1979], Wei [1981]) $\alpha(G) \ge \sum_{v \in V(G)} \frac{1}{d(v)+1}$ for every graph G.

Proof: (Alon–Spencer [1992, p81]) Given an ordering of the vertices of G, the set of vertices that appear before all their neighbors form an independent set. When the ordering is chosen uniformaly at random, the probability that v appears before all its neighbors is $1/(d(v)+1)$. Thus the right side of the inequality is the expected size of the independent set formed by choosing the vertices appearing before their neighbors in a random vertex ordering. ∎

When a randomly generated object is close to having a desired property, a slight alteration may produce it. This technique is called the **deletion method**, the **alteration method**, or the **two-step method**. Ramsey numbers furnish a classical application (Exercise 16). We provide two others.

Recall that $S \subseteq V(G)$ is a *dominating set* in G if every vertex outside S has a neighbor in S (Definition 3.1.26). When G is k-regular, every vertex dominates $k+1$ vertices (including itself), so every dominating set has at least $n(G)/(k+1)$ vertices. The alteration method yields a dominating set close to that bound in every graph with minimum degree k. The argument, like many involving these techniques, uses the fundamental inequality $1-p < e^{-p}$ (Exercise 2).

8.5.10. Theorem. (Alon [1990]) Every n-vertex graph with minimum degree $k > 1$ has a dominating set of size at most $n\frac{1+\ln(k+1)}{k+1}$.

Proof: In such a graph G, select a random set $S \subseteq V(G)$ by including each vertex independently with probability $p = \ln(k+1)/(k+1)$. Given S, let T be the set of vertices outside S having no neighbor in S; adding T to S yields a dominating set. We seek the expected size of $S \cup T$.

Since each vertex appears in S with probability p, linearity yields $E(|S|) = np$. The random variable $|T|$ is the sum of n indicator variables for whether individual vertices belong to T. We have $v \in T$ if and only if v and its neighbors all fail to be in S. This has probability bounded by $(1-p)^{k+1}$, since v has degree at least k. Since $(1-p)^{k+1} < e^{-p(k+1)}$, we have $E(|S|+|T|) \le np + ne^{-p(k+1)} = n\frac{1+\ln(k+1)}{k+1}$. The pigeonhole property of the expectation completes the proof. ■

This easy bound yields almost the smallest s_k such that every graph G with minimum degree k has a dominating set of size at most $s_k n(G)$ (Alon [1990]). A greedy algorithm proves the same result constructively (Theorem 3.1.30).

A striking and famous application of the deletion method is the existence of graphs with large girth and chromatic number. Explicit constructions came much later (Lovász [1968a], Nešetřil–Rödl [1979], Kriz [1989]). We present a simplification of the original proof (Alon–Spencer [1992, p35]). It uses a property of the expectation that we will prove in Lemma 8.5.17.

8.5.11. Theorem. (Erdős [1959]) Given $m \ge 3$ and $g \ge 3$, there exists a graph with girth at least g and chromatic number at least m.

Proof: We generate graphs with vertex set $[n]$ by letting each pair be an edge with probability p, independently. A graph with no large independent set has large chromatic number, since $\chi(G) \ge n(G)/\alpha(G)$. We therefore choose p large enough to make large independent sets unlikely. We also choose p small enough to make the expected number of short cycles (length less than g) small. Given a graph satisfying both conditions, we can delete a vertex from each short cycle to obtain the desired graph.

To make it unlikely that we generate more than $n/2$ short cycles, we let $p = n^{t-1}$, where $t < 1/g$. Each of the possible cycles of length j occurs with probability p^j. Since there are $n_{(j)}/(2j)$ of these for each j, the total number X of cycles of length less than g has expectation

$$E(X) = \textstyle\sum_{i=3}^{g-1} n_{(i)} p^i/(2i) \le \sum_{i=3}^{g-1} n^{ti}/(2i).$$

Since $tg < 1$, this implies that $E(X)/n \to 0$ as $n \to \infty$. In Markov's Inequality we will complete the details of concluding from this that $P(X \ge n/2) \to 0$ as $n \to \infty$. For n large enough, $P(X \ge n/2) < 1/2$.

Since $\alpha(G)$ cannot grow when we delete vertices, at least $(n - -X)/\alpha(G)$ independent sets are needed to color the vertices remaining when we delete a vertex of each cycle. If $X < n/2$ and $\alpha(G) \le n/(2k)$, then at least k colors are needed for the graph remaining. With $r = \lceil 3\ln n/p \rceil$, we have

$$P(\alpha(G) \ge r) \le \binom{n}{r}(1-p)^{\binom{r}{2}} < [ne^{-p(r-1)/2}]^r.$$

This approaches 0 as n grows.

Since $r = \lceil 3n^{1-t}\ln n \rceil$ and k is fixed, we can choose n large enough to obtain $r < n/(2k)$. If we also choose n large enough so that $P(X \ge n/2) < 1/2$ and $P(\alpha(G) \ge r) < 1/2$, then there will exist an n-vertex graph G such that $\alpha(G) \le n/(2k)$ and such that G has fewer than $n/2$ cycles of length less than g. We delete a vertex from each short cycle and retain a graph with girth at least g and chromatic number at least k. ■

PROPERTIES OF ALMOST ALL GRAPHS

We have proposed studying properties that "almost always" hold. This phrase has meaning in the context of a probability model.

8.5.12. Definition. Given a sequence of probability spaces, let q_n be the probability that property Q holds in the nth space. Property Q **almost always** holds if $\lim_{n\to\infty} q_n = 1$.

For us, the nth space is a probability distribution over n-vertex graphs. When property Q almost always holds, we say "almost every graph has property Q". Making all graphs with vertex set $[n]$ equally likely is equivalent to letting each vertex pair appear as an edge with probability $1/2$. Models where edges arise independently with the same probability are the most common for random graphs because they lead to the simplest computations. We allow this probability to depend on n.

8.5.13. Definition. Model A: Given n and $p = p(n)$, generate graphs with vertex set $[n]$ by letting each pair be an edge with probability p, independently. Each graph with m edges has probability $p^m(1-p)^{\binom{n}{2}-m}$. The random variable G^p denotes a graph drawn from this probability space. "*The* random graph" means Model A with $p = 1/2$, which makes all graphs with vertex set $[n]$ equally likely.

Computations are much simpler for graphs with a fixed vertex set ("labeled" graphs) than for random isomorphism classes. Since inputs to algorithms are graphs with specified vertex sets, this model is consistent with applications.

We often measure running times of algorithms in terms of the number of vertices and number of edges; hence we may want to control the number of edges. This suggests a model in which the n-vertex labeled graphs with m edges are equally likely. (We use m to count edges in this section because the number $e = 2.71828...$ plays an important role in asymptotic arguments.)

8.5.14. Definition. Model B: Given n and $m = m(n)$, let each graph with vertex set $[n]$ and m edges occur with probability $\binom{N}{m}^{-1}$, where $N = \binom{n}{2}$. The random variable G^m denotes a graph generated in this way.

These two are the most common of many models studied. Model B seems more pertinent for applications. We ask questions like "as a function of n, how many edges are needed to make a graph almost surely connected?" In Model A we would say, "as a function of n, what edge probability is needed to make a graph almost surely connected?" Unfortunately, calculations needed to answer such questions are messier in Model B than in Model A.

Fortunately, Model B is accurately described by Model A when n is large and $p = m/\binom{n}{2}$, because the actual number of edges generated in Model A is almost always very close to the resulting expectation m. The correspondence is valid for most properties of interest. The proof of this requires detailed use of the binomial distribution for the number of edges. A graph property Q is **convex** if G satisfies Q whenever $F \subseteq G \subseteq H$ and F, H satisfy Q.

8.5.15. Theorem. (Bollobás [1985, p34-35]) If Q is convex and $p(1-p)\binom{n}{2} \to \infty$, then almost every G^p satisfies Q if and only if, for every fixed x, almost every G^m satisfies Q, where $m = \lfloor p\binom{n}{2} + x[p(1-p)\binom{n}{2}]^{1/2} \rfloor$. ■

Theorem 8.5.15 justifies restricting our attention to Model A. It also motivates letting p be a function of n; to study graphs with a linear number of edges, we must let p vanish at a rate like c/n, where c is constant. Constant p yields dense graphs.

Proving $P(Q) \to 1$ is usually much easier than computing $P(Q)$; this distinction is important. Exact computation of probabilities is difficult, unnecessary, and avoided wherever possible. Instead we use asymptotic analysis, which rests on limits. We write $a_n \to L$ for $\lim_{n\to\infty} a_n = L$. To compare growth rates of sequences, we use "big O" and "little o" notation (see Appendix B for definitions). We write $a_n = b_n(1+o(1))$ when $\langle a \rangle$ and $\langle b \rangle$ differ by a sequence that grows more slowly than $\langle b \rangle$; equivalently, $a_n/b_n \to 1$. When $a_n/b_n \to 1$, we say that a_n is **asymptotic** to b_n, written $a_n \sim b_n$.

We use asymptotic statements to discard lower-order terms that don't affect whether $\lim_{n\to\infty} P(Q) = 1$. Computing $P(Q)$ first and then proving that the formula tends to 1 is harder and is unnecessary. We need only show that $P(\neg Q)$ is *bounded* by something tending to 0. Many asymptotic arguments are "sloppy" in this sense; we don't care how loose the bound is as long as it tends to 0. Experience refines our intuition about what can be discarded safely.

8.5.16. Theorem. (Gilbert [1959]) When p is constant, almost every G^p is connected.

Proof: We can make G disconnected by picking a vertex partition into two sets and forbidding edges between the two sets. Occurrence of edges within the sets is irrelevant. We bound the probability q_n that G^p is disconnected by summing $P([S,\overline{S}] = \varnothing)$ over all bipartitions $S, \overline{S}$. Graphs with many components are counted many times. When $|S| = k$, there are $k(n-k)$ possible edges in $[S,\overline{S}]$. Each has probability $1-p$ of not appearing, independently, so $P([S,\overline{S}] = \varnothing) = (1-p)^{k(n-k)}$. Considering all S generates each partition from each side, so $q_n \le \frac{1}{2}\sum_{k=1}^{n-1}\binom{n}{k}(1-p)^{k(n-k)}$.

This formula is symmetric in k and $n-k$; hence q_n is bounded by $\sum_{k=1}^{\lfloor n/2 \rfloor}\binom{n}{k}(1-p)^{k(n-k)}$. We loosen the bound to simplify it. Using $\binom{n}{k} < n^k$ and $(1-p)^{n-k} \le (1-p)^{n/2}$ (for $k \le n/2$) yields $q_n < \sum_{k=1}^{\lfloor n/2 \rfloor}(n(1-p)^{n/2})^k$. For large enough n, we have $n(1-p)^{n/2} < 1$. This makes our bound the initial portion of a convergent geometric series. We obtain $q_n < x/(1-x)$, where $x = n(1-p)^{n/2}$. Since $n(1-p)^{n/2} \to 0$ when p is constant, our bound on q_n approaches 0 as $n \to \infty$. ■

We avoid struggling with probability formulas by introducing integer-valued random variables and techniques involving expectation. If X is a nonnegative random variable such that $X = 0$ when G^p has property Q, then $E(X) \to 0$ implies that almost every G^p satisfies Q. This is a special case of the following lemma. We prove it only for integer variables, but it also holds for continuous variables.

8.5.17. Lemma. (Markov's Inequality) If X takes only nonnegative values, then $P(X \ge t) \le E(X)/t$. In particular, if X is integer-valued, then $E(X) \to 0$ implies $P(X = 0) \to 1$.

Proof: $E(X) = \sum_{k\ge 0} kp_k \ge \sum_{k\ge t} kp_k \ge t\sum_{k\ge t} p_k = tP(X \ge t)$. ■

For connectedness, we can define $X(G^p)$ by $X = 1$ if G is disconnected and $X = 0$ otherwise. The expectation of an indicator variable is the probability that it equals 1. We proved $P(X = 1) \to 0$ (when p is constant) to prove that almost every G^p is connected. With a different random variable we can simplify the proof and strengthen the result. We still want G to satisfy Q if $X = 0$ (in order to apply Markov's Inequality), but we don't need $(X = 0) \Leftrightarrow (G \text{ satisfies } Q)$. We use a sum X of many indicator variables, such that G satisfies Q if $X = 0$. The linearity of expectation and convenience of $E(X_i) = P(X_i = 1)$ for the indicator variables simplify the task of proving $E(X) \to 0$.

8.5.18. Theorem. If p is constant, then almost every G_p has diameter 2 (and hence is connected).

Proof: Let $X(G^p)$ be the number of unordered vertex pairs with no common neighbor. If there are none, then G_p is connected and has diameter 2. By Markov's Inequality, we need only show $E(X) \to 0$. We express X as the sum of $\binom{n}{2}$ indicator variables $X_{i,j}$, one for each vertex pair $\{v_i, v_j\}$, where $X_{i,j} = 1$ if and only if v_i, v_j have no common neighbor.

When $X_{i,j} = 1$, the $n-2$ other vertices fail to have edges to both of these, so $P(X_{i,j} = 1) = (1-p^2)^{n-2}$ and $E(X) = \binom{n}{2}(1-p^2)^{n-2}$. When p is fixed, $E(X) \to 0$, and hence almost every G_p has diameter 2. ■

The intuition behind this argument, made precise by Markov's Inequality, is that if we expect almost no bad pairs, then almost every graph has none. The summation disappears, and for the limit we need only know that $(1 - p^2)^{n-2}$ tends to 0 faster than any polynomial function of n.

THRESHOLD FUNCTIONS

Roughly speaking, random graphs with constant edge probability are connected because they have many more edges than needed to be connected. To improve Theorem 8.5.18, we want to make $p(n)$ as small as possible to have almost every G^p connected. We need the notion of a threshold probability function. By the relationship between Model A and Model B, a threshold edge probability also yields a threshold number of edges.

8.5.19. Definition. A **monotone property** is a graph property preserved by addition of edges. A **threshold probability function** for a monotone property Q is a function $t(n)$ such that $p(n)/t(n) \to 0$ implies that almost no G^p satisfies Q, and $p(n)/t(n) \to \infty$ implies that almost every G^p satisfies Q. **Threshold edge function** is defined similarly for Model B.

This is a broad notion of threshold function; it allows a property to have many threshold functions. A threshold function $t(n)$ is "sharper" when the "almost surely" behavior occurs when $p(n)/t(n)$ approaches nonzero constants. Still sharper is a threshold $t(n)$ such that this behavior occurs when $p(n)$ differs from $t(n)$ by the subtraction or addition of a lower-order term.

Markov's Inequality does half the job of deriving a threshold function. If $X = 0$ implies property Q and we prove that $E(X) \to 0$, then $P(Q) \to 1$. We obtain candidates for threshold functions by determining which functions $p(n)$ yield $E(X) \to 0$. Often we obtain $p(n)$ such that $E(X) \to 0$ or $E(X) \to \infty$, depending on the value of a parameter c. The property $E(X) \to \infty$ suggests that $P(X = 0) \to 0$, but this does not always follow. For example, $E(X) \to \infty$ when $P(X = 0) = .5$ and $P(X = n) = .5$. To obtain $P(X = 0) \to 0$, we must prevent the probability from spreading out like this.

8.5.20. Definition. The rth **moment** of X is the expectation of X^r. The **variance** of X, written $Var(X)$, is the quantity $E[(X - E(X))^2]$. The **standard deviation** of X is the square root of $Var(X)$.

8.5.21. Lemma. (Second Moment Method) If X is a random variable, then $P(X = 0) \le \frac{E(X^2)-E(X)^2}{E(X)^2}$. In particular, $P(X = 0) \to 0$ when $\frac{E(X^2)}{E(X)^2} \to 1$.

Proof: Applied to the variable $(X - E(X))^2$ and the value t^2, Markov's Inequality yields $P[(X - E(X))^2 \ge t^2] \le E[(X - E(X))^2]/t^2$. We rewrite this as $P[|X - E(X)| \ge t] \le Var(X)/t^2$ (Chebyshev's Inequality). Since

$$E[(X - E(X))^2] = E[X^2 - 2XE(X) + (E(X))^2] = E(X^2) - (E(X))^2,$$

Chebyshev's Inequality becomes $P[|X - E(X)| \ge t] \le (E(X^2) - E(X)^2)/t^2$. Since $X = 0$ only when $|X - E(X)| \ge E(X)$, setting $t = E(X)$ completes the proof. ■

Intuitively, if the mean grows and the standard deviation grows more slowly, then all the probability is pulled away from 0, and $P(X = 0) \to 0$ results. We illustrate the method by considering the disappearance of isolated vertices. Since a connected graph has no isolated vertices, a threshold for connectedness must be at least as large as a threshold for disappearance of isolated vertices. The computations for the latter are simpler, because we can express this condition using a sum of identically distributed indicator variables with easily computed expectations. In fact, both properties have the same threshold, since it happens that at the threshold almost every graph consists of one huge component plus isolated vertices.

8.5.22. Theorem. In Model A, $\ln n/n$ (natural logarithm) is a threshold probability function for the disappearance of isolated vertices (that is, $\delta(G) \ge 1$). (The corresponding threshold in Model B is $\frac{1}{2}n \ln n$.)

Proof: Let X be the number of isolated vertices, with X_i indicating whether vertex i is isolated. Then $E(X) = \sum E(X_i) = n(1-p)^{n-1}$. We study the asymptotic behavior of $E(X)$ in terms of $p(n)$. Since

$$(1-p)^n = e^{n\ln(1-p)} = e^{-np}e^{-np^2[1/2+p/3+\cdots]},$$

our expression for $E(X)$ simplifies asymptotically if $np^2 \to 0$. This is equivalent to $p \in o(1/\sqrt{n})$ and implies $(1-p)^n \sim e^{-np}$ and $(1-p)^{-1} \sim 1$, yielding $E(X) \sim ne^{-np}$. To simplify further, set $p = c\ln n/n$ to obtain $ne^{-np} = n^{1-c}$, where c may depend on n. Constant c yields $p \in o(1/\sqrt{n})$, as we needed earlier. When $c > 1$, we have $E(X) \sim n^{1-c} \to 0$, which proves one side of the threshold.

When $c < 1$, we have $E(X) \to \infty$ and use the second moment method. We need only show that $E(X^2) \sim E(X)^2$. This uses another helpful property of indicator variables: $X_i^2 = X_i$. Thus,

$$E(X^2) = \textstyle\sum_{i=1}^n E(X_i^2) + \sum_{i\ne j} E(X_iX_j) = E(X) + n(n-1)E(X_iX_j).$$

The indicator variable X_iX_j has value 1 only when v_i and v_j are both isolated, which forbids $2(n-2)+1$ edges. Thus $E(X_iX_j) = (1-p)^{2n-3}$. Again $(1-p)^n \sim e^{-np}$, so $E(X_iX_j) \sim e^{-2np}$, and

$$E(X^2) \sim E(X) + n(n-1)e^{-2np} \sim E(X) + E(X)^2.$$

Since $E(X) \to \infty$, this yields $E(X^2) \sim E(X)^2$. ∎

Theorem 8.5.22 is stronger than required by the definition of threshold function. The threshold is sharper: we guarantee or forbid isolated vertices when the ratio of $p(n)$ to $\ln n/n$ approaches a nonzero constant, not 0 or ∞.

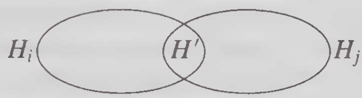
In fact, yet sharper information is known about the threshold for isolated vertices. When $p = \lg n/n + x/n$ and X counts the isolated vertices, $P(X=k) \sim e^{-\mu}\mu^k/k!$, where $\mu = e^{-x}$. (Readers may recognize this limiting distribution as the **Poisson distribution**.) For $k = 0$, we have $P(X=0) \sim e^{-\mu}$. Thus this additive term in p describes the movement through the threshold from almost always isolated vertices to almost never isolated vertices. Many such sharp thresholds are known, but the techniques for deriving the asymptotic Poisson distribution are beyond our scope here.

Next we derive a threshold function for the appearance of fixed subgraphs. A graph is **balanced** if the average vertex degree in every induced subgraph is no larger than the average degree of the entire graph. All regular graphs and all forests are balanced.

8.5.23. Theorem. If H is a balanced graph with k vertices and l edges, then $p = n^{-k/l}$ is a threshold function in Model A for the appearance of H as a subgraph of almost every G^p.

Proof: Let X be the number of copies of H in G^p; X is the sum of indicator variables for the possible copies of H in K_n. There are $n(n-1)\cdots(n-k+1)$ ways to map $V(H)$ into $[n]$. Each copy of H arises A times, where A is the number of automorphisms of H. We thus have $\frac{1}{A}\prod_{j=0}^{k-1}(n-j)$ variables X_i. Since a copy of H occurs when its edges occur, $P(X_i = 1) = p^l$. Because k is fixed, $E(X) \sim n^k p^l / A$.

Setting $p(n) = c_n n^{-k/l}$ yields $E(X) \sim c_n^l / A$. Hence $c_n \to 0$ yields $E(X) \to 0$, and $c_n \to \infty$ yields $E(X) \to \infty$. It remains only to obtain $E(X^2) \sim E(X)^2$ when $c_n \to \infty$. Again $E(X^2) = E(X) + \sum_{i \neq j} E(X_i X_j)$. The summands are not equal; $E(X_i X_j)$ depends on $H' = H_i \cap H_j$. We group the terms by the choice of $H' \subseteq H$. When H' has r vertices and s edges, the number of edges needed to create H_i and H_j is $2l - s$, so $E(X_i X_j) = p^{2l-s}$.



To specify pairs i, j such that $H' = H_i \cap H_j$, we choose r vertices for H', $k - r$ vertices for each of $H_i - H'$ and $H_j - H'$, and an extension of H' to each of those sets. The number of ways to choose the vertex sets is $\frac{n!}{r!(k-r)!(k-r)!(n-2k+r)!}$, which is asymptotic to $n^{2k-r}/[r!(k-r)!^2]$. The number M of ways to extend H' to obtain copies of H in both specified k-sets depends only on H and H'; it is independent of n and p. Let $\alpha_{H'}$ be the constant $M/[r!(k-r)!^2]$. The contribution to $\sum E(X_i X_j)$ from pairs i, j such that $H_i \cap H_j = H'$ is asymptotic to $\alpha_{H'} n^{2k-r} p^{2l-s}$; we call this $E_{H'}$.

When $r = s = 0$, we have $M = (k!/A)^2$. Hence $\alpha_{H'} \sim n^{2k} p^{2l} / A^2 \sim E(X)^2$ when H' is the "null graph". This is the total contribution to $\sum E(X_i X_j)$ for all i, j with H_i, H_j disjoint and is asymptotic to $E(X)^2$. The proof is completed by showing that the total contribution from all other choices of H' has lower order. We have $E_{H'} \sim \alpha_{H'} A^2 E(X)^2 n^{-r} p^{-s}$. Since $2s/r$ is the average degree of H', the hypothesis that H is balanced yields $2r/s \geq 2k/l$, or $pn^{r/s} \geq pn^{k/l} \to \infty$ when $c_n \to \infty$. Since $pn^{r/s} \to \infty$ is equivalent to $n^{-r} p^{-s} \to 0$, we obtain $E_{H'} \in o(E(X)^2)$ for $H' \neq \emptyset$. Since the number of possible subgraphs H' is bounded (by an expression involving the constants k and l), this implies that $E(X^2) \sim E(X) + E_{\emptyset} \sim E(X)^2$. ■

This result generalizes for all H. The ratio $d(H) = e(H)/n(H)$ is the **density** of H, and $\rho(H) = \max_{F \subseteq H} d(F)$ is the **maximum density**. These are equal precisely when H is balanced, and then $p = n^{-1/\rho(H)}$ is the threshold for appearance of H. Every graph H has a balanced subgraph F such that $d(F) = \rho(H)$. When $pn^{\rho(H)} \to 0$, almost every G^p has no copy of F; hence it also has no copy of H. In fact, $p = n^{-1/\rho(H)}$ is always a threshold function for the appearance of H (Exercise 25).

EVOLUTION AND GRAPH PARAMETERS

In the subtitle to his book, Palmer [1985] tells us that random graphs involve the study of

> "THRESHOLD FUNCTIONS, which facilitate the careful study of the structure of a graph as it grows, and specifically reveal the mysterious circumstances surrounding the abrupt appearance of the UNIQUE GIANT COMPONENT, which systematically absorbs its neighbors, devouring the larger first and ruthlessly continuing until the last ISOLATED VERTICES have been sucked up, whereupon the Giant is suddenly brought under control by a SPANNING CYCLE."

The evolutionary viewpoint generates random graphs with m edges in a way that yields the same probability space as Model B but makes intuitive reasoning easier. Almost everything suggested about random graphs by intuition or experimentation is true. The evolutionary viewpoint develops this intuition.

Generating m edges simultaneously or one-by-one yields the same probability distribution, making the graphs with m edges equally likely. By studying the likely effect of a new edge on the present structure, we can make intuitive hypotheses about the properties of the graph at any stage. A *stage* of evolution is a range of values for $m(n)$ (or $p(n)$) in which the structural description of the typical graph doesn't change much. We have studied the basic techniques for verifying these descriptions, but the computations can be difficult. Hence we will only describe the stages using the evolutionary intuition.

We remark first that a constant multiple of almost nothing is almost nothing. Therefore, when each of $A_1, \ldots, A_r$ happens almost always (r is fixed), it follows that almost always they all happen.

Beginning with many vertices and no edges, each new edge is likely to be isolated. The random graph is a matching until a substantial fraction of the vertices are involved in edges. The thresholds $p \sim cn^{-k/(k-1)}$ for appearance of fixed subtrees generalize this. Let $t_k(n) = n^{-k/(k-1)}$. If $p/t_k \to \infty$ but $p/t_{k+1} \to 0$, then every fixed subtree on k vertices appears, but none on $k+1$ vertices appears. (The statements about individual trees become statements about all trees of that order.) Furthermore, this p is also below the threshold for appearance of fixed cycles (density 1, length bounded by k), so G^p is a forest of trees of order at most k, and every tree on k vertices appears as a component.

Intuitively, the random graph has no cycles in this stage of evolution because when there is no large component a random added edge is much more likely to join two components than to lie in one component. To make the intuition precise, we let X be the number of cycles in G^p and compute

$$E(X) = \sum_{k=3}^{n} \binom{n}{k} \tfrac{1}{2}(k-1)!p^k < \sum_{k=3}^{n} (np)^k/2k.$$

If $pn \to 0$, then $E(X) \to 0$.

The next major stage of evolution is $p = c/n$ with $0 < c < 1$. With X counting cycles, we can no longer say that $E(X) \sim \sum_{k=3}^{n} (np)^k/2k$, because when

k is a substantial fraction of n the ratio $n^k/(n)_k$ does not approach 1. We must break $E(X)$ into two sums, and the arguments become more difficult. When $pn \to c$, we find that $E(X)$ approaches a constant c', and the number of cycles in G^p is asymptotically Poisson distributed. With cycles in a few components and all components small, we still expect the next edge to join two components or create a cycle in a component that doesn't have one. In this range, the size of the largest component is about $\log n$, there are many components, each having at most one cycle. Most vertices still belong to acyclic components.

When c reaches and passes 1, the structure of G^p changes radically. This is called the **double jump** because the structure of G^p is significantly different for $c < 1$, $c \sim 1$, and $c > 1$. At $pn = 1$, the second moment method guarantees that almost every G^p has a cycle. Also, the order of the largest component jumps from $\log n$ to $n^{2/3}$. For $pn = c > 1$, the number of vertices outside the "giant component" becomes $o(n)$. Also G^p is likely to have some cycle with three crossing chords and be nonplanar.

Next, let p approach $c \ln n/n$. With $c < 1$, we have proved that almost every G^p has isolated vertices. With $c > 1$, these disappear. As we add edges to a disconnected graph, the edges may go within a component or connect two components. When the components are all small, added edges will almost surely join components. Eventually, this results in the creation of a giant component. At this point, added edges are likely to lie within the giant component or to join it to one of the small components. Of the small components, those most likely to receive such edges are the larger ones. In other words, as c passes through 1 the last remaining small components swallowed by the giant component are isolated vertices. This explains intuitively why the threshold for connectedness is the same as the threshold for the disappearance of isolated vertices. With $c > 1$, suddenly almost every G^p also has a spanning cycle. Minimum degree k (and the appearance of the Hamiltonian cycle when $k = 2$) has a threshold that involves a lower-order term: $\ln n/n + (k-1)\ln\ln n/n$.

The last stages of evolution are those where $pn/\ln n \to \infty$ but $p = o(1)$, and then finally $p = c$; this brings us back to where we began our study.

When $p = c\log n/n$ with $c \to \infty$, we leave the domain of sparse graphs. The evolutionary viewpoint becomes less valuable, and we study properties of the random graph. We pay less attention to probability threshold functions and concentrate on the likely value of graph parameters, especially when p is constant. Given a parameter μ, we want to show that $\mu(G^p) \sim f(n)$ for almost every G^p. We can view this as a threshold when $\mu(G^p)$ is almost always between $(1-\epsilon)f(n)$ and $(1+\epsilon)f(n)$, for each $\epsilon > 0$. If $\mu(G^p)$ is almost always between $f(n) - \epsilon g(n)$ and $f(n) + \epsilon g(n)$, where $g(n) = o(f(n))$, then we have a stronger statement, written as $\mu(G^p) \in f(n)(1 + o(1))$.

Some properties that are true for almost all graphs occur in no known examples! For the known lower bound on Ramsey numbers, there is still no construction of an infinite class of graphs such that $\alpha(G) < \log_{\sqrt{2}}(n(G))$ and $\omega(G) < \log_{\sqrt{2}}(n(G))$, even though almost all graphs have this property.

Properties of the random graph can lead to a fast algorithm that solves a difficult problem on almost all inputs. For example, after stating two results about vertex degrees in random graphs, we show how to use properties of the degree sequence to design a fast algorithm to test isomorphism "almost always". In the literature of random graphs, ω_n denotes a function that is unbounded but grows arbitrarily slowly.

8.5.24. Theorem. (Erdős–Rényi [1966]) If $p = \omega_n \log n/n$ and $\epsilon > 0$ is fixed, then almost every G^p satisfies

$$(1-\epsilon)pn < \delta(G^p) \le \Delta(G^p) \le (1+\epsilon)pn. \qquad \blacksquare$$

Most vertices have degree near the average, but there is still considerable variation. Bollobás [1982] showed that for $p \le 1/2$, the vertex of maximum degree is unique in almost every G^p if and only if $pn/\log n \to \infty$. When we complete evolution by returning to the realm of constant edge probability, more detailed results are known about the degree distribution. There will almost always be some vertices with isolated high degrees before the degrees begin to bunch up. Bollobás determined how many distinct degrees can be guaranteed.

8.5.25. Theorem. (Bollobás [1981b]) In Model A with p fixed and $t \in o(n/\log n)^{1/4}$, almost every G^p has different degrees for its t vertices of highest degree. If $t \notin o(n/\log n)^{1/4}$, then almost every G^p has $d_i = d_{i+1}$ for some $i < t$. ■

We apply this result to isomorphism testing. No polynomial-time algorithm is known for this problem, but Babai–Erdős–Selkow [1980] used the degree results for the random graph to develop a fast algorithm that almost always works. We define a set **H** that contains almost all graphs and show that isomorphism with a graph in **H** can be tested quickly.

The testing is done by a *canonical labeling algorithm*, which accepts and labels a graph in a canonical way if it belongs to **H**. The desired property is that when vertices are labeled as $v_1, \ldots, v_n$ in one graph and $w_1, \ldots, w_n$ in another, only the bijection mapping v_i to w_i is a possible isomorphism. Isomorphism can then be tested by comparing the adjacency matrices under this labeling.

8.5.26. Corollary. (Babai–Erdős–Selkow [1980]) There is a quadratic algorithm that tests isomorphism for almost all pairs of graphs.

Proof: Given a graph G on n vertices, presented by its adjacency matrix, compute and sort the vertex degrees, labeling the vertices in decreasing order of degree. Fix $r = \lfloor 3 \lg n \rfloor$. If $d(v_i) = d(v_{i+1})$ for any $i < r$, reject G. Using $p = 1/2$ in Theorem 8.5.25 implies that almost every graph successfully passes this test.

Let $U = \{v_1, \ldots, v_r\}$. With $r = \lfloor 3 \lg n \rfloor$, there are about n^3 distinct subsets of the vertices of U. Since only $n - r$ vertices remain outside U, there is a chance that they can be distinguished by their neighborhoods in U. The set **H** will be all the graphs reaching this stage for which this holds: the vertices of $V - U$ have distinct neighborhoods in U. To test this in $O(n^2)$ time and complete the labeling, for each $x \in V - U$ encode $N(x) \cap U$ as a binary r-tuple. Evaluate these as binary integers, and sort them! These steps take $O(n \log n)$ time. Relabel the vertices v_{r+1} to v_n as $w_{r+1}, \ldots, w_n$ in decreasing order of these values. If two consecutive values are the same, reject G.

If G has passed this far, then G has no nontrivial automorphisms. A graph isomorphic to G has only one isomorphism to G, given by applying the canonical labeling algorithm to it. The last stage, if both graphs pass canonical labeling, is to compare the adjacency matrices with rows and columns indexed by the canonical labeling. The graphs are isomorphic if and only if the matrices are now identical. This comparison takes $O(n^2)$ time.

We must show that for almost every G^p, the adjacency vectors within a specified set of r vertices are distinct for the remaining vertices. If $p \leq 1/2$, then the probability for any pair x, y that x, y have the same adjacencies in U is bounded approximately by $(1-p)^r$. We say approximately because U is not chosen at random; choosing U as the set of vertices of highest degree impairs randomness, increasing the probability of a specified edge incident to these vertices. Nevertheless, it doesn't change by much, and the expected number of pairs of vertices outside U with identical adjacencies in U is bounded by $O(\binom{n-r}{2}(1-p)^r)$. Given our choice of r, we can bound the base 2 logarithm of this by $2 \lg n - 3 \lg b \lg n$, where $b = 1/(1-p) \geq 2$ (if $p \leq 1/2$). This tends to $-\infty$, so almost all graphs have distinct adjacency vectors in this set. ■

The probability of rejection in this labeling algorithm is bounded by $n^{-1/7}$ for sufficiently large n. Later improvements led to an algorithm running in time $O(n^2)$ with rejection probability c^{-n} (Babai–Kučera [1979]).

CONNECTIVITY, CLIQUES, AND COLORING

Studying the "typical behavior" of a random structure often involves studying probability distributions of its parameters. Here we consider connectivity, cliques, and colorings for random graphs.

For random graphs, naive algorithms may become good. For example, finding a maximum clique is NP-hard. If we know that almost every graph has clique number about $2 \lg n$, then we can test all vertex subsets up to size $3 \lg n$ for being cliques. If $\omega(G) < 3 \lg n$, then this computes $\omega(G)$, since every set of size $\omega(G) + 1$ is not a clique. If $\omega(G) \geq 3 \lg n$, then the algorithm fails to compute $\omega(G)$, but this rarely happens. There are too many subsets of size $2 \lg n$ for this to be a polynomial-time algorithm, but it's close, and it illustrates one way in which the properties of random graphs can be used algorithmically.

Some NP-hard problems are trivial for random graphs. Although $\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$ for every simple graph G (Vizing [1964]), deciding between these values is NP-hard (Holyer [1981]). Vizing proved that $\chi'(G) = \Delta(G) + 1$ only when G has at least 3 vertices of maximum degree. Thus Erdős and Wilson [1977], who noted the uniqueness of the vertex of maximum degree when $p = 1/2$, also observed that $\chi'(G) = \Delta(G)$ for the random graph.

For sparse graphs and constant k, the thresholds for connectivity k and minimum degree k are the same. Does this also hold for constant edge probability? Theorem 8.5.18 can be generalized and strengthened to show that if $k \in o(n/\log n)$ and p is fixed, then almost every G^p has k common neighbors for every vertex pair and hence is k-connected (Exercise 33). Improving this requires other methods; Bollobás [1981b] showed for constant p that almost every G^p has connectivity equal to minimum degree.

What about clique number? For fixed k, Theorem 8.5.23 yields a probability threshold for the appearance of a k-clique, but for constant p the clique number grows with n. Determining the clique number is NP-complete, but for a random graph we can guess the correct value with high probability without looking at the graph! Amazingly, for fixed p almost every G^p has one of two possible values for the clique number (as a function of n), and for each $k \in \mathbb{N}$ there is a range of n where the clique number almost always equals k. The approach is to find bounds on $r(n)$ such that almost every G^p has an r-clique and almost none has an $r+1$-clique.

8.5.27. Theorem. (Matula [1972]) For fixed $p = 1/b$ and fixed $\epsilon > 0$, almost every G^p has clique number between $\lfloor d - \epsilon \rfloor$ and $\lfloor d + \epsilon \rfloor$, where $d = 2\log_b n - 2\log_b \log_b n + 1 + 2\log_b(e/2)$.

Proof: (sketch) If X_r is the number of r-cliques, then $E(X_r) = \binom{n}{r} p^{\binom{r}{2}}$. Since $r! \sim (r/e)^r\sqrt{2\pi r}$ (Stirling's approximation), also $E(X_r) \sim (2\pi r)^{-1/2}(enr^{-1}p^{(r-1)/2})^r$. If $r \to \infty$ and $(enr^{-1}p^{(r-1)/2}) \le 1$, then we expect that $E(X_r) \to 0$. To determine $r(n)$ such that this holds, take logarithms (base b) in the inequality and solve for r to find

$$r \ge 2\log_b n - 2\log_b r + 1 + 2\log_b e.$$

This is approximately equivalent to $r \ge d(n)$ as defined above. More precisely, if $r > d + \epsilon$, then almost every G^p has no clique of size r.

The lower bound comes from careful application of the second moment method, as in Theorem 8.5.23, but the dependence of r on n makes the analysis more difficult. The expectation of X_r^2 sums the probability of common occurrence for all ordered pairs of r-cliques. This probability depends only on the number of common vertices, so

$$E(X_r^2) = \binom{n}{r}\sum_{k=0}^{r}\binom{r}{k}\binom{n-r}{r-k}p^{2\binom{r}{2}-\binom{k}{2}}.$$

We want to show that the term for $k = 0$ (disjoint cliques) dominates. Let $E(X_r^2)/E(X_r)^2 = \alpha_n + \beta_n$, where $\alpha_n = \binom{n}{r}^{-1}\binom{n-r}{r}$ and $\beta_n = \binom{n}{r}^{-1}\sum_{k=1}^{r}\binom{r}{k}\binom{n-r}{r-k}b^{\binom{k}{2}}$. We seek $\alpha_n \sim 1$ and $\beta_n \to 0$. When $r \sim 2\log_b n$, an asymptotic formula for $\binom{a}{k}/\binom{b}{k}$ leads to $\alpha_n \sim e^{-r^2/(n-r)} \to 1$. The discussion of β_n is more difficult; see Palmer [1985, p75-80]. ■

Our study of graph parameters can be applied to measure the strength of conditions for Hamiltonian cycles (Palmer [1985, p81-85]). A theorem proves nothing if its hypotheses are never satisfied; this suggests saying that such a theorem has strength 0. A theorem is strong if the conclusion is satisfied only when the hypothesis is satisfied; then the hypotheses cannot be weakened. Define the **strength** of a theorem to be the probability that its hypotheses are satisfied divided by the probability that its conclusion is satisfied.

Consider sufficient conditions for Hamiltonian cycles. Since $p = \log n/n$ is a threshold for a Hamiltonian cycle, almost every G^p is Hamiltonian when p is fixed. Dirac [1952b] showed that G is Hamiltonian when every vertex degree is at least $n/2$ (Theorem 7.2.8). When $p > 1/2$, this condition holds for almost every G^p; when $p \le 1/2$, it almost never holds. Hence the asymptotic strength of Dirac's Theorem is 0 when p is a constant at most 1/2. The same fate befalls the other degree conditions of Section 7.2.

Meanwhile, Chvátal and Erdős [1972] proved that G is Hamiltonian whenever its connectivity exceeds its independence number (Theorem 7.2.19). Our thresholds for these parameters imply that this result is strong for every constant $p > 0$. We know that $\alpha(G^p) < 2(1+\epsilon)\log_b n$ almost always, and we know that $\kappa(G^p) \ge k$ almost always (when $k = o(n/\log n)$). Hence $\kappa > \alpha$ for almost every G^p, and the asymptotic strength of the theorem is 1.

Finally, we consider chromatic number for constant p. Since $1 - p$ is also constant, we can apply the results on clique number: Almost every G^p has no stable set with more than $(1+o(1))2\log_b n$ vertices, where $b = 1/(1-p)$. Hence $\chi(G^p) \ge (1/2 + o(1))n/\log_b n$ almost always. Achieving this bound requires finding many disjoint stable sets with near-maximum sizes. For a decade, the best result was an algorithmic guarantee of a coloring with at most twice the number of colors in the lower bound.

Bollobás [1988] proved that the lower bound is achievable, by using another probabilistic technique that guarantees finding enough large stable sets. He proved that, in almost every G^p, *every* set having at least $n/(\log_b n)^2$ vertices contains a clique of order at least $2\log_b n - 5\log_b\log_b n$. This allows stable sets of near-maximum size to be extracted until too few vertices remain to cause trouble; the remainder can be given distinct colors.

Before developing Bollobás' approach, we present the earlier result for its algorithmic interest; the greedy algorithm uses at most $(1+\epsilon)n/\log_b n$ colors on almost every G^p. Thus it "almost always works" as an approximation algorithm in the same sense that our earlier isomorphism algorithm almost always works. Garey and Johnson [1976] showed there is no fast algorithm that uses at most twice the optimum number of colors on *every* graph unless P = NP. Bollobás' proof does not yield a fast algorithm for coloring almost every graph with an asymptotically optimal number of colors; it is an existence proof only.

8.5.28. Theorem. (Grimmett–McDiarmid [1975]) Given edge probability p, let $b = 1/(1-p)$. For constant p and constant $\epsilon > 0$, almost every G^p satisfies

$$(1/2-\epsilon)n/\log_b n \le \chi(G^p) \le (1+\epsilon)n/\log_b n.$$

Proof: The lower bound follows using stable sets as suggested above. For the upper bound, we show that the greedy coloring of $v_1, \ldots, v_n$ in order uses at most $f(n) = (1+\epsilon)n/\log_b n$ colors on almost every G^p (for simplicity, choose ϵ so that $f(n)$ is an integer). Within the set of n-vertex graphs using more colors, let $\mathbf{B}_m$ be the set such that v_m is the first vertex to use color $f_n + 1$. We prove that $\sum_{m=1}^{n} P(\mathbf{B}_m) \to 0$ as $n \to \infty$.

Given G, let $G_m = G[\{v_1, \ldots, v_{m-1}\}]$. Before color $f_n + 1$ is used, color f_n must be used, so for each $G \in \mathbf{B}_m$ the greedy coloring of G_m uses f_n colors. Let k_i be the number of times color i appears in this coloring. To require use of color $f_n + 1$, v_{m+1} must have at least one neighbor of each color $1, \ldots, f_n$. Given the numbers $\{k_i\}$, the probability of this is $\prod_{i=1}^{f(n)}[1-(1-p)^{k_i}]$.

Bollobás and Erdős [1976] simplified the subsequent computations involving this bound by observing that the bound is maximized when the k_i's are all equal (Exercise 8.3.37). Thus

$$\prod_{i=1}^{f(n)}[1-(1-p)^{k_i}] \le [1-(1-p)^{(m-1)/f}]^f < [1-(1-p)^{n/f}]^f.$$

Given G_m, we have $b_n = [1-(1-p)^{n/f(n)}]^{f(n)}$ as a bound on the probability that the full graph G belongs to $\mathbf{B}_m$. Since this holds for each G_m, we conclude that $P(\mathbf{B}_m) < b_n$. This holds for all m, so $\sum_{m=1}^{n} P(\mathbf{B}_m) < nb_n$.

Using $(1-p)^{-x} < e^{-x}$, we obtain $nb_n < ne^{-f(1-p)^{n/f}}$. Substituting $f_n = cn/\log_b n$ yields $(1-p)^{n/f} = n^{-1/c}$. The logarithm of the bound becomes $\log n - cn^{1-1/c}/\log_b n$. This tends to $-\infty$ for $c > 1$, so the probability that the greedy algorithm uses more than $f(n)$ colors is bounded by a function tending to 0. ■

The order of growth of $\chi(G)$ sheds light on other famous problems in graph theory. Hajós conjectured that every r-chromatic graph contains a subdivision of K_r (see Remark 5.2.21). This was disproved by Catlin [1979] (Exercise 5.2.40). Erdős and Fajtlowicz [1981] observed that the chromatic number of G^p almost always grows like $\Theta(n/\log n)$). On the other hand, the largest r such that G^p contains a subdivision of K_r grows like $\Theta(\sqrt{n})$). Thus the chromatic number is almost always much larger, and Hajós' Conjecture is almost always very false.

In contrast, almost every G^p has a subgraph contractible to K_r when $r \in \Theta(n/\sqrt{\log n})$. Thus almost every graph satisfies the weaker conjecture of Hadwiger (Remark 5.2.21), which states that every r-chromatic graph has a subgraph contractible to K_r.

MARTINGALES

Advanced techniques in probability lead to elegant results on combinatorial structures without the drudgery involved in second moment and higher moment computations. The theory aims to develop paradigms that can be applied without repeating computational details.

Some of these methods employ lists of related random variables. The resulting stochastic process displays more consistent and predictable global behavior than the individual random variables do.

In the classical random walk on a line, at each step there is probability p of moving one unit to the left, probability p of moving one unit to the right, and probability $1 - 2p$ of not moving. No matter what the earlier history of the walk has been, the expected position after t steps equals the actual position after $t - 1$ steps. This is the defining property of a martingale.

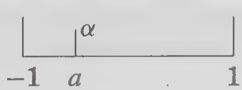
8.5.29. Definition. A **martingale** is a list of random variables $X_0, \dots, X_n$ such that the expectation of X_i, given the values of $X_0, \dots, X_{i-1}$, equals X_{i-1}.

The expected position of the random walk after n steps is at the origin. Less obvious is that the walk is highly unlikely to be very far from the origin, as a function of n. We shall see that this follows from its inability to move more than one unit in each step.

Martingales can make it easy to show that a random variable is highly concentrated around its expected value. When the technique applies, it makes the detailed computation in the Second Moment Method unnecessary. The hard work is accomplished by Azuma's Inequality, also called the Martingale Tail Inequality. This inequality states that if successive random variables in a martingale always differ by at most 1, then the probability that $X_n - X_0$ exceeds $\lambda\sqrt{n}$ is bounded by $e^{-\lambda^2/2}$. We first prove two lemmas. These statements hold for continuous random variables, but again we consider only discrete variables.

8.5.30. Lemma. Let Y be a random variable such that $E(Y) = 0$ and $|Y| \le 1$. If f is a convex function on $[-1, 1]$, then $E(f(Y)) \le \frac{1}{2}[f(-1) + f(1)]$. In particular, $E(e^{tY}) \le \frac{1}{2}(e^t + e^{-t})$ for all $t > 0$.

Proof: When Y takes only the values ± 1, each with probability .5, we have $E(f(Y)) = \frac{1}{2}[f(-1) + f(1)]$. For other distributions, pushing probability "out to the edges" increases $E(f(Y))$. For discrete variables, we can use induction on the number of values with nonzero probability. Convexity implies that $f(a) \le \frac{1-a}{2} f(-1) + \frac{a+1}{2} f(1)$. If $P(Y = a) = \alpha$, then we can decrease the probability at a to 0, increase $P(Y = -1)$ by $\alpha\frac{1-a}{2}$ and increase $P(Y = 1)$ by $\alpha\frac{a+1}{2}$ to obtain a new variable Y' with the same expectation. By the convexity inequality and the induction hypothesis, $E(f(Y)) \le E(f(Y')) \le \frac{1}{2}[f(-1) + f(1)]$. ∎



8.5.31. Definition. For events A and B, the **conditional probability** of A *given* B is obtained by treating the event B as the full probability space, which means normalizing by $P(B)$. Thus we define $P(A|B) = \frac{P(A \text{ and } B)}{P(B)}$.

When Y, X are random variables, we write $Y|X$ for "Y *given* X". This defines a random variables for each value of X; we treat X as a constant i and normalize the resulting distribution for Y by $P(X = i)$.

For Azuma's Inequality, we use expectation of conditional variables. For each i, we compute the expected value of Y when restricted to the sample points where $X = i$. The expectation $E(E(Y|X))$ is the expectation of $E(Y|X = i)$ over the choices for i, which occur with probability $P(X = i)$. The result is an expectation over the entire sample space. It removes the effect of conditioning, and we obtain $E(E(Y|X)) = E(Y)$.

8.5.32. Lemma. $E(E(Y|X)) = E(Y)$.

Proof: Let $p_{i,j} = P(X = i \text{ and } Y = j)$. Since $E(Y|X = i) = \frac{\sum_j jp_{i,j}}{P(X=i)}$,

$$E(E(Y|X)) = \sum_i E(Y|X = i)P(X = i) = \sum_i \sum_j jp_{i,j} = E(Y). \qquad \blacksquare$$

8.5.33. Theorem. (Azuma's Inequality) If $X_0, \ldots, X_n$ is a martingale with $|X_i - X_{i-1}| \le 1$, then $P(X_n - X_0 \ge \lambda\sqrt{n}) \le e^{-\lambda^2/2}$.

Proof: By translation, we may assume that $X_0 = 0$. For $t > 0$, we have $X_n \ge \lambda\sqrt{n}$ if and only if $e^{tX_n} \ge e^{t\lambda\sqrt{n}}$, and hence $P(X_n \ge \lambda\sqrt{n}) = P(e^{tX_n} \ge e^{t\lambda\sqrt{n}})$. Applied to e^{tX_n}, Markov's Inequality yields $P(e^{tX_n} \ge e^{t\lambda\sqrt{n}}) \le E(e^{tX_n})/e^{\lambda t\sqrt{n}}$. This bound holds for each $t > 0$, and later we will choose t to minimize the bound.

First we prove by induction on n that $E(e^{tX_n}) \le \frac{1}{2}(e^t + e^{-t})$. We introduce X_{n-1} to condition on it. Lemma 8.5.32 yields

$$E(e^{tX_n}) = E(e^{tX_{n-1}}e^{t(X_n - X_{n-1})}) = E(E(e^{tX_{n-1}}e^{t(X_n - X_{n-1})}|X_{n-1})).$$

When we condition on X_{n-1}, the value of X_{n-1} is constant for the inner expectation. Hence we can remove $e^{tX_{n-1}}$ from the inner expectation to obtain $E(e^{tX_n}) = E(e^{tX_{n-1}}E(e^{tY}|X_{n-1}))$, where $Y = X_n - X_{n-1}$. Because $\{X_n\}$ is a martingale, $E(Y) = 0$, and by hypothesis $|Y| \le 1$. Hence Lemma 8.5.30 applies, yielding $E(e^{tY}|X_{n-1}) \le \frac{1}{2}(e^t + e^{-t})$. This itself is now a constant, yielding $E(e^{tX_n}) = \frac{1}{2}(e^t + e^{-t})E(e^{tX_{n-1}})$. The induction hypothesis completes the proof.

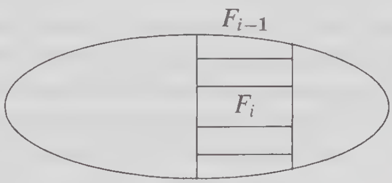
We weaken the bound to a more useful form by observing that $\frac{1}{2}(e^t + e^{-t}) \le e^{t^2/2}$. This holds because the left side is $\sum t^{2k}/(2k)!$ and the right side is $\sum t^{2k}/(2^k k!)$. Hence our original probability is bounded by $e^{nt^2/2 - \lambda t\sqrt{n}}$ for each $t > 0$. We obtain the best bound by minimizing over t. The exponent is quadratic; we minimize it by choosing t to solve $tn - \lambda\sqrt{n} = 0$, or $t = \lambda/\sqrt{n}$. The resulting bound is $e^{-\lambda^2/2}$. $\blacksquare$

Azuma's Inequality is one-sided; it bounds the probability that X_n is much larger than X_0. Since the conditions are symmetric in sign, applying the inequality to $\{-X_i\}$ yields the same inequality for the other tail, in which X_n is much smaller than X_0.

8.5.34. Example. *The pragmatic gambler.* A gambler can bet up to n times, where n is fixed. Each time he bets, he wins or loses 1 with equal probability.

His goal is winning $\lambda\sqrt{n}$, so he stops if he reaches that value. Letting X_i be his winnings after i games, we have $X_i = X_{i-1}$ if $X_{i-1} \geq \lambda\sqrt{n}$, and otherwise $X_i = X_{i-1} \pm 1$, each with probability .5. Hence $\{X_i\}$ is a martingale that changes by at most 1 at each step, and Azuma's Inequality applies. The probability that the gambler will earn $\lambda\sqrt{n}$ is bounded by $e^{-\lambda^2/2}$. If $\lambda = 1$, then there may be a reasonable chance of success, but if $\lambda = 10$, then there is little hope. ■

In combinatorial applications, we consider a special type of martingale. We have an underlying probability space, and X_0 is the expectation of a random variable X. The variable X_n is the value of X at one sample point. We define a martingale $X_0, \ldots, X_n$ that describes a gradual process of learning more about the final value $X_n = X$.



8.5.35. Lemma. Let X be a random variable defined on a probability space. Let $F_0 \supseteq F_1 \supseteq \cdots \supseteq F_n$ be a chain of subsets of the space, where F_0 is the full space, F_n is a single outcome, and F_i is a random variable that is a block in a partition of F_{i-1}. The probability of choosing F_i within F_{i-1} is proportional to its probability in the underlying space. If $X_i = E(X|F_i)$, then the list $X_0, \ldots, X_n$ is a martingale.

Proof: We must prove that $E(X_i|X_0, \ldots, X_{i-1}) = X_{i-1}$. In a particular instance of the process, the list of values is the outcome of a particular sequence of restrictions. Each sequence of restrictions that generates the given values $X_0, \ldots, X_{i-1}$ reaches some F_{i-1} such that $E(X|F_{i-1})$ has the given value of X_{i-1}. For every such F_{i-1}, we can take the expectation of X_i over the possible values of F_i. In each case, we obtain X_{i-1}, so the desired formula holds regardless of which F_{i-1} generated the list $X_0, \ldots, X_{i-1}$.

We thus condition on a fixed choice of F_{i-1} to compute $E(X_i|X_0, \ldots, X_{i-1})$. Within F_{i-1}, Lemma 8.5.32 yields $E(X_i) = E(E(X|F_i)) = E(X)$. This is the expectation within the event F_{i-1} (treated as a probability space), so all of these expresssions are conditioned on F_{i-1}, and the final expression is actually $E(X|F_{i-1}) = X_{i-1}$. ■

Such martingales, which we call **restriction martingales**, arise when we gradually discover a randomly generated object. Here F_i is the subset of the probability space where the object is confined after i steps (F for "inFormation"). In coin-flipping, the sample points are list of length n, and F_i may be the knowledge of the first i values. In random graphs, F_i may be the subgraph induced by the vertices $\{v_1, \ldots, v_i\}$, or F_i may be the knowledge of which among the first i edges are present.

To apply Azuma's Inequality, we need to bound $|X_i - X_{i-1}|$. The knowledge of which edges arise incident to a fixed vertex v_i can change the chromatic number by at most 1, since $\chi(G - v_i)$ equals $\chi(G)$ or $\chi(G) - 1$. From this we can conclude that $|X_i - X_{i-1}| \le 1$ in the restriction martingale defined by revealing vertices one by one.

8.5.36. Lemma. Consider a random structure specified by independent steps $S_1, \dots, S_n$. Let F_i be the knowledge of $S_1, \dots, S_i$, and let $X_0, \dots, X_n$ be the corresponding restriction martingale for a random variable X. Let A be the knowledge of S_j for all $j \ne i$, with S_i unknown. If for each such A the values of X on points in A differ by at most 1, then $|X_i - X_{i-1}| \le 1$ for all i (and hence $P(X - E(X)) > \lambda\sqrt{n}) \le e^{-\lambda^2/2}$).

Proof: Consider a particular instance of F_{i-1}, with $X_{i-1} = E(X|F_{i-1})$ given. We arrange the points of F_{i-1} in the cells of a grid. For all these points, the outcomes of $S_1, \dots, S_{i-1}$ are the same. Each row is a choice for F_i: a block in the partition of F_{i-1}. Each column is an A in which $S_{i+1}, \dots, S_n$ are fixed and only S_i varies. By hypothesis, in each column the maximum and minimum values of X differ by at most 1. Let m_s, M_s be the minimum and maximum of X in column s.

Choices of A ($S_{i+1}, \dots, S_n$ fixed within column)

Choices					
of F_i					
(or S_i)					

Because S_i and $S_{i+1}, \dots, S_n$ are specified independently, the probability of the outcome in row r and column s is $q_r p_s$, where q_r is the probability that S_i yields this row and p_s is the probability that $S_{i+1}, \dots, S_n$ yields this column. The computation of X_i is the expectation across a single row:

$$\sum m_s p_s \le E(X|F_i) \le \sum M_s p_s \le 1 + \sum m_s p_s.$$

Since these upper and lower bounds are independent of the row index, taking the expectation over the entire grid to compute X_{i-1} yields the same inequalities. Hence X_{i-1} and X_i are confined to a single interval of length 1 and differ by at most 1. Therefore, Azuma's Inequality applies. ■

When the conditions of Lemma 8.5.36 hold, we conclude immediately that the value of X is highly concentrated around its mean.

8.5.37. Example. *Chromatic number of random graphs.* Fix n, and consider Model A with edge probability p. Suppose we reveal the random n-vertex graph one vertex at a time. At stage i, we learn the edges from v_i to the previous vertices; this is S_i, and Model A specifies the outcomes of the S_i's independently. The event A in which all but S_i are specified is the subgraph $G - v_i$ of the random graph G plus the knowledge of edges from v_i to *later* vertices. Since $\chi(G - v_i) \le \chi(G) \le \chi(G - v_i) + 1$, the value of X differs by at most one over all possibilities in A. The hypotheses of Lemma 8.5.36 hold. Using both tails, we conclude that

$$P(|\chi(G) - E(\chi(G))|) \ge \lambda\sqrt{n}) \le 2e^{-\lambda^2/2}. \qquad \blacksquare$$

The result of Example 8.5.37 says nothing about the value of $E(\chi(G))$. To approximate this we again use Azuma's Inequality. With constant edge probability p, we know that the clique number of G^p is almost always within 1 of $d = 2\log_b n - 2\log_b \log_b n + 1 + 2\log_b(e/2)$, where $b = 1/p$. The same result holds for stable sets using the base $c = 1/(1-p)$ for the logarithm. To show that the chromatic number of G^p is close to $n/(2\log_c n)$, Bollobás showed that it is possible to extract stable sets of almost the maximum size until the number of vertices remaining is too small to matter.

8.5.38. Theorem. (Bollobás [1988]) For almost every G^p with constant $p = 1 - 1/c$, every induced subgraph of order at least $m = \lceil n/\log_c^2 n \rceil$ has a stable set of size at least $r = 2\log_c n - 5\log_c \log_c n$.

Proof: (sketch) We use r-**staset**, by analogy with r-clique, to mean a stable set of size r. Let S be a set of m vertices. We bound the probability that S has no r-staset by $e^{-dm^{1+\epsilon}}$ for some d, ϵ. This in turn bounds the probability that there exists an m-set with no r-staset by $\binom{n}{m} e^{-dm^{1+\epsilon}} < 2^n e^{-dm^{1+\epsilon}}$. Since $n = m^{1+o(1)}$, this bound goes to 0, and the first moment method implies that almost every G^p has no bad m-set.

It suffices to study the subgraph G induced by $[m]$. Let X be the maximum number of pairwise pair-disjoint r-stasets in this subgraph, where *pair-disjoint* means they share at most one vertex. We will show that $X \ge 1$ almost always. To do this, it suffices to show that (1) X is highly concentrated around its mean, and (2) $E(X)$ is bigger than something large (and growing).

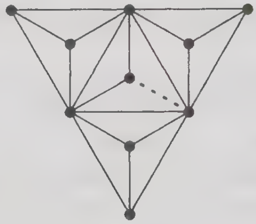
We invoke Azuma's Inequality for (1). Consider the restriction martingale for X that results from revealing G *one edge-slot at a time*. At each step, we learn whether one additional pair of vertices induces an edge. We have $X_0 = E(X)$ and $X_{\binom{m}{2}} = X$. The status of one edge slot changes the value of X by at most 1, so Lemma 8.5.36 applies, and $P(X - E(X)) \le -\lambda\binom{m}{2}^{1/2}) \le e^{-\lambda^2/2}$. With $\lambda = E(X)/\binom{m}{2}^{1/2}$, we have

$$P(X = 0) = P(X - E(X) \le -E(X)) \le e^{-E(X)^2/(m^2-m)}.$$

Hence it suffices to show that $E(X)/m \to \infty$.

To prove this, we consider another random variable $\hat{X}$, the number of r-stasets in G that have no pair in common with *any* other r-staset. Such a collection forms a pairwise pair-disjoint collection of r-stasets, so $X \ge \hat{X}$. We introduced X because the restriction martingale for $\hat{X}$ does not satisfy

$|\hat{X}_i - \hat{X}_{i-1}| \le 1$. In the drawing of $\overline{G}$ in the figure below, for example, we have $r = 4$ and seek 4-cliques; if the last (dotted) edge is present in $\overline{G}$ (absent in G), then $\hat{X} = 0$, but if it is absent from $\overline{G}$ (present in G), then $\hat{X} = 3$.



It is easier to compute $E(\hat{X})$ than $E(X)$. Expressing $\hat{X}$ as the sum of $\binom{m}{r}$ indicator variables, we obtain $E(\hat{X})$ as $\binom{m}{r}$ times the probability that $[r]$ induces an r-staset that is pair-disjoint from all others. This is $(1-p)^{\binom{r}{2}}$ times the conditional probability that $[r]$ does not conflict with other r-stasets, given the event Z that $[r]$ is in fact independent. Let Y be the number of other r-stasets overlapping $[r]$ in at least two elements. By Markov's Inequality, $E(Y|Z) \to 0$ implies $P(Y = 0|Z) \to 1$. Since each set counted shares at least two vertices with $[r]$, we have

$$E(Y|Z) = \textstyle\sum_{i \ge 2, r-1} \binom{r}{i}\binom{m-r}{r-i}(1-p)^{\binom{r}{2}-\binom{i}{2}}.$$

As $m \to \infty$, this tends to 0; this follows from the expression for r in terms of m. Hence $E(\hat{X})$ is asymptotic to $\binom{m}{r}(1-p)^{\binom{r}{2}}$. The expression for r in terms of m yields $E(\hat{X}) \in \Omega(m^{5/3})$. Thus $E(X)/m \to \infty$, which completes the proof. ■

8.5.39. Corollary. (Bollobás [1988]) For constant edge probability $p = 1 - 1/c$, almost every G^p satisfies

$$(1+\epsilon)n/(2\log_c n) \le \chi(G^p) \le (1+\epsilon')n/(2\log_c n),$$

where $\epsilon = \log_c \log_c n/\log_c n$ and $\epsilon' = 5\log_c \log_c n/\log_c n$.

Proof: The lower bound holds because almost every G^p has no stable set larger than $2\log_c n - 2\log_c \log_c n$. The upper bound follows from Theorem 8.5.38, because we can almost always select stable sets of size $2\log_c n - 5\log_c \log_c n$ until we have only $n/\lg_c^2 n$ vertices left. Since $n/\lg_c^2 n \in o(n/\log_c n)$, we can complete the coloring by using distinct new colors on the remaining vertices. ■

EXERCISES

8.5.1. (−) *Expectation.*

a) Compute the expected number of fixed points in a random permutation of $[n]$.

b) Determine the expected number of vertices of degree k in a random n-vertex graph with edge probability p.

8.5.2. (−) Prove that $1 - p < e^{-p}$ for $p > 0$.

8.5.3. (−) Determine the expected number of monochromatic triangles in a random 2-coloring of $E(K_6)$.

8.5.4. (−) Prove that some 2-coloring of the edges of $K_{m,n}$ has at least $\binom{m}{r}\binom{n}{s}2^{1-rs}$ monochromatic copies of $K_{r,s}$.

8.5.5. (−) The statement "$f(G_n) \le (1+\epsilon)n$" means that for all $\epsilon > 0$, the inequality holds for sufficiently large n. The statement "$f(G_n) \le n + o(n)$" means that $f(G_n)/n \to 1$ as $n \to \infty$. Prove that these two statements are equivalent.

8.5.6. Compute explicitly the probability that the Hamiltonian closure of a random graph with vertex set [5] is complete.

8.5.7. Let G be a graph with p vertices, q edges, and automorphism group of size s. Let $n = (sk^{q-1})^{1/p}$. Prove that some k-coloring of $E(K_n)$ has no monochromatic copy of G. (Chvátal–Harary [1973])

8.5.8. (!) a) Use a random partition of the vertices to prove that every graph has a bipartite subgraph with at least half its edges.

b) Use equipartitions of the vertices to improve part (a): if G has m edges and n vertices, then G has a bipartite subgraph with at least $m\frac{\lceil n/2\rceil}{2\lceil n/2\rceil-1}$ edges.

8.5.9. An army of computers is configured as a complete k-ary tree with leaves at distance l from the root. At a fixed time, each node is working with probability p, independently of other nodes. When a node is not working, the entire subtree below it is inaccessible. What is the expected number of nodes accessible from the root?

8.5.10. Let G be a matching of size n. Select a set of k vertices at random. Compute the expected number of edges induced by the selected vertices.

8.5.11. Consider a drawing in the plane of a simple graph G with n vertices and m edges, where $m \ge 4n$. Let H be a random induced subdrawing, generated by letting each vertex be retained with probability p, independently. Let Y be the number of edge crossings in H. Let $X = Y - [e(H) - (3n(H) - 6)]$. Use expectations to prove that $3n + p^3\nu(G) - pm > 0$, and conclude that $\nu(G) \ge m^3/[64n^2]$, where $\nu(G)$ is the minimum number of crossings in a drawing of G. (Comment: This is an alternative proof of Theorem 6.3.16.)

8.5.12. Given a random permutation of the vertices of a simple graph G, orient each edge toward the vertex with higher index in the permutation. Compute the expected number of sink vertices (outdegree 0) in the resulting orientation. In terms of $n(G)$, determine the minimum and maximum values of this expectation. Prove that the probably of having only one sink is at most $e(G)/\binom{n(G)}{2}$. (Jeurissen [1997])

8.5.13. (!) A **hypergraph** consists of a collection of vertices and a collection of edges; if the vertex set is V, then the edges are subsets of V. The **chromatic number** $\chi(H)$ of a hypergraph H is the minimum number of colors needed to label the vertices so that no edge is monochromatic. A hypergraph is k-**uniform** if its edges all have size k.

a) Prove that every k-uniform hypergraph with fewer than 2^{k-1} edges is 2-colorable. (Erdős [1963])

b) Use part (a) to prove that if each vertex of an n-vertex bipartite graph has a list of more than $1 + \lg n$ usable colors, then a proper coloring can be chosen from the lists.

8.5.14. (!) Use the deletion method to prove that a graph with n vertices and average degree $d \ge 1$ has an independent set with at least $n/(2d)$ vertices. (Hint: Choose a random subset by including each vertex independently with a probability p to be chosen later. Compute the expected number of edges induced.)

8.5.15. The maximum size of an n-vertex graph not containing H is $ex(n; H)$. Use the deletion method to prove that $ex(n; C_k) \in \Omega(n^{1+1/(k-1)})$. (Comment: One can also show that $ex(n; C_k) \in O(n^{1+2/k})$ by considering the average degree.) (Bondy–Simonovits)

8.5.16. (!) For $n \in \mathbb{N}$, prove that $R(k,k) > n - \binom{n}{k}2^{1-\binom{k}{2}}$. Use this to conclude that $R(k,k) > (1/e)(1 - o(1))k2^{k/2}$.

8.5.17. For natural numbers n, t, let $m = n - \binom{n}{t}^2 2^{1-t^2}$. Prove that there is a 2-coloring of the edges of $K_{m,m}$ with no monochromatic copy of $K_{t,t}$.

8.5.18. (+) *Off-diagonal Ramsey numbers.* Suppose that $0 < p < 1$.

a) Prove that if $\binom{n}{k}p^{\binom{k}{2}} + \binom{n}{l}(1-p)^{\binom{l}{2}} < 1$, then $R(k,l) > n$.

b) Prove that $R(k,l) > n - \binom{n}{k}p^{\binom{k}{2}} - \binom{n}{l}(1-p)^{\binom{l}{2}}$ for all $n \in \mathbb{N}$.

c) Choose n and p in part (b) to prove that $R(3,k) > k^{3/2-o(1)}$. What lower bound on $R(3,k)$ can be obtained from part (a)? (Spencer [1977])

8.5.19. Let H be a graph. For constant p, prove that almost every G^p contains H as an induced subgraph.

8.5.20. a) Fix k, s, t, p. Prove that almost every G^p has the following property: for every choice of disjoint vertex sets S, T of sizes s, t, there are at least k vertices that are adjacent to every vertex of S and no vertex of T. (Blass–Harary [1979])

b) Conclude that almost every G^p is k-connected.

c) Apply the same argument to random tournaments: almost every one has the property that for every choice of disjoint vertex sets S, T of sizes s, t, there are at least k vertices with edges to every vertex of S and from every vertex of T.

8.5.21. A random labeled tournament is generated by orienting each edge v_iv_j as $v_i \to v_j$ or $v_j \to v_i$ independently with probability $1/2$.

a) Prove that almost every tournament is strongly connected.

b) In a tournament, a "king" is a vertex such that every other vertex can be reached from it by a path of length at most 2. It is known that every tournament contains a king. Is it true that in almost every tournament every vertex is a king? (Palmer [1985])

8.5.22. Find a threshold probability function for the property that at least half the possible edges of a graph are present. How sharp is the threshold?

8.5.23. For $p = 1/n$ and fixed $\epsilon > 0$, show that almost every G^p has no component with more than $(1+\epsilon)n/2$ vertices. (Hint: Instead of trying to bound the probability directly, show that it is bounded by the probability of another event, which tends to 0.)

8.5.24. Determine the smallest connected simple graph that is not balanced.

8.5.25. Extend the second moment argument of Theorem 8.5.23 to prove that $n^{-1/\rho(H)}$ is a threshold function for the appearance of H as a subgraph of G^p, where $\rho(G) = \max_{G \subseteq H} e(G)/n(G)$. (Bollobás [1981a], Ruciński–Vince [1985])

8.5.26. Let Q be the following graph property: for every choice of disjoint vertex sets S, T of size $c \lg n$, there is an edge with endpoints in S and T. Prove that almost every graph has property Q if $c > 2$. (Comment: This implies that the random graph has bandwidth at least $n - 2\log n$.)

8.5.27. Prove that if $k = \lg n - (2+\epsilon)\lg\lg n$, then almost every n-vertex tournament has the property that every set of k vertices has a common successor.

8.5.28. A tournament is **transitive** if it has a vertex ordering $u_1, \ldots, u_n$ such that $u_i \to u_j$ if and only if $i < j$. Prove that every tournament has a transitive subtournament with $\lg n$ vertices, and almost every tournament has no transitive subtournament with more than $2\lg n + c$ vertices if c is a constant greater than 1.

8.5.29. (!) *The Coupon Collector.*

a) Consider repetitions of an experiment with independent success probability p. Prove that the expected number of the trial on which the first success occurs is $1/p$.

b) Every box of a certain type of candy contains one of n prizes, each with probability $1/n$. Receiving the grand prize requiries obtaining each of these prizes at least once. Prove that the expected number of the box on which the last prize is obtained is $n\sum_{i=1}^{n} 1/i$.

c) Prove that $m(n) = n\ln n + (k-1)n\ln\ln n$ is a threshold function for the number of boxes needed to obtain at least k copies of each prize. (Hint: Prove that when $p = o(1)$ and k is constant, the probability of at most k successes in m trials with success probability p is asymptotic to the probability of exactly k successes.)

8.5.30. Prove that the length of the longest run in a list of n random heads and tails is $(1+o(1))\lg n$. In other words, for $\epsilon > 0$, almost no list has at least $(1+\epsilon)\lg n$ consecutive identical flips, and almost every list has at least $(1-\epsilon)\lg n$ consecutive identical flips.

8.5.31. With $p = (1-\epsilon)\log n/n$, find a large m such that almost every graph has at least m isolated vertices. What $m(n)$ results from Chebyshev's Inequality?

8.5.32. Given a graph G, say that a k-set S is *bad* if G has no vertex v such that $S \subseteq N(v)$. For fixed p, how large can k be so that almost every G^p has no bad k-set? How slowly can k grow so that almost every G^p has a bad k-set?

8.5.33. By examining common neighbors, prove that if p is fixed and $k = o(n/\log n)$, then almost every G^p is k-connected.

8.5.34. (!) With $p = (1-\epsilon)\log n/n$, how large can m be such that almost every graph has at least m isolated vertices? (Hint: Use Chebyshev's Inequality.)

8.5.35. A **t-interval** is a subset of $\mathbb{R}$ that is the union of at most t intervals. The **interval number** of a graph G is the minimum t such that G is an intersection graph of t-intervals (each vertex is assigned a set that is the union of at most t intervals). Prove that almost all graphs (edge probability 1/2) have interval number at least $(1-o(1))n/(4\lg n)$. (Hint: Compare the number of representations with the number of simple graphs. Comment: Scheinerman [1990] showed that almost all graphs have interval number $(1+o(1))n/(2\lg n)$.) (Erdős–West [1985])

8.5.36. (!) *Threshold for perfect matching in a random bipartite graph.* Let G be a random subgraph of $K_{n,n}$ with partite sets A, B, generated by independent edge probability $p = (1+\epsilon)\ln n/n$, where ϵ is a nonzero constant. Call S a *violated set* if $|N(S)| < |S|$.

a) Prove that if $\epsilon < 0$, then almost every G has no perfect matching.

b) Let S be a minimal violated set. Prove that $|N(S)| = |S| - 1$ and that $G[S \cup N(S)]$ is connected.

c) Suppose that G has no perfect matching. Prove that A or B contains a violated set with at most $\lceil n/2 \rceil$ elements.

d) For $r, s \geq 1$, the number of spanning trees of $K_{r,s}$ is $r^{s-1}s^{r-1}$. Use this, part (b), part (c), and Markov's Inequality to prove that if $\epsilon > 0$, then G almost surely has a perfect matching. (Hint: A summation in the bound on the expected number of minimal violated sets can be bounded by a geometric series.)

8.5.37. Suppose that $0 < p < 1$ and that $k_1, \ldots, k_r$ are nonnegative integers summing to m. Prove that $\prod_{i=1}^{r}[1-(1-p)^{k_i}] \le [1-(1-p)^{m/r}]^r$.

8.5.38. *Tail inequality for binomial distribution.* Let $X = \sum X_i'$, where each X_i' is an indicator variable with success probability $P(X_i' = 1) = .5$, so $E(X) = n/2$. Applying Markov's Inequality to the random variable $Z = (X - E(X))^2$ yields $P(|Z| \ge t) \le Var(X)/t^2$. Setting $t = \alpha\sqrt{n}$ yields a bound on the tail probability: $P(|X - np| \ge \alpha\sqrt{n}) \le 1/(2\alpha^2)$. Use Azuma's Inequality to prove the stronger bound that $P(|X - np| > \alpha\sqrt{n}) < 2e^{-2\alpha^2}$. (Hint: Let $Y_i' = X_i' - .5$. Let F_i be the knowledge of $Y_1', \ldots, Y_i'$, and let $Y_i = E(Y|F_i)$.)

8.5.39. *Bin-packing.* Let the numbers $S = \{a_1, \ldots, a_n\}$ be drawn uniformly and independently from the interval $[0, 1]$. The numbers must be placed in bins, each having capacity 1. Let X be the number of bins needed. Use Lemma 8.5.36 to prove that $P(|X - E(X)|) \ge \lambda\sqrt{n}) \le 2e^{-\lambda^2/2}$.

8.5.40. (!) *Azuma's Inequality and the Traveling Salesman Problem.*

a) Prove the generalization of Azuma's Inequality to general martingales: If $E(X_i) = X_{i-1}$ and $|X_i - X_{i-1}| \le c_i$, then $P(X_n - X_0) \ge \lambda\sqrt{\sum c_i^2}) \le e^{-\lambda^2/2}$.

b) Let Y be the distance from a given point z in the unit square to the nearest of n points chosen uniformly and independently in the unit square. Prove that $E(Y) < c/\sqrt{n}$, for some constant c. (Hint: For a nonnegative continuous random variable Y, $E(Y) = \int_0^\infty P(Y \ge y)dy$, which can be verified using integration by parts. In order to bound this integral, use (somewhere) the inequality $1 - a < e^{-a}$ and the definite integral $\int_0^\infty e^{-t^2}dt = \sqrt{\pi}/2$.)

c) Apply parts (a) and (b) to prove that the smallest length of a polygon bounding a random set of n points in the unit square is highly concentrated around its expectation. In particular, the probability that this deviates from the expected tour length by more than $\lambda c\sqrt{\ln n}$ is bounded by $2e^{-\lambda^2/2}$, for some appropriate c. (Hint: For the martingale in which X_i is the expected length of the tour when the first i points are known, prove that $|X_i - X_{i-1}| < c(n-i)^{-1/2}$. Lemma 8.5.36 does not apply directly.)

8.6. Eigenvalues of Graphs

Techniques from group theory and linear algebra assist in studying the structure and enumeration of graphs.

From linear algebra, we have seen hints of vector spaces and determinants. In a graph G with edges $e_1, \ldots, e_m$, the **incidence vector** for a set $F \subseteq E(G)$ has coordinates $a_i = 1$ when $e_i \in F$ and $a_i = 0$ when $e_i \notin F$. Let **C** be the set of incidence vectors of even subgraphs (those with all vertex degrees even), and let **B** be the set of incidence vectors of edge cuts. Because these sets are closed under binary vector addition, **C** and **B** are vector spaces (Exercises 1–2), called the **cycle space** and **bond space** of G. Since an even subgraph and an edge cut share an even number of edges, **C** and **B** are orthogonal. This is closely related to the duality between cycles and bonds in Theorem 6.1.14 and Corollary 8.2.42 and to the use of determinants in the Matrix Tree Theorem (Theorem 2.2.12). For further discussion of these vector spaces, see Biggs [1993, Part 1].

Groups arise in studying graph isomorphism, embeddings, and enumeration. The automorphisms of a graph form a group of permutations of its vertices. Group-theoretic ideas lead to algorithms for testing isomorphism and to constructions for embedding on surfaces. Conversely, every group can be modeled using graphs. An introduction to this interplay appears in White [1973]; see also Gross–Yellen [1999, Chapters 13–15].

We restrict our attention to eigenvalues of adjacency matrices. We interpret the characteristic polynomial in terms of subgraphs, relate the eigenvalues to other graph parameters, and characterize the sets of eigenvalues for bipartite graphs and regular graphs. We close with applications to expander graphs and the "Friendship Theorem". An encyclopedic discussion of graph eigenvalues appears in Cvetković–Doob–Sachs [1979]. Chung [1997] presents the modern approach, modifying the adjacency matrix in a way that normalizes the eigenvalues and yields analogous results that hold more generally. For our brief presentation, we use the classical version.

THE CHARACTERISTIC POLYNOMIAL

8.6.1. Definition. The **eigenvalues** of a matrix A are the numbers λ such that $Ax = \lambda x$ has a nonzero solution vector; each such solution is an **eigenvector** associated with λ. The **eigenvalues** of a graph are the eigenvalues of its adjacency matrix A. These are the roots $\lambda_1, \ldots, \lambda_n$ of the **characteristic polynomial** $\phi(G;\lambda) = \det(\lambda I - A) = \prod_{i=1}^{n}(\lambda - \lambda_i)$. The **spectrum** is the list of distinct eigenvalues with their multiplicities $m_1, \ldots, m_t$; we write Spec $(G) = \binom{\lambda_1 \cdots \lambda_t}{m_1 \cdots m_t}$.

8.6.2. Remark. *Elementary properties of eigenvalues.*

0) The eigenvalues are the values λ such that the square matrix $\lambda I - A$ is singular, which is equivalent to $\det(\lambda I - A) = 0$.

1) $\sum \lambda_i = \text{Trace}\, A$. The **trace** is the sum of the diagonal elements and is the coefficient of λ^{n-1} in $\det(\lambda I - A)$. Since $\det(\lambda I - A) = \prod_{i=1}^{n}(\lambda - \lambda_i)$, that coefficient is also $\sum \lambda_i$. For simple graphs, it is 0.

2) $\prod \lambda_i = (-1)^n \phi(G; 0) = \det A = \sum_\sigma \text{sign}(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)}$, where the sum runs over permutations σ of $[n]$.

3) For a symmetric real n-by-n matrix A and $\lambda \in \mathbb{R}$, the multiplicity of λ as an eigenvalue of A is $n - \text{rank}(\lambda I - A)$.

4) Adding c to the diagonal shifts the eigenvalues by c, since $\alpha + c$ is a root of $\det(\lambda I - (cI + A))$ if and only if α is a root of $\det(\lambda I - A)$.

8.6.3. Example. *Spectra of cliques and bicliques.* The adjacency matrix of K_n is $J - I$, where J is the matrix of all 1s. Hence the eigenvalues of K_n are 1 less than those of J. Since Spec $J = \binom{n \quad 0}{1 \quad n-1}$, we have Spec $K_n = \binom{n-1 \quad -1}{1 \quad n-1}$.

The adjacency matrix of $K_{m,n}$ has rank 2, so it has two nonzero eigenvalues λ_1, λ_2. The trace is 0, so $\lambda_1 = -\lambda_2$; call this constant b. Hence $\phi(K_{m,n}, \lambda) = \lambda^n - b^2\lambda^{n-2}$. We compute b using $\phi(G; \lambda) = \det(\lambda I - A)$. Since λ appears only on the diagonal, contributions in the permutation expansion to the coefficient of λ^{n-2} arise only from permutations that use $n-2$ positions on the diagonal. The remaining two positions must be $-a_{i,j}$ and $-a_{j,i}$ for some i, j. There are mn nonzero contributions of this form, all negative. Hence $b^2 = mn$, and $\text{Spec}\,(K_{m,n}) = \binom{\sqrt{mn} \quad 0 \quad -\sqrt{mn}}{1 \quad m+n-2 \quad 1}$. ■

We index the coefficients of the characteristic polynomial so that $\phi(G; \lambda) = \sum_{i=0}^{n} c_i\lambda^{n-i}$. Since $\phi(G; \lambda) = \det(\lambda I - A)$, always $c_0 = 1$ and $c_1 = -\text{Trace}\, A = 0$. Our computation of c_2 for $K_{m,n}$ extends to all graphs.

8.6.4. Definition. A **principal submatrix** of a square matrix A is a submatrix selecting rows and columns with the same indices.

Since contributions to $c_2\lambda^{n-2}$ involve $n-2$ factors of λ from the diagonal, the coefficient c_2 is the sum of the principal 2×2 subdeterminants of $-A$. For a simple graph, $-a_{i,j}$ is -1 when $v_i \leftrightarrow v_j$ and 0 otherwise, so $c_2 = -e(G)$.

Similarly, c_3 is the sum of the principal 3×3 subdeterminants of $-A$. For triple i, j, k, the determinant depends only on the number of edges among v_i, v_j, v_k. The determinant is 0 unless they form a triangle, and then it is -2. Hence c_3 is -2 times the number of 3-cycles in G.

Since principal submatrices are the adjacency matrices of induced subgraphs, in general we have $c_i = (-1)^i \sum_{|S|=i} \det A(G[S])$.

8.6.5. Theorem. (Harary [1962b]) Given a simple graph G, let $\mathbf{H}$ be the set of spanning subgraphs in which every component is an edge or a cycle. If $k(H)$ and $s(H)$ denote the number of components of H and the number of components that are cycles, respectively, then $\det A(G) = \sum_{H\in\mathbf{H}}(-1)^{n(H)-k(H)}2^{s(H)}$.

Proof: The determinant formula is $\det A = \sum_\sigma (-1)^{t(\sigma)} \prod a_{i,\sigma(i)}$, where the sum is over permutations of $[n]$ and $t(\sigma)$ is the number of row exchanges (transpositions) needed to put the positions $i, \sigma(i)$ on the diagonal. When A is a 0,1-matrix, the contribution from σ is nonzero if and only if these entries all equal 1.

We view such a σ as a vertex permutation mapping each v_i to $v_{\sigma(i)}$. This partitions $V(G)$ into orbits. Since $a_{i,\sigma(i)} = 1$ means $v_i \leftrightarrow v_{\sigma(i)}$, there are no orbits of size 1, orbits of size 2 correspond to edges, and longer orbits correspond to cycles. Thus the permutation makes a nonzero contribution when it describes a spanning subgraph H of G in which the components are edges and cycles.

The sign of the contribution is determined by the number of transpositions needed to move the entries to the diagonal. Row exchanges move one element of an orbit at a time to the diagonal, but the last switch moves the last two elements to the diagonal. Hence $t(\sigma) = n(H) - k(H)$. Finally, each cycle of length at least 3 in H can appear in one of two ways in the permutation matrix, since we can follow the cycle in one of two directions. Hence the number of permutations that give rise to H is $2^{s(H)}$. ■

8.6.6. Corollary. (Sachs [1967]) Let $\mathbf{H}_i$ denote the collection of i-vertex subgraphs of a simple graph G whose components are edges or cycles. The characteristic polynomial of G is $\sum c_i\lambda^{n-i}$, where $c_i = \sum_{H\in\mathbf{H}_i}(-1)^{k(H)}2^{s(H)}$.

Proof: This follows from Theorem 8.6.5 and the earlier observation that $c_i = (-1)^i \sum_{|S|=i} \det A(G[S])$. ∎

This formula leads to a recursive expression for the characteristic polynomial (Exercise 5). The formula can be used to construct nonisomorphic trees with the same characteristic polynomial (and only eight vertices) (Exercise 7).

We next discuss the properties of eigenvalues for bipartite graphs.

8.6.7. Proposition. The (i, j)th entry of A^k counts the v_i, v_j-walks of length k. The eigenvalues of A^k are the kth powers of the eigenvalues of A.

Proof: The statement about walks holds easily by induction on k (Exercise 1.2.30). For the second statement, $Ax = \lambda x$ implies $A^k x = \lambda^k x$, by repeated multiplication. Using an arbitrary eigenvector x ensures that the multiplicities of the eigenvalues don't change. ∎

8.6.8. Lemma. If G is bipartite and λ is an eigenvalue of G with multiplicity m, then $-\lambda$ is also an eigenvalue with multiplicity m.

Proof: Adding isolated vertices to give the partite sets equal size merely adds rows and columns of 0's to the adjacency matrix, which does not change the rank and hence changes the spectrum only by including one extra 0 for each vertex added. Hence we may assume that the partite sets have equal sizes.

Since G is bipartite, we can permute the rows and columns of A to obtain the form $A = \begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix}$, where B is square. If λ is an eigenvalue associated with eigenvector $v = \begin{pmatrix} x \\ y \end{pmatrix}$ (partitioned according to the bipartition of G), then $\lambda v = Av = \begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} By \\ B^T x \end{pmatrix}$. Hence $By = \lambda x$ and $B^T x = \lambda y$.

Let $v' = \begin{pmatrix} x \\ -y \end{pmatrix}$. We compute $Av' = \begin{pmatrix} B(-y) \\ B^T x \end{pmatrix} = \begin{pmatrix} -\lambda x \\ \lambda y \end{pmatrix} = -\lambda v'$. Hence v' is an eigenvector of A for the eigenvalue $-\lambda$. Furthermore, m independent eigenvectors for λ yield m independent eigenvectors for $-\lambda$ in this way. Hence $-\lambda$ is an eigenvector of A with the same multiplicity as λ. ∎

8.6.9. Theorem. The following are equivalent statements about a graph G.

A) G is bipartite.

B) The eigenvalues of G occur in pairs λ_i, λ_j such that $\lambda_i = -\lambda_j$.

C) $\phi(G; \lambda)$ is a polynomial in λ^2.

D) $\sum_{i=1}^n \lambda_i^{2t-1} = 0$ for any positive integer t.

Proof: We proved A $\Rightarrow$ B in the lemma.

B $\Leftrightarrow$ C: $(\lambda - \lambda_i)(\lambda - \lambda_j) = (\lambda^2 - a)$ if and only if $\lambda_j = -\lambda_i$. Hence the roots occur in such pairs if and only if $\phi(G; \lambda)$ is a product of linear factors in λ^2.

B $\Rightarrow$ D: If $\lambda_j = -\lambda_i$, then $\lambda_j^{2t-1} = -\lambda_i^{2t-1}$.

D $\Rightarrow$ A: Because $\sum \lambda_i^k$ counts the closed k-walks in the graph (from each starting vertex), condition D forbids closed walks of odd length. This forbids odd cycles, since an odd cycle is an odd closed walk, and hence G is bipartite. ∎

LINEAR ALGEBRA OF REAL SYMMETRIC MATRICES

Relating eigenvalues to other parameters requires several results from linear algebra, including the Spectral Theorem and Cayley–Hamilton Theorem for real symmetric matrices. These are usually stated in more generality, but adjacency matrices are real and symmetric, and here the theorems have shorter proofs. We begin with a lemma that follows *from* the Spectral Theorem when the latter is proved using complex matrices. The proofs of these results may be skipped, especially by readers well-versed in linear algebra.

8.6.10. Lemma. If $f(x) = x^T Ax$, where A is a real symmetric matrix, then f attains its maximum and minimum over unit vectors x at eigenvectors of A, where it equals the corresponding eigenvalues.

Proof: The function f is continuous in $x_1, \dots, x_n$. For constrained optimization, we use Lagrangian multipliers. Given the constraint $x^T x = 1$, we let $g(x) = x^T x - 1$. Forming $L(x, \lambda) = f(x) - \lambda g(x)$, the extreme values occur where all partial derivatives of L are 0. With respect to λ, this yields $x^T x = 1$.

Let ∇ denote the vector of partial derivatives with respect to $x_1, \dots, x_n$. We compute $\nabla L(x, \lambda) = \nabla f(x) - \lambda \nabla g(x) = 2Ax - 2\lambda x$. The statement $\nabla f(x) = 2Ax$ uses the symmetry of A. We have $\nabla L = 0$ precisely when $Ax = \lambda x$, which requires x to be an eigenvector of A for eigenvalue λ. This yields $f(x) = x^T Ax = \lambda x^T x = \lambda$. ■

Since our variables in the optimization are real, we have found at least one real eigenvector and eigenvalue. We can use this inductively to show that all eigenvectors have this property.

8.6.11. Theorem. (Spectral Theorem) A real symmetric $n \times n$ matrix has real eigenvalues and n orthonormal eigenvectors.

Proof: We use induction on n. The claim is trivial for $n = 1$; consider $n > 1$. Let v_n be the eigenvector maximizing $x^T Ax$. Let W be the orthogonal complement of the space spanned by v_n; it has dimension $n - 1$. If $w \in W$, then $v_n^T Aw = w^T Av_n = \lambda_n w^T v_n = 0$. Hence $Aw \in W$. Viewing multiplication by A as a mapping f_A, we have $f_A \colon W \to W$.

Let S be a matrix whose columns are the vectors of an orthonormal basis of $\mathbb{R}^n$ with v_n as the last column. Since the basis is orthonormal, $S^{-1} = S^T$. The matrix for f_A with respect to this basis is $S^T AS$. Since the basis is orthonormal and v_n is an eigenvector, the last column of $S^T AS$ is 0, except for λ_n in the last position. Furthermore, the matrix is symmetric. Hence its first $n - 1$ rows and columns form the matrix A' for f_A on W with respect to this basis.

By the induction hypothesis, A' has orthonormal eigenvectors $v_1, \dots, v_{n-1}$, with real eigenvalues. Using S, we convert these back into real eigenvectors for A. They have the same real eigenvalues, and they form an orthonormal set. ■

Next we consider polynomial functions of a matrix. Viewed as members of $\mathbb{R}^{n^2}$, the matrices $I, A, A^2, \dots, A^{n^2}$ cannot be independent, since there are $n^2 + 1$

of them. Using an equation of linear dependence, we obtain a polynomial p such that $p(A)$ is the zero matrix. The characteristic polynomial itself suffices. This holds for all A, but again we consider only real symmetric matrices.

8.6.12. Theorem. (Cayley–Hamilton Theorem) If $\phi(\lambda)$ is the characteristic polynomial of a real symmetric matrix A, then $\phi(A)$ is the zero matrix (A "satisfies" its own characteristic polynomial).

Proof: Let the eigenvalues of A be $\lambda_1, \ldots, \lambda_n$, so $\phi(\lambda) = \prod_{i=1}^n (\lambda - \lambda_i)$. Since powers of A commute, the matrix polynomial obtained by using A for λ factors as $\phi(A) = \prod_{i=1}^n (A - \lambda_i I)$. To prove that $\phi(A) = 0$, we need only show that the matrix $\phi(A)$ maps every vector to 0. Write an arbitrary vector x as a linear combination of the basis of eigenvectors guaranteed by the Spectral Theorem. Applying $A - \lambda_i I$ kills the coefficient of v_i. Successively multiplying by all the factors $A - \lambda_i I$ produces the zero vector. ■

8.6.13. Definition. The **minimum polynomial** ψ of a matrix A is the polynomial of minimum degree satisfied by A and having leading coefficient 1. When A is the adjacency matrix of G, we call this the **minimum polynomial** $\psi(G; \lambda)$ of G.

The minimum polynomial is unique: if A satisfies two such polynomials of the same degree, then A satisfies their difference, which has lower degree.

8.6.14. Theorem. The minimum polynomial of A is $\psi(A) = \prod_{i=1}^t (\lambda - \lambda_i)$, where $\{\lambda_1, \ldots, \lambda_t\}$ are the distinct eigenvalues of A.

Proof: The minimum polynomial divides every polynomial satisfied by A, since otherwise the remainder would be a polynomial of lower degree satisfied by A. The Cayley–Hamilton Theorem now implies that ψ divides ϕ and must be the product of some of its factors. Killing the vectors in the subspace of eigenvectors for eigenvalue λ_i requires a factor of the form $A - \lambda_i I$. This factor kills all vectors in that subspace, so we only need one copy of each such factor. ■

8.6.15. Lemma. (Sylvester's Law of Inertia) Let A be a real symmetric matrix. If $x^T Ax$ can be written as a sum of N products of linear expressions, that is $x^T Ax = \sum_{m=1}^N (\sum_{i \in S_m} a_{i,m} x_i)(\sum_{j \in T_m} b_{j,m} x_j)$, then N is at least the maximum of the number of positive and the number of negative eigenvalues of A.

Proof: (Tverberg [1982]) Write the linear expressions as $u_m(x)$ and $v_m(x)$. For each m, we have $u_m(x)v_m(x) = L_m^2(x) - M_m^2(x)$, where $L = \frac{1}{2}(u + v)$ and $M = \frac{1}{2}(u - v)$ are also linear combinations of $x_1, \ldots, x_n$. This expresses the quadratic form as $x^T Ax = \sum_{m=1}^N [L_m^2(x) - M_m^2(x)]$.

On the other hand, A is a real symmetric matrix and thus has orthonormal eigenvectors $w^1, \ldots, w^n$. Using this, we write $x^T Ax = x^T S \Lambda S^T x$, where Λ is the diagonal matrix of eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$ and S has columns $w^1, \ldots, w^n$. If S has p positive and q negative eigenvalues, then this becomes $x^T Ax = \sum_{i=1}^p (y^i \cdot x)^2 - \sum_{i=n-q+1}^n (z^i \cdot x)^2$, where each y^i or z^i is $|\lambda_i|^{1/2} w^i$.

Now we consider a homogeneous system of linear equations. We require $L_m(x) = 0$ for $1 \le m \le N$, also $z^i \cdot x = 0$ for $n-q < i \le n$, and $w^i \cdot x = 0$ for $p < i \le n-q$. This places $N + n - p$ homogeneous linear constraints on n variables. If $N < p$, then these equations have a nonzero simultaneous solution x'. Setting x to x' in the two expressions for $x^T Ax$ yields $\sum_{i=1}^{p}(y^i \cdot x')^2 = -\sum_{m=1}^{N} M_m^2(x')$. Since x' is orthogonal to all eigenvectors with nonpositive eigenvalues, the left side is positive, while the right is nonpositive. The contradiction yields $N \ge p$; an analogous argument yields $N \ge q$. ■

EIGENVALUES AND GRAPH PARAMETERS

Eigenvalues provide bounds on various parameters, or alternatively graph parameters yield bounds on the eigenvalues. Our first result uses only the minimum polynomial.

8.6.16. Theorem. The diameter of a graph G is less than the number of distinct eigenvalues of G.

Proof: Let A be the adjacency matrix; A satisfies a polynomial of degree r if and only if some linear combination of $A^0, \dots, A^r$ is 0. Since the number of distinct eigenvalues is the degree of the minimum polynomial, we need only show that $A^0, \dots, A^k$ are linearly independent when $k \le \text{diam}\,(G)$.

It suffices to show for $k \le \text{diam}\,(G)$ that A^k is not a linear combination of $A^0, \dots, A^{k-1}$. Choose $v_i, v_j \in V(G)$ such that $d(v_i, v_j) = k$. By counting walks, we have $A^k_{i,j} \ne 0$ but $A^t_{i,j} = 0$ for $t < k$. Therefore, A^k is not a linear combination of the smaller powers. ■

Since the Spectral Theorem guarantees real eigenvalues, we can index our eigenvalues as $\lambda_1 \ge \cdots \ge \lambda_n$. We also refer to λ_1 and λ_n as $\lambda_{\max}(G)$ and $\lambda_{\min}(G)$.

8.6.17. Lemma. If G' is an induced subgraph of G, then

$$\lambda_{\min}(G) \le \lambda_{\min}(G') \le \lambda_{\max}(G') \le \lambda_{\max}(G)\ .$$

Proof: Since A is a real symmetric matrix, Lemma 8.6.10 yields $\lambda_{\min}(A) \le x^T Ax \le \lambda_{\max}(A)$ for every unit vector x. Consider the adjacency matrix A' of G'. By permuting the vertices of G, we can view A' as an upper left principal submatrix of $A = A(G)$. Let z' be the unit eigenvector of A' such that $A'z' = \lambda_{\max}(G')z'$. Let z be the unit vector in R_n obtained by appending zeros to z'. Then $\lambda_{\max}(G') = z'^T A'z' = z^T Az \le \lambda_{\max}(G)$. Similarly, $\lambda_{\min}(G') \ge \lambda_{\min}(G)$. ■

The behavior of the extreme eigenvalues under vertex deletion is a special case of the "Interlacing Theorem": If G has eigenvalues $\lambda_1 \ge \cdots \ge \lambda_n$ and $G - x$ has eigenvalues $\mu_1 \ge \cdots \ge \mu_{n-1}$, then $\lambda_1 \ge \mu_1 \ge \lambda_2 \ge \cdots \ge \mu_{n-1} \ge \lambda_n$. We will not need this and hence omit the proof, which involves only linear algebra.

8.6.18. Lemma. For every graph G, $\delta(G) \le \frac{2e(G)}{n(G)} \le \lambda_{\max}(G) \le \Delta(G)$.

Proof: Let x be an eigenvector for eigenvalue λ, and let $x_j = \max_i x_i$ be the largest coordinate value in x. Then $\lambda \le \Delta(G)$ follows from

$$\lambda x_j = (Ax)_j = \sum_{v_i \in N(v_j)} x_i \le d(v_j)x_j \le \Delta(G)x_j .$$

For the lower bound, we apply Lemma 8.6.10 to the unit vector with equal coordinates. Since the sum of the entries in the adjacency matrix is twice the number of edges of G, we have

$$\lambda_{\max} \ge \frac{\mathbf{1}_n^T}{\sqrt{n}} A \frac{\mathbf{1}_n}{\sqrt{n}} = \frac{1}{n} \sum\sum a_{ij} = \frac{2e(G)}{n}. \qquad \blacksquare$$

Lemma 8.6.18 enables us to improve the trivial bound $\chi(G) \le 1 + \Delta(G)$ given by the greedy coloring algorithm. Replacing $\Delta(G)$ with the average degree is too small; $K_n + K_1$ has chromatic number n and average degree less than $n-1$. Since $\lambda_{\max}$ is always at least the average degree, $1 + \lambda_{\max}(G)$ has a chance to work and can't be much improved.

8.6.19. Theorem. (Wilf [1967]) For every graph G, $\chi(G) \le 1 + \lambda_{\max}(G)$.

Proof: If $\chi(G) = k$, then we can successively delete vertices without reducing the chromatic number until we obtain a subgraph H such that $\chi(H - v) = k - 1$ for all $v \in V(H)$. As observed in Lemma 5.1.18, $\delta(H) \ge k - 1$. Since H is an induced subgraph of G, Lemma 8.6.18 and then Lemma 8.6.17 yield

$$k \le 1 + \delta(H) \le 1 + \lambda_{\max}(H) \le 1 + \lambda_{\max}(G) . \qquad \blacksquare$$

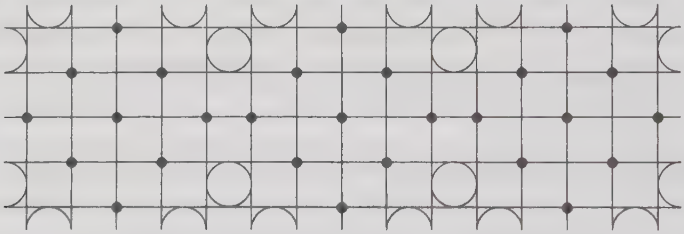
Sylvester's Law of Inertia yields a lower bound on the number of bicliques needed to decompose a graph. Because stars are bicliques and every subgraph of a star is a star, the number of bicliques needed is at most the vertex cover number $\beta(G) = n(G) - \alpha(G)$. Erdős conjectured that equality almost always holds, but this remains open. Graphs with special structure may have efficient partitions using other bicliques. The general lower bound using eigenvalues appears explicitly in Reznick–Tiwari–West [1985], but it is implicit in earlier work decomposing the complete graph (Tverberg [1982], Peck [1984]).

8.6.20. Theorem. For a simple graph G, the number of bicliques needed to decompose G is at least the maximum of the number of positive and number of negative eigenvalues of the adjacency matrix $A(G)$.

Proof: When G decomposes into subgraphs $G_1, \ldots, G_t$, we may write $A(G) = \sum_{i=1}^t B_i$, where B_i is the adjacency matrix of the spanning subgraph of G with edge set $E(G_i)$. When G_i is the biclique with bipartition S_i, T_i, we have $x^T B_i x = 2\sum_{j\in S_i} x_j \sum_{k \in T_i} x_k$. Writing these linear expressions as $u_i(x) = \sqrt{2}\sum_{j\in S_i} x_j$ and $v_i(x) = \sum_{k\in T_i} x_k$, we have $x^T Ax = \sum_{i=1}^t x^T B_i x = \sum_{i=1}^t u_i(x)v_i(x)$. Sylvester's Law of Inertia (Lemma 8.6.15) now yields the claimed lower bound. $\blacksquare$

8.6.21. Example. *Biclique decomposition of $C_{(2t+1)n} \,\square\, C_n$.* There are simple formulas for the eigenvalues of a cycle (Exercise 6) and for computing the eigenvalues of a cartesian product from the eigenvalues of the factors (Exercise 10). These yield simple formulas for the numbers of positive and negative eigenvalues of $C_m \,\square\, C_n$ when m is an odd multiple of n. In particular, $C_{(2t+1)n} \,\square\, C_n$ has $(2t+1)(n^2+1)/2$ positive eigenvalues and $(2t+1)(n^2-1)/2$ negative eigenvalues when n is odd (0 is not an eigenvalue).

Furthermore, such a product decomposes into $(2t+1)(n^2+1)/2$ bicliques, consisting of $(2t+1)(n-1)/2$ 4-cycles and $(2t+1)(n+1)/2$ stars (Kratzke–West). Note that 4-cycles and stars are the only subgraphs of $C_m \,\square\, C_n$ that are bicliques. The optimal decomposition of $C_{15} \,\square\, C_5$ appears below. Edges wrap around from top to bottom and right to left, and all grid points indicate vertices. The heavy dots indicate vertices that are centers of stars in the decomposition, and the circles indicate 4-cycles in the decomposition. ■



EIGENVALUES OF REGULAR GRAPHS

Like bipartite graphs, regular graphs can be characterized using spectra. The n-vector $\mathbf{1}_n$ with all coordinates 1 plays a special role in this and many other arguments involving eigenvalues, as does the matrix J of all 1s.

8.6.22. Theorem. The eigenvalue of G with largest absolute value is $\Delta(G)$ if and only if some component of G is $\Delta(G)$-regular. The multiplicity of $\Delta(G)$ as an eigenvalue is the number of $\Delta(G)$-regular components.

Proof: Let A be the adjacency matrix. The ith entry of $A\mathbf{1}_n$ is $d(v_i)$. When G is k-regular, we obtain $A\mathbf{1}_n = k\mathbf{1}_n$, and thus k is an eigenvalue with eigenvector $\mathbf{1}_n$. In general, let x be an eigenvector for eigenvalue λ, and let x_j be a coordinate of largest absolute value among coordinates of x corresponding to the vertices of some component H of G. For the jth coordinate of Ax, we have

$$|\lambda|\,|x_j| = |(Ax)_j| = \left|\sum_{v_i \in N(v_j)} x_i\right| \le d(v_j)\,|x_j| \le \Delta(G)\,|x_j|\,.$$

Hence $|\lambda| \le \Delta(G)$. Equality requires $d(v_j) = \Delta(G)$ and $x_i = x_j$ for all $v_i \in N(v_j)$. We can iterate this argument to reach all coordinates for vertices in H. Hence the eigenvalue associated with x has absolute value as large as $\Delta(G)$ only if H is $\Delta(G)$-regular.

Thus the eigenvalue associated with an eigenvector x has absolute value as large as $\Delta(G)$ if and only if (1) each component of G containing a vertex where x is nonzero is $\Delta(G)$-regular, and (2) x is constant on the coordinates corresponding to each such component. We can choose the constant independently for each $\Delta(G)$-regular component, so the dimension of the space of eigenvectors associated with $\Delta(G)$ is the number of $\Delta(G)$-regular components. ■

When G is connected and not regular, it remains true that eigenvalues of largest absolute value have multiplicity 1 and that coordinates of the associated eigenvector have the same sign. This is related to the Perron–Frobenius Theorem of linear algebra and uses arguments like those above; we omit the proof.

Powers of the adjacency matrix yield another characterization.

8.6.23. Theorem. (Hoffman [1963]) A graph G is regular and connected if and only if J is a linear combination of powers of $A(G)$.

Proof: *Sufficiency*. If J can be so expressed, then for each i, j we have $(A^k)_{ij} \neq 0$ for some $k \geq 0$, which requires a v_i, v_j-walk of length k. Hence G is connected. For regularity, consider the matrices JA and AJ. The i, jth position of AJ is $d(v_i)$ (constant on rows), and the i, jth position of JA is $d(v_j)$ (constant on columns). Since J is a linear combination of powers of A, each of which commutes with A, we have $JA = AJ$. Thus the i, jth position is both $d(v_i)$ and $d(v_j)$ and the graph is regular.

Necessity. Since G is k-regular, k is an eigenvalue, and the minimum polynomial is $\psi(G; \lambda) = (\lambda - k)g(\lambda)$ for some polynomial g. Since $\psi(G; A) = 0$, we have $Ag(A) = kg(A)$. Hence each column of $g(A)$ is an eigenvector of A with eigenvalue k. Since G is regular and connected, each such eigenvector is a multiple of $\mathbf{1}_n$. Hence the columns of $g(A)$ are constant. However, $g(A)$ is a linear combination of powers of a symmetric matrix and therefore must itself be symmetric. Hence the columns are equal and $g(A)$ is a multiple of J. ■

When G is simple and regular, $\overline{G}$ is also regular, and the eigenvalues of $\overline{G}$ can be obtained from the eigenvalues of G. This rests on the matrix expression for complementation: $A(\overline{G}) = J - I - A(G)$.

8.6.24. Lemma. $\phi(\overline{G}; \lambda) = (-1)^n \det[(-\lambda - 1)I - A(G) + J]$.

Proof: Direct computation yields $\det(\lambda I - A(\overline{G})) = \det(\lambda I - (J - I - A)) = \det[(\lambda + 1)I - J + A] = (-1)^n \det[(-\lambda - 1)I - A + J]$. ■

8.6.25. Theorem. If a simple graph G is k-regular, then G and $\overline{G}$ have the same eigenvectors. The eigenvalue associated with $\mathbf{1}_n$ is k in G and $n - k - 1$ in $\overline{G}$. If $x \neq \mathbf{1}_n$ is an eigenvector of G for eigenvalue λ of G, then its associated eigenvalue in $\overline{G}$ is $-1 - \lambda$.

Proof: Since $\overline{G}$ is $n - k - 1$-regular, $\mathbf{1}_n$ is an eigenvector for both G and $\overline{G}$, with eigenvalue k for G and $n - k - 1$ for $\overline{G}$. Let x be another eigenvector of G in an orthonormal basis of eigenvectors, and let $\overline{A} = A(\overline{G})$. Since $\mathbf{1}_n \cdot x = 0$, $\sum x_i = 0$. We compute $\overline{A}x = Jx - x - Ax = 0 - x - Ax = (-1 - \lambda)x$. ■

This yields a lower bound on the smallest eigenvalue of a regular graph and another derivation of the spectrum of K_n.

8.6.26. Corollary. For a k-regular simple graph, $\lambda_n \geq k - n$.

Proof: If G is k-regular and $\lambda_1 \geq \cdots \geq \lambda_n$, then the eigenvalues of $\overline{G}$ are $(n-k-1, -1-\lambda_n, \ldots, -1-\lambda_2)$, by Theorems 8.6.22–8.6.25. In particular, $n-k-1 \geq -\lambda_n - 1$. ■

The eigenvalues of a connected regular simple graph G can be used to count its spanning trees. The eigenvalues need not be rational, yet the result $\tau(G)$ is an integer. The Matrix Tree Theorem (Theorem 2.2.12) says that $\tau(G)$ equals each minor of $Q = D - A$, where A is the adjacency matrix and D is the diagonal matrix of degrees. When G is k-regular, $D = kI$. Letting Adj Q denote the adjugate matrix of Q (the matrix of signed cofactors), the Matrix Tree Theorem is the statement that $\operatorname{Adj} Q = \tau(G)J$. Using Cayley's Formula (Theorem 2.2.3) for spanning trees in K_n, we have $\operatorname{Adj}(nI - J) = n^{n-2}J$.

8.6.27. Lemma. Let D be the diagonal matrix of vertex degrees in a simple graph G, let $A = A(G)$, and let $Q = D - A$. The number of spanning trees of G is $\tau(G) = \det(J + Q)/n^2$.

Proof: Observe that $J^2 = nJ$, $JQ = 0$, and $\operatorname{Adj}(AB) = \operatorname{Adj}(A)\operatorname{Adj}(B)$. We apply this using $J + Q$ and the matrix $nI - J$ that arises from K_n. We have

$$\operatorname{Adj}(nI - J)\operatorname{Adj}(J + Q) = \operatorname{Adj}[(nI - J)(J + Q)] = \operatorname{Adj}(nQ),$$

since $J^2 = nJ$ and $JQ = 0$. We have computed that $\operatorname{Adj}(nI - J) = n^{n-2}J$. Also, $\operatorname{Adj}(nQ) = n^{n-1}\operatorname{Adj} Q$ for any matrix Q. Canceling common factors of n yields $J\operatorname{Adj}(J + Q) = n\tau(G)J$. Multiplying both sides of this on the right by $(J + Q)^T$ yields $J(\det(J + Q)I) = n\tau(G)nJ$. Both sides are multiples of J, so the desired equality holds. ■

We can now compute $\tau(G)$ from the eigenvalues if G is regular. (This analysis extends to all graphs when the modified system of eigenvalues is used.)

8.6.28. Theorem. If G is a k-regular connected simple n-vertex graph with spectrum $\binom{k \;\; \lambda_2 \cdots \lambda_t}{1 \;\; m_2 \cdots m_t}$, then $\tau(G) = n^{-1}\phi'(G; k) = n^{-1}\prod_{j=2}^{t}(k - \lambda_j)^{mj}$.

Proof: Since $J + Q = J + kI - A$, the determinant of $J + Q$ is the value at k of the characteristic polynomial of $A - J$. Since G is k-regular and connected, it has $\mathbf{1}_n$ as an eigenvector with eigenvalue k, and the other eigenvectors are orthogonal to $\mathbf{1}_n$. Every such eigenvector of A is also an eigenvector of $A - J$, with the same eigenvalue, since $(A - J)x = Ax - Jx = Ax = \lambda x$.

Also, $\mathbf{1}_n$ is an eigenvector of $A - J$ with eigenvalue $k - n$. This produces a full set of eigenvalues for $A - J$. Evaluating the characteristic polynomial at k yields $\det(J + Q) = n\prod_{j=2}^{t}(k - \lambda_j)$. The product is $\phi'(G; k)$, since $\phi(G; \lambda)$ has $\lambda - k$ as a non-repeated factor when G is k-regular and connected. By Lemma 8.6.27, we obtain $\tau(G)$ upon dividing by n^2. ■

The results in Lemma 8.6.24–Theorem 8.6.28 were extended to arbitrary (non-regular) graphs in Kelmans [1967b] (also Kelmans–Chelnokov [1974]) using the eigenvalues of the Laplacian matrix of the graph. This is the matrix Q used above. Another method for counting spanning trees appears in Kelmans [1965, 1966], and another variation on the Matrix Tree Theorem appears in Hartsfield–Kelmans–Shen [1996].

EIGENVALUES AND EXPANDERS

Many applications in computer science require "expander graphs". Walters [1996] collects many definitions that have been used for such graphs. The basic notion of expansion is that all small sets should have large neighborhoods. The aim is to establish good connectivity properties without having many edges.

8.6.29. Definition. An (n, k, c)-**expander** is an X, Y-bigraph G with $|X| = |Y| = n$ such that $\Delta(G) \leq k$ and that $|N(S)| \geq (1 + c(1 - |S|/n)) \cdot |S|$ for every $S \subseteq X$ with $|S| \leq n/2$. An (n, k, c)-**magnifier** is an n-vertex graph G such that $\Delta(G) \leq k$ and that $\left|N(S) \cap \overline{S}\right| \geq c \cdot |S|$ for every $S \subseteq V(G)$ with $|S| \leq n/2$. An n-**superconcentrator** is an acyclic digraph with n sources and n sinks such that for every set A of sources and every set B of $|A|$ sinks, there are $|A|$ disjoint A, B-paths.

Expanders appear in the parallel sorting network of Ajtai, Komlós, and Szemerédi [1983]. The condition for expansion strengthens Hall's Condition; we have not one matching but many. This facilitates using expanders to construct superconcentrators. Applications of superconcentrators are discussed in Alon [1986a]. The bound on maximum degree makes the number of edges linear in n, thereby limiting the cost of constructing the network.

Probabilistic methods (Exercise 22) yield the *existence* of expanders (and superconcentrators) with large n and bounded average degree (Pinsker [1973]), Pippenger [1977], Chung [1978b]). Margulis [1973] used algebraic ideas to construct an explicit example (see also Gabber–Galil [1981]).

Although an appropriately generated random graph will almost always have good expansion properties, it is hard to measure expansion. Tanner [1984] and Alon–Milman [1984, 1985] independently used eigenvalues to remedy this. They proved that graphs have good expansion properties when the two largest eigenvalues are far apart. Since eigenvalues are easy to compute (or approximate), we can generate a graph randomly and then compute its eigenvalues to check the amount of expasion.

We consider only the special case of regular graphs. Expanders are more useful than magnifiers in applications, but it is easy to obtain an $(n, (k+1), c)$-expander from an (n, k, c)-magnifier (Exercise 21). Hence we consider the relationship between eigenvalues and magnification. Our presentation follows that of Alon–Spencer [1992, p119ff], which discusses additional properties of the eigenvalues of regular (and random) graphs.

8.6.30. Theorem. If G is a k-regular n-vertex graph with second-largest eigenvalue λ, and S is a nonempty proper subset of $V(G)$, then

$$\left|[S, \overline{S}]\right| \geq (k - \lambda)\,|S|\,\left|\overline{S}\right|/n.$$

Proof: Since G is k-regular, $\lambda_{\max}(G) = k$. The claim is trivial if $k - \lambda = 0$, so we may assume that G is connected. We compute

$$x^T(kI - A)x = k\sum x_i^2 - 2\sum_{ij\in E(G)} x_i x_j = \sum_{ij\in E(G)} (x_i - x_j)^2.$$

Now let $s = |S|$ and set $x_i = -(n - s)$ for $i \in S$ and $x_i = s$ for $i \notin S$. The sum on the right above becomes $n^2\left|[S, \overline{S}]\right|$.

Because $|S| = s$ implies $\sum x_i = 0$, the vector x is orthogonal to the eigenvector $\mathbf{1}_n$ of A with eigenvalue k. The eigenvector $\mathbf{1}_n$ is also the eigenvector of $kI - A$ for its smallest eigenvalue 0. Using Lemma 8.6.10 and Theorem 8.6.11, the minimum of $\frac{x^T(kI-A)x}{x^Tx}$ over vectors orthogonal to 1_n is the next smallest eigenvalue of $kI - A$, which is $k - \lambda$. Hence

$$x^T(kI - A)x \geq (k - \lambda)x^Tx = (k - \lambda)\left(s(n - s)^2 + (n - s)s^2\right) = (k - \lambda)s(n - s)n.$$

Since $x^T(kI - A)x = n^2\left|[S, \overline{S}]\right|$, we have $\left|[S, \overline{S}]\right| \geq (k - \lambda)s(n - s)/n$. ■

8.6.31. Corollary. If G is a k-regular n-vertex graph with second-largest eigenvalue λ, then G is an (n, k, c)-magnifier, where $c = (k - \lambda)/2k$.

Proof: If S is a set of $s \leq n/2$ vertices in G, then Theorem 8.6.30 yields $\left|[S, \overline{S}]\right| \geq$ $k - \lambda)s(n - s)/n$. Each vertex of $\overline{S}$ receives at most k of these edges, so S must have at least $(k - \lambda)s(n - s)/(nk)$ neighbors in $\overline{S}$. Since $(n - s)/n \geq 1/2$, the result follows. ■

Greater separation between the two largest eigenvalues yields greater magnification. Alon and Milman [1984] improved the lower bound to $c \geq (2k-2\lambda)/(3k-2\lambda)$. Alon [1986b] proved a partial converse: If a k-regular graph G is an (n, k, c)-magnifier, then the separation $k - \lambda$ is at least $c^2/(4 + 2c^2)$.

Explicit constructions of regular graphs are known with separation between λ_1 and λ_2 nearly as large as possible. The second largest eigenvalue of a k-regular graph with diameter d is at least $2\sqrt{k} - 1(1 - O(1/d))$ (see Nilli [1991]). Lubotzky–Phillips–Sarnak [1986] and Margulis [1988] constructed infinite families of regular graphs where the degree k is 1 more than a prime congruent to 1 mod 4 and the second largest eigenvalue is at most $2\sqrt{k} - 1$.

STRONGLY REGULAR GRAPHS

We close with an application to a special class of regular graphs.

8.6.32. Definition. A simple n-vertex graph G is **strongly regular** if there are parameters k, λ, μ such that G is k-regular, every adjacent pair of vertices have λ common neighbors, and every nonadjacent pair of vertices have μ common neighbors.

Properties of eigenvalues of strongly regular graphs provide a short proof of a curious result called the "Friendship Theorem". This theorem says that at any party at which every pair of people have exactly one common acquaintance, there is one person who knows everyone (presumably the host). The resulting graph of the acquaintance relation consists of some number of triangles sharing a vertex. Another motivation for studying strongly regular graphs is their connection with the theory of designs. Strongly regular graphs with $\lambda = \mu$ correspond to symmetric balanced incomplete block designs. Other regular graphs with rich algebraic structure appear in Biggs [1993, part 3].

8.6.33. Theorem. If G is a strongly regular graph with n vertices and parameters k, λ, μ, then $\overline{G}$ is strongly regular with parameters $k' = n - k - 1$, $\lambda' = n - 2 - 2k + \mu$, and $\mu' = n - 2k + \lambda$.

Proof: For each adjacent pair $v \leftrightarrow w$ in G, there are $2(k-1) - \lambda$ other vertices in $N(v) \cup N(w)$, so v and w have $n - 2 - 2(k-1) + \lambda$ common nonneighbors. When $v \not\leftrightarrow w$ there are $2k - \mu$ vertices in $N(v) \cup N(w)$ and thus $n - 2k + \mu$ common nonneighbors. ■

8.6.34. Theorem. If G is a strongly regular graph with n vertices and parameters k, λ, μ, then $k(k - \lambda - 1) = \mu(n - k - 1)$.

Proof: We count induced copies of P_3 with a fixed vertex v as an endpoint. The middle vertex w can be picked in k ways. For each such w, the third vertex can be any neighbor of w not adjacent to v. With v unavailable, there are always $k - \lambda - 1$ ways to pick the third vertex. On the other hand, the third vertex can be picked in $n - k - 1$ ways as a nonneighbor of v, and for each such choice there are μ common neighbors with v that can serve as w. ■

8.6.35. Example. *Degenerate cases:* $\mu = 0$ *or* $\lambda = k - 1$ *or* $k = n - 1$. We show that such a strongly regular graph is a disjoint union of $k+1$-cliques. By Theorem 8.6.34, $\lambda = k - 1$ if and only if $\mu = 0$ or $k = n - 1$. Hence we may assume that $\lambda = k - 1$. Now every neighbor of v is adjacent to every other, which forbids an induced P_3 and forces G to be a disjoint union of cliques. ■

Henceforth, we assume that $\mu > 0$ and $\lambda < k - 1$. Theorem 8.6.34 states a necessary condition on the set of parameters for a strongly regular graph. Another necessary condition arises from the eigenvalues.

8.6.36. Theorem. (Integrality Condition) If G is strongly regular with n vertices and parameters k, λ, μ, then the two numbers below are nonnegative integers.

$$\frac{1}{2}\left(n - 1 \pm \frac{(n-1)(\mu - \lambda) - 2k}{\sqrt{(\mu - \lambda)^2 + 4(k - \mu)}}\right)$$

Proof: These are nonnegative integers because they are multiplicities of eigenvalues. Consider A^2. The ijth entry of A^2 is k if $i = j$, is λ if $v_i \leftrightarrow v_j$, and is μ if $v_i \not\leftrightarrow v_j$. Since $v_i \leftrightarrow v_j$ marks the 1s in the adjacency matrix and $v_i \not\leftrightarrow v_j$ marks the 1s in the adjacency matrix of the complement, we have $A^2 = kI + \lambda A + \mu(J - I - A)$. Rearranging terms yields $A^2 = (k-\mu)I + (\lambda-\mu)A + \mu J$.

Multiplying $\mathbf{1}_n$ by both expressions for A^2 yields

$$k^2\mathbf{1}_n = (k-\mu)\mathbf{1}_n + (\lambda-\mu)k\mathbf{1}_n + \mu n\mathbf{1}_n,$$

which yields another proof of $k(k-\lambda-1) = \mu(n-k-1)$. Let x be an eigenvector for another eigenvalue $\theta \neq k$. Since x is orthogonal to $\mathbf{1}_n$, we have $Jx = \mathbf{0}_n$. Multiplying x by both expressions for A^2 produces $\theta^2 - (\lambda-\mu)\theta - (k-\mu) = 0$. This quadratic equation for θ has two roots r, s, which must be the values of all the other eigenvalues. The values are $\frac{1}{2}\left(\lambda - \mu \pm \sqrt{(\lambda-\mu)^2 + 4(k-\mu)}\right)$.

Now let a and b be the multiplicities of the eigenvalues r and s. Example 8.6.35 describes all cases when $\mu = 0$. Hence we may assume that $\mu > 0$, and thus G is connected. Because G is connected, eigenvalue k has multiplicity 1, and we have $1+a+b = n$. Since the eigenvalues sum to 0, we have $k+ra+sb = 0$. The solution to these two linear equations for a and b is $a = -\frac{k+s(n-1)}{r-s}$ and $b = \frac{k+r(n-1)}{r-s}$. These are the values claimed above to be nonnegative integers. ■

The argument above can also be traced in the opposite direction.

8.6.37. Theorem. A k-regular connected graph G is strongly regular with parameters k, λ, μ if and only if it has exactly three eigenvalues $k > r > s$ and these satisfy $r + s = \lambda - \mu$ and $rs = -(k-\mu)$. ■

8.6.38. Example. *Classes of strongly regular graphs.* We consider two cases: $(n-1)(\mu-\lambda) = 2k$ and $(n-1)(\mu-\lambda) \neq 2k$. Excluding the trivial values, the first case requires $\mu = \lambda + 1$, because $0 < 2k < 2n-2$. By Theorem 8.6.33, G and $\overline{G}$ are thus strongly regular graphs with the same parameters. In this case, we also know that $n = 4\mu+1$ and that n is the sum of two perfect squares. Furthermore, the eigenvalues r and s have the same multiplicity.

In the second case, rationality requires $(\mu-\lambda)^2 + 4(k-\mu) = d^2$ for some positive integer d, and d must divide $(n-1)(\mu-\lambda) - 2k$. Here the eigenvalues must be integers. Various such examples are known. In the special case $\lambda = 0$ and $\mu = 2$, three such graphs are known, but it is not known whether the list is finite! The known examples, listing the parameters (n, k, λ, μ), are the square $(4, 2, 0, 2)$, the Clebsch graph $(16, 5, 0, 2)$, and the Gewirtz graph $(56, 10, 0, 2)$ (see Cameron–van Lint [1991], p43). The Clebsch graph arises in Exercise 23. Other strongly regular graphs appear in Exercises 24–26. ■

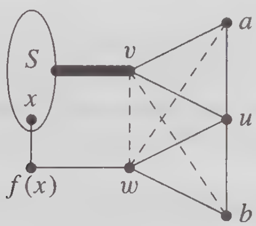
Finally, we prove the Friendship Theorem. It is startling that such a combinatorial-sounding result seems to have no short combinatorial proof. There do exist proofs avoiding eigenvalues (see Hammersley [1983]), but they require complicated numerical arguments to eliminate regular graphs.

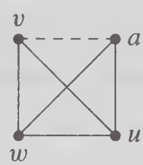
8.6.39. Theorem. (Friendship Theorem—Wilf [1971]). If G is a graph in which any two distinct vertices have exactly one common neighbor, then G has a vertex joined to all others.

Proof: The symmetry of the condition suggests that G might be regular. If G is regular, then it is strongly regular with $\lambda = \mu = 1$. By Theorem 8.6.36, $\frac{1}{2}(n-1 \pm k/\sqrt{k-1})$ now must be an integer. Hence $k/\sqrt{k-1}$ is an integer, which happens only when $k = 2$. However, K_3 is the only 2-regular graph satisfying the condition, and it does have vertices of degree $n-1$.

Now suppose that G is not regular. We show that $v \not\leftrightarrow w$ requires $d(v) = d(w)$. Insistence on unique common neighbors forbids 4-cycles. Let u be the common neighbor of $\{v, w\}$. Let a be the common neighbor of $\{u, v\}$, and let b be the common neighbor of $\{u, w\}$. Every $x \in S = N(v) - \{u, a\}$ has a common neighbor $f(x)$ with w. If $f(x) = b$ for some $x \in S$, then x, b, u, v is a 4-cycle. If $f(x) = f(x')$ for distinct $x, x' \in S$, then $x, v, x', f(x)$ is a 4-cycle. We have thus shown that $d(w) \geq d(v)$. By symmetry, $d(v) \geq d(w)$.

Since G is not regular, it has two vertices v, w with $d(w) \neq d(v)$. By the preceding paragraph, $v \leftrightarrow w$. Let u be their common neighbor. Since u cannot have the same degree as each of them, we may assume that $d(u) \neq d(v)$. If G has a vertex $x \notin N(v)$, then $d(x) = d(v)$, but this requires $x \leftrightarrow w$ and $x \leftrightarrow u$. This creates the 4-cycle v, u, x, w. Hence $d(v) = n - 1$. ■





EXERCISES

8.6.1. *Interpretation of cycle space and bond space.* Given a graph G, prove that
a) The symmetric difference of two even subgraphs is an even subgraph.
b) The symmetric difference of two edge cuts is an edge cut, and
c) Every edge cut shares an even number of edges with every even subgraph.

8.6.2. *Dimension of cycle space and bond space.* By parts (a) and (b) of Exercise 8.6.1, the cycle space $\mathbf{C}$ and bond space $\mathbf{B}$ of a graph G are binary vector spaces. Prove that when G is connected, $\mathbf{C}$ has dimension $e(G) - n(G) + 1$ and $\mathbf{B}$ has dimension $n(G) - 1$. (Hint: Show that the cycles created by adding one edge to a particular spanning tree form a basis for the cycle space. Show that $n(G) - 1$ bonds that isolate single vertices form a basis for the bond space, or use orthogonality.)

8.6.3. Recall that the *closed neighborhood* of a vertex v is $N(v) \cup \{v\}$.

a) Let S be a set of vertices in a simple graph G whose neighborhoods are identical. Prove that some eigenvalue of G has multiplicity at least $|S| - 1$. What is it?

b) Let S be a set of vertices in a simple graph G whose closed neighborhoods are identical. Prove that some eigenvalue of G has multiplicity at least $|S| - 1$. What is it?

8.6.4. Let σ_k be the number of subgraphs of a graph G that are k-cycles. Let $L_k = \sum \lambda_i^k$ and $D_k = \sum d_i^k$ be the sums of the kth powers of the eigenvalues and the vertex degrees. Obtain formulas for σ_3 and σ_4 in terms of $\{L_k\}$ and $\{D_k\}$.

8.6.5. *Deletion formulas for the characteristic polynomial.* For clarity in this problem, we write $\phi(G; \lambda)$ as ϕ_G. Let v $[xy]$ be an arbitrary vertex [edge] of G, and let $Z(v)$ $[Z(xy)]$ be the collection of cycles containing v $[xy]$. Prove that the characteristic polynomial satisfies the following recurrences.

a) $\phi_G = \lambda\phi_{G-v} - \sum_{u\in N(v)} \phi_{G-v-u} - 2\sum_{C\in Z(v)} \phi_{G-V(C)}$.

b) $\phi_G = \phi_{G-xy} - \phi_{G-x-y} - 2\sum_{C\in Z(xy)} \phi_{G-V(C)}$.

(Hint: Induction or Sach's formula can be used. Also, the edge-deletion formula can be proved from the vertex-deletion formula. Comment: When G is a forest and v is a leaf with neighbor u, the formulas reduce to $\phi_G = \lambda\phi_{G-v} - \phi_{G-v-u}$ and $\phi_G = \phi_{G-xy} - \phi_{G-x-y}$.)

8.6.6. *Characteristic polynomial for paths and cycles.*

a) Use Exercise 8.6.5 to find recurrences for $\phi(P_n; \lambda)$ and for $\phi(C_n; \lambda)$.

b) Without solving the recurrence, prove that $\{2\cos(2\pi j/n)\colon 0 \le j \le n-1\}$ are the eigenvalues of C_n.

c) Given Spec (C_n), compute Spec G, where G is the graph obtained from C_n by adding edges joining vertices at distance 2 in C_n.

8.6.7. For a tree, prove that the coefficient of λ^{n-2k} in the characteristic polynomial is $(-1)^k\mu_k(G)$, where $\mu_k(G)$ is the number of matchings of size k. Use this to construct a pair of nonisomorphic "co-spectral" 8-vertex trees; both have characteristic polynomial $\lambda^8 - 7\lambda^6 + 9\lambda^4$. (Comment: As $n \to \infty$, almost no trees are uniquely determined by their spectra.) (Schwenk [1973])

8.6.8. (+) Let T be a tree. Prove that $\alpha(T)$ is the number of nonnegative eigenvalues of T. (Hint: See Theorem 8.6.20.) (Cvetković–Doob–Sachs [1979, p233])

8.6.9. Let λ be an eigenvalue of a graph G with n vertices and m edges. Prove that $|\lambda| \le \sqrt{2m(n-1)/n}$.

8.6.10. Let $\lambda_1, \ldots, \lambda_m$ and $\mu_1, \ldots, \mu_n$ be the eigenvalues of G and H, respectively. Show that the mn eigenvalues of $G \square H$ are $\{\lambda_i + \mu_j\}$. Use this to derive the spectrum of the k-cube. (Hint: Given an eigenvector of $A(G)$ associated with λ_i and an eigenvector of $A(H)$ associated with μ_j, construct an eigenvector for $A(G \square H)$ associated with $\lambda_i + \mu_j$.)

8.6.11. Compute the spectrum of the complete p-partite graph $K_{m,\ldots,m}$. (Hint: Use the expression $A(\overline{G}) = J - I - A(G)$ for the adjacency matrix of the complement.)

8.6.12. Given $\phi(G; x) = x^8 - 24x^6 - 64x^5 - 48s^4$, determine G.

8.6.13. (!) Prove that G is bipartite if G is connected and $\lambda_{\max}(G) = -\lambda_{\min}(G)$.

8.6.14. (!) Given a graph G, let $R(G)$ be the matrix whose i, jth entry is $d_G(v_i, v_j)$. Prove that the squashed-cube dimension of a graph (Definition 8.4.12) is at least the maximum of the number of positive eigenvalues and the number of negative eigenvalues of $R(G)$. Conclude that the squashed cube dimension of K_n is $n - 1$. (Hint: Rewrite the quadratic form $x^T Rx$ as a sum of squares of linear functions, and apply Sylvester's Law of Inertia.)

8.6.15. (!) The **Laplacian matrix** Q of a graph G is $D - A$, where D is the diagonal matrix of degrees and A is the adjacency matrix. The **Laplacian spectrum** is the list of eigenvalues of Q.

a) Prove that the smallest eigenvalue of Q is 0.

b) Prove that if G is connected, then eigenvalue 0 has multiplicity 1.

c) Prove that if G is k-regular, then $k - \lambda$ is a Laplacian eigenvalue if and only if λ is an ordinary eigenvalue of G, with the same multiplicity.

8.6.16. Given a real symmetric matrix partitioned as $M = \binom{P \ \ Q}{Q^T \ R}$ with P, R square, a lemma in linear algebra yields $\lambda_{\max}(M) + \lambda_{\min}(M) \le \lambda_{\max}(P) + \lambda_{\max}(R)$.

a) Let A be a real symmetric matrix partitioned into t^2 submatrices $A_{i,j}$ such that the diagonal submatrices A_{ii} are square. Prove that

$$\lambda_{\max}(A) + (t-1)\lambda_{\min}(A) \le \sum_{i=1}^{m} \lambda_{\max}(A_{ii}).$$

b) Prove that $\chi(G) \ge 1 + \lambda_{\max}(G)/(-\lambda_{\min}(G))$ when G is nontrivial. (Wilf)

c) Use the Four Color Theorem to prove that $\lambda_1(G) + 3\lambda_n(G) \le 0$ for planar graphs.

8.6.17. (!) Use Theorem 8.6.28 to count the spanning trees in $K_{m,m}$. (Comment: See Exercise 2.2.11.)

8.6.18. (+) Given a matrix A, let $b_{i,j}$ equal $(-1)^{i+j}$ times the matrix obtained by deleting row i and column j of A. Let Adj A be the matrix whose entry in position i, j is $b_{j,i}$. The definition of the determinant by expansion along rows of A yields $A(\text{Adj } A) = (\det A)I$. Use this formula to prove that if the sum of the columns of A is the vector 0, then $b_{i,j}$ is independent of j. (Comment: With the next exercise, this completes the proof of the Matrix Tree Theorem (Theorem 2.2.12).)

8.6.19. (+) Let $C = AB$, where A and B are $n \times m$ and $m \times n$ matrices. Given $S \subseteq [m]$, let A_S be the $n \times n$ matrix whose columns are the columns of A indexed by S, and let B_S be the $n \times n$ matrix whose rows are the rows of B indexed by S. Prove the Binet–Cauchy Formula: $\det C = \sum_S \det A_S \det B_S$, where the summation extends over all n-element subsets of $[m]$. (Hint: Consider the matrix equation $\begin{pmatrix} I_m & 0 \\ A & I_n \end{pmatrix}\begin{pmatrix} -I_m & B \\ A & 0 \end{pmatrix} = \begin{pmatrix} -I_m & B \\ 0 & AB \end{pmatrix}$.)

8.6.20. A matrix is **totally unimodular** if every square submatrix has determinant in $\{0, 1, -1\}$. Prove that the incidence matrix of a simple graph is totally unimodular if and only if the graph is bipartite. (Reminder: The incidence matrix of a simple graph has two $+1$'s in each column).

8.6.21. (−) Let G be an (n, k, c)-magnifier with vertices $v_1, \dots, v_n$. Let H be the bipartite graph with partite sets $X = \{x_1, \dots, x_n\}$ and $Y = \{y_1, \dots, y_n\}$ such that $x_i y_j \in E(H)$ if and only if $i = j$ or $v_i v_j \in E(G)$. Prove that H is an $(n, k+1, c)$-expander.

8.6.22. *Existence of expanders of linear size.*

a) Let X be a random variable giving the size of the union of k s-subsets of $[n]$ chosen at random from $\binom{[n]}{s}$. Prove that $P(X \le l) \le \binom{n}{l}(l/n)^{ks}$.

b) (+) For $\alpha\beta < 1$, prove that there is a constant k such that, when n is sufficiently large, there exists a subgraph of $K_{n,n}$ with maximum degree at most k such that $|N(S)| \ge \beta\,|S|$ whenever $|S| \le \alpha n$. (Hint: Generate bipartite subgraphs of $K_{n,n}$ by taking the union of k random perfect matchings.)

c) Conclude the existence of k such that n, k, c-expanders exist for all sufficiently large n. An (n, α, β, d)**-expander** is a bipartite graph $G \subseteq K_{A,B}$ with $|A| = |B| = n$, $\Delta(G) \le d$, and $|N(S)| \ge \beta\,|S|$ whenever $|S| \le \alpha n$.

8.6.23. Let G be a triangle-free graph on n vertices in which every pair of nonadjacent vertices has exactly two common neighbors. Prove that G is regular and that $n = 1 +$

$\binom{k+1}{2}$, where k is the degree of the vertices in G. Prove that G is strongly regular. What constraints on k are implied by the integrality conditions? Construct examples for all $k \in \{1, 2, 5\}$. A realization for $k = 10$ is known using combinatorial designs.)

8.6.24. (+) Prove that the Petersen graph is strongly regular, and determine its spectrum (the spectrum is easy with properties of strongly regular graphs and not hard without them). Apply the spectrum to show that edges of the complete graph K_{10} cannot be partitioned into three disjoint copies of the Petersen graph. (Hint: Use the spectrum to prove that two copies of the Petersen matrix have a common eigenvector other than the constant vector.) (Schwenk [1983])

8.6.25. Let $F = G \square H$, where G and H are simple graphs. Prove that if every two nonadjacent vertices in F have exactly two common neighbors, then G and H are complete graphs.

8.6.26. The **subconstituents** of a graph are the induced subgraphs of the form $G[U]$, where $v \in V(G)$ and $U = N(v)$ or $U = \overline{N[v]}$. Vince [1989] defined G to be **superregular** if G has no vertices or if G is regular and every subconstituent of G is superregular. Let **S** be the class consisting of $\{aK_b\colon a, b \geq 0\}$ (disjoint unions of isomorphic cliques), $\{K_m \square K_m\colon m \geq 0\}$, C_5, and the complements of these graphs.

a) Prove that every graph in **S** is superregular and that every disconnected superregular graph is in **S**. (Comment: In fact, every superregular graph is in **S**, but the complete inductive proof of this requires several pages (Maddox [1996], West [1996])

b) Prove that every superregular graph is strongly regular.

8.6.27. (+) *Automorphisms and eigenvalues.*

a) Prove that σ is an automorphism of G if and only if the permutation matrix corresponding to σ commutes with the adjacency matrix of G; that is, $PA = AP$.

b) Let x be an eigenvector of G for an eigenvalue of multiplicity 1, and let P be the permutation matrix for an automorphism of G. Prove that $Px = \pm x$.

c) Conclude that when every eigenvalue of G has multiplicity 1, every automorphism of G is an involution, meaning that repeating it yields the identity. (Mowshowitz [1969], Petersdorf–Sachs [1969])

8.6.28. (+) Light bulbs $l_1, \ldots, l_n$ are controlled by switches $s_1, \ldots, s_n$. The ith switch changes the on/off status of the ith light and possibly others, but s_i changes the status of l_j if and only if s_j changes the status of l_i. Initially all the lights are off. Prove that it is possible to turn all the lights on. (Peled [1992]) (Hint: This uses vector spaces, not eigenvalues.)