Math 396: Applied Combinatorics and Graph Theory Lecture Notes

Created by: Dr. Amanda Harsy and Dr. Marie Meyer

June 21, 2019

.

Contents

1 Syllabus and Schedule **5**
1.1 Group Project Description: 14

2 Introduction to Graph Theory **19**
2.1 Why Study Graphs? 19
2.2 ICE 1: Graphical Models 27
2.3 Basic Graph Theory Terms 29
2.3.1 Paths, Walks, Trails, and more 33
2.4 ICE 2: Intro to Graphs 35
2.5 Families of Graphs 37
2.5.1 Path Graphs 37
2.5.2 Cycle Graphs 37
2.5.3 Complete Graphs 38
2.5.4 Bipartite Graphs 39
2.5.5 K-partite Graphs 40
2.5.6 Star Graphs 41
2.6 Creating New Graphs 41
2.6.1 Union Graph 41
2.6.2 Complement Graph 41
2.6.3 Join Graph 42
2.6.4 Cartesian Product of Graphs 42
2.6.5 Hypercubes Graphs 43
2.7 Digraphs 43
2.8 ICE 3: Classes of Graphs 45

3 Degrees **47**
3.1 ICE 4: Degrees 51
3.2 Regular Graph 53
3.3 Degree Sequences 53
3.4 Matrices 56

4 Isomorphic Graphs **59**
4.1 Useful Theorems to help with the Isomorphism Question 62
4.1.1 Degree Sequences 62
4.1.2 Subgraphs 64
4.2 Other Theorems 65
4.3 Ice 5: Isomorphisms 67

5 Trees **69**
5.1 Search Trees and Spanning Trees . . . 74
5.1.1 Minimal Spanning Tree Algorithms for Weighted Trees . . . 78
5.2 Ice 6- Trees . . . 80

6 Traversability **83**
6.1 Hamiltonian Graphs . . . 83
6.2 Activity: Constructing Disjoint Hamiltonian Cycles of K_{2n+1} . . . 85
6.3 ICE 7: Eulerian and Hamiltonian Graphs . . . 88

7 Digraphs **90**
7.1 Strong Digraphs . . . 91
7.2 Tournaments . . . 94
7.3 ICE 8: Digraphs and Tournaments . . . 97

8 Network Algorithms **99**
8.1 Shortest Path . . . 99
8.2 Network Flows . . . 101
8.3 ICE 9: Network Algorithms . . . 107

9 Combinatorics **109**
9.1 Basic Counting Principles . . . 109
9.1.1 Ice -Counting Principles . . . 113
9.1.2 Ice -Does Order Matter . . . 115
9.2 Permutations and Combinations . . . 117
9.2.1 Ice -Permutations and Combinations . . . 121
9.3 Pigeon Hole Principle . . . 123
9.4 ICE- More Counting and Pigeon Hole Principle . . . 125

10 Generating Functions **127**
10.1 Generating Function Models . . . 127
10.1.1 ICE -Generating Function Models . . . 131
10.2 Calculating Coefficients of Generating Functions . . . 133
10.2.1 Coefficients of Generating Functions . . . 137

1 Syllabus and Schedule

"Combinatorial reasoning underlies all analysis of computer systems. It plays a similar role in discrete operations research problems and in finite probability. Two of the most basic mathematical aspects of computer science concern the speed and logical structure of a computer program. Speed involves enumeration of the number of times each step in a program can be performed. Logical structure involves flow charts, a form of graphs. Analysis of the speed and logical structure of operations research algorithms to optimize efficient manufacturing or garbage collection entails similar combinatorial mathematics. Determining the probability that one of a certain subset of equally likely outcomes occurs requires counting the size of the subset. Such combinatorial probability is the basis of many nonparametric statistical tests. Thus, enumeration and graph theory are used pervasively throughout the mathematical sciences". -Alan Tucker, *Applied Combinatorics*

Thanks for taking Applied Combinatorics and Graph Theory with me! I have never taught or taken a course like this, so I am excited to go through this journey with you!

As this is my first time teaching this course, I have a few goals in mind when putting this workbook together for you. First I am trying to pick out a selection of topics which are interesting, mathematically rigorous, but also practical, intuitive, and algorithmic. As such we will have lots of definitions, theorems, lemmas throughout the course along with practical examples using these definitions and theorems. We will be doing some proofs, so it is important to review your Introduction to Proofs materials and/or read through Appendix 1.2 and 3 from our textbook.

During the last third of the course, we will go into combinatorics, the study of counting, which like graph theory has many applications in a variety of disciplines especially in computer science, mathematics, and the physical sciences. Some of the main combinatorial questions one may ask are, "Is there an existence of an arrangement" (is it possible to arrange these objects in the following way) and "what is the enumeration or classification of the arrangement."[1]. One we know such an arrangement exists, we can then explore it further and try to find the best or optimal arrangement. A typical solution of a combinatorial problem often includes the following steps:

1. Set up a mathematical model.
2. Study the model.
3. Do some computation for small cases in order get get some intuition and insight on the problem.
4. Use careful reasoning, previous results, and theorems to solve the problem.

[1] *Introductory Combinatorics* by Richard Brualdi

I am looking forward to teach you these exciting topics which are useful in many areas of applied mathematics. Specifically, two of my main research areas: Predictive Modeling of Sports Ranking Systems and Modeling DNA Self-Assembly use graphs, so I am very excited to go through this material with you! I have worked hard to create this course packet for you, but it is still a work in progress. Please be understanding of the typos I have not caught, and politely bring them to my attention so I can fix them for the next time I teach this course. I look forward to meeting you and guiding you through this course!

Cheers,
Dr. H

Acknowledgments: No math teacher is who she is without a little help. I would like to thank my own undergraduate professors from Taylor University: Dr. Ken Constantine, Dr. Matt Delong, and Dr. Jeremy Case for their wonderful example and ideas for structuring excellent learning environments. I also want to thank Dr. Marie Meyer who will teach this course for 6 weeks and also has given significant insight for this course. *And finally, I would like to thank you and all the other student for making this job worthwhile and for all the suggestions and encouragement you have given me over the years to improve.*

MATH-396-03 APPLIED COMBINATORICS AND GRAPH THEORY
Spring 2019 SYLLABUS

Instructor:	Amanda Harsy	Office: AS-124-A
Email:	harsyram@lewisu.edu	Office phone: 815-836-5688
Class Time:	TR 12:30-1:45	Class Room: AS-157-A
Office Hours:	M 12-2, T 2-3, W 1:30-2:00, R 2:30-3:30,	or by appt

Office hours are subject to change, once I find out meeting times for my committees, but I will let you know if they change. Remember you can ALWAYS make an appointment.

Special Note: I am due to have a baby at the end of February. During this time, we will have class as normal and Dr. Marie Meyer will be taking over for me. Her office is AS-120-A and her email is mmeyer2@lewisu.edu . During these 4-6 weeks, I will not be holding normal office hours, but will be available through email. Dr. Meyer will be available during this time. Her office hours are M 12-1pm and 2-3pm, T 9-10am, W 2-3pm, F 12-1pm, or by appointment.

Course description

This course covers a selection of standard topics from combinatorics and graph theory with an emphasis on the applications of these topics.

About the Course

Text: *A First Course in Graph Theory* By Gary Chartrand, Ping Zhang. (ISBN: 0486483681)
Suggested Text: *Applied Combinatorics* By Alan Tucker (ISBN: 0470458380)
Prerequisites: none.

Course Objectives:

To help each student

- understand and use concepts of Graph Theory and Combinatorics,
- apply graph theory and combinatorial concepts to various applications,
- become better at solving mathematical problems,
- experience doing mathematics cooperatively,
- effectively communicate mathematics, and
- enjoy his/her experience with mathematics.

Resources

Blackboard: Check the Blackboard website regularly (at least twice a week) during the semester for important information, announcements, and resources. It is also where you will find the course discussion board. Also, check your Lewis email account every day. I will use email as my primary method of communication outside of office hours.

Help: Don't wait to get help. Visit me during my office hours, use the discussion forum in Blackboard, go to the Math Study Tables, find a study partner, get a tutor!

Dr. Harsy's web page: For information on undergraduate research opportunities, about the Lewis Math Major, or about the process to get a Dr. Harsy letter of recommendation, please visit my website: `http://www.cs.lewisu.edu/~harsyram`.

Course Requirements And Grading Policy

Homework: Almost every week, I will collect a homework assignment. I will post these homework assignments on Blackboard. You may work with others on the homework, but it must be your own work. If I catch you copying homework, you will get a 0. Please see the academic honesty section below. Sometimes the homework include a “warm-up” assignment to either supplement what was learned in class or to introduce what will be covered in the following class.

Graph Theory and Applications Final Project: You, as part of a 3-4 person group, will select a topic pertaining to the course, read additional material pertaining to the topic, write a short paper summarizing the main ideas and then give a 20 minute presentation to the class on your topic. This presentation will during our final exam block which is on 5/13 at 10:30am-12:30pm. Rubrics for this project are posted in Blackboard.

Mastery-Based Testing: This course will use a testing method called, “Mastery-Based Testing.” There will be three (3) paper-and-pencil, in-class Mastery Exams given periodically throughout the semester. In mastery-based testing, students receive credit only when they display “”mastery,” but they receive multiple attempts to do so. The primary source of extra attempts comes from the fact that test questions appear on every subsequent test. Since this course has never been offered before, the timing and number of concepts for MBT may change, but for now we expect Test 1 will have 6 questions. Test 2 will have 12 questions -a remixed version of the questions from Test 1 and new questions on the new concepts. Test 3 will have 16 questions- a remixed version of the questions from Test 2 and four new questions. We will also have three testing weeks. During these weeks, students can use doodle to sign up to retest concepts during (extended) office hours and math study tables. Students are allowed to test any concept, but cannot retest that concept the rest of the week. So for example, a student can test concepts 2, 3 and 5 on Monday and concept 6 on Tuesday, but would not be able to test concept 5 again. These testing weeks are tentatively set for 3/18-3/22, 4/11-17, and during Finals week. Students will be able to retest the new concepts from the last exam twice during the final testing week.

Grading of Mastery-Based Tests: The objectives of this course can be broken down into 18 main concepts/problems. For each sort of problem on the exam, I identify three levels of performance: master level, journeyman level, and apprentice level. I will record how well the student does on each problem (an M for master level, a J for journeyman level, a 0 for apprentice level) on each exam. After the Final testing week, I will make a record of the highest level of performance the student has made on each sort of problem or project and use this record to determine the student’s total exam grade. Each of the first 8 concept/questions students master will count 9% points towards their exam grade. After that, each concept/question will be worth 3.5% towards your exam grade. So for example, if you master 10 of the 15 concepts your grade will be a 79%.

This particular way of arriving at the course grade is unusual. It has some advantages. Each of you will get several chances to display mastery of almost all the problems. Once you have displayed mastery of a problem there is no need to do problems like it on later exams. So it can certainly happen that if you do well on the midterms you might only have to do one or two problems on the Final. (A few students may not even have to take the final.) On the other hand, because earlier weak performances are not averaged in, students who come into the Final on shaky ground can still manage to get a respectable grade for the course. This method of grading also stresses working out the problems in a completely correct way, since accumulating a lot of journeyman level performances only results in a journeyman level performance. So it pays to do one problem carefully and correctly as opposed to trying get four problems partially correctly. Finally, this method of grading allows you to see easily which parts of the course you are doing well with, and which parts deserve more attention. The primary disadvantage of this grading scheme is that it is complicated. At any time, if you are uncertain about how you are doing in the class I would be more than glad to clarify the matter. The tentative test dates are below.

Exam #	Date
1	Thursday, 2/28
2	Thursday, 4/4
3	Tuesday, 4/30

Grade scale

90%-92%	A-	93%-96%	A	> 97%	A+
80%-82%	B-	83%-86%	B	87%-89%	B+
70%-72%	C-	73%-76%	C	77%-79%	C+
60%-62%	D-	63%-66%	D	67%-69%	D+
		< 60%	F		

Category	Percentage
Mastery Exams	65
Homework	20
Project	15

Course Policies and Procedures

Calculator/Technology/Phone Policy: No phones or other forms of technology can be used on in-class exams unless specifically noted. A basic scientific or graphing calculator is ok. Please do not have your phones out unless it is a class activity. If I see a phone out during a quiz or exam, you will receive an F on that assessment.

Make-Ups: There will be no make-ups for any assignments. If you are late or miss class, your assignment will not be accepted and there will be no make-up offered, except in extenuating and unpredictable circumstances. If you will miss class for a justifiable and unavoidable reason, you can contact me **before** you miss class and it is possible you can have a make-up. If you do not contact me and explain your absence, you will not be allowed a make-up.

Class Attendance: Students are expected to attend all classes as part of the normal learning process. In addition, students must be especially consistent in attendance, both on-ground and online, during the first two weeks of the semester to confirm registration and to be listed on the official course roster. Students who fail to follow this procedure and who have not received prior approval from the instructor for absences will be withdrawn from the courses in question by certification of the instructor on the official class lists. Instructors may publish specific, additional reasonable standards of attendance for their classes in the course syllabus. Students may receive failing grades if they do not observe attendance requirements. The Illinois Student Assistance Commission also requires attendance as a demonstration of academic progress toward a degree as one criterion for retaining financial aid awards. (2015-2016 Undergraduate Catalog, p. 34).

Attendance is critical for success in this course and is required of all students without exception. A student absent from class is responsible for all material covered that day. I usually post solutions to ICE sheets and post notes. I will stop doing this if there is a trend of absences. In other words, I will not post lecture notes unless there is consistent attendance. Students are expected to attend all classes as part of the normal learning process.

Course Relationship to Mission: Lewis University, guided by its Catholic and Lasallian heritage, provides to a diverse student population programs for a liberal and professional education grounded in the interaction of knowledge and fidelity in the search for truth.Lewis promotes the development of the complete person through the pursuit of wisdom and justice. Fundamental to its Mission is a spirit of association, which fosters community in all teaching, learning and service. This course embraces the Mission of the University by fostering an environment in which each student is respected as an individual within a community of learners. In the spirit of the vision of Lewis University, the goals and objectives of this course seek to prepare students to be successful, life-long learners who are intellectually engaged, ethically grounded, socially responsible, and globally aware.

Academic Integrity: Scholastic integrity lies at the heart of Lewis University. Plagiarism, collusion and other forms of cheating or scholastic dishonesty are incompatible with the principles of the University. Students engaging in such activities are subject to loss of credit and expulsion from the University. Cases involving academic dishonesty are initially considered and determined at the instructor level. If the student is not satisfied with the instructors explanation, the student may appeal at the department/program level. Appeal of the department /program decision must be made to the Dean of the college/school. The Dean reviews the appeal and makes the final decision in all cases except those in which suspension or expulsion is recommended, and in these cases the Provost makes the final decision.

University Copyright and Intellectual Properties Guidelines can be found at https://www.lewisu.edu/academics/library/pdf/Copyright-Intellectual-Property-Guidelines.pdf

University Student Complaint Policy: The University Student Complaint Policy can be found at lewisu.edu/studentcomplaints.

University Grade Appeal Policy: The University Grade Appeal Policy can be found at lewisuedu/studentcomplaints.

Classroom Decorum: In order to maintain an environment conducive to learning and student development, it is expected that classroom discourse is respectful and non-disruptive. The primary responsibility for managing the classroom environment rests with the faculty. Students who engage in any prohibited or unlawful acts that result in disruption of a class may be directed by the faculty member to leave class for the remainder of the class period. Students considered to be a disruption or who present a threat of potential harm to self or others may be referred for action to the Dean of Student Services. 2015-2016 Student Handbook, p. 14, Lewis University website https://www.lewisu.edu/sdl/pdf/studenthandbook.pdf.

Sanctified Zone: This learning space is an extension of Lewis Universitys Sanctified Zone, a place where people are committed to working to end racism, bias and prejudice by valuing diversity in a safe and nurturing environment. This active promotion of diversity and the opposition to all forms of prejudice and bias are a powerful and healing expression of our desire to be Signum Fidei, Signs of Faith, in accordance with the Lewis Mission Statement. To learn more about the Sanctified Zone, please visit: http://www.lewisu.edu/sanctifiedzone.

Center for Health & Counseling Services
To support student success, all Lewis students are eligible for free health and mental health services on the Romeoville campus. This includes commuters and those living on campus, part-time and full-time students, graduate and undergraduate students, and those taking Lewis classes at other locations. For more information, visit the Center for Health & Counseling website at www.lewisu.edu/studentservices/health or call (815)836-5455.

Requests for Reasonable Accommodations Lewis University is committed to providing equal access and opportunity for participation in all programs, services and activities. If you are a student with a disability who would like to request a reasonable accommodation, please speak with the Learning Access Coordinator at the Center for Academic Success and Enrichment (CASE). Please make an appointment by calling 815-836-5593 or emailing learningaccess@lewisu.edu. Since accommodations require early planning and are not provided retroactively, it is recommended that you make your request prior to or during the first week of class. It is not necessary to disclose the nature of your disability to your instructor.

For more information about academic support services, visit the website at: www.lewisu.edu/CASE.

Lewis University has adopted Blackboard Ally providing alternative formats for files uploaded by instructors. Students can click the down arrow next to any file, and select Alternative Formats.

Applied Combinatorics and Graph Theory Student Learning Outcomes:

Upon successful completion of this course, students should be able to

(1) Understand and identify the fundamental definitions and concepts of graph theory
(2) Use Graphs to model and solve mathematical problems
(3) Use Combinatorics to solve mathematical problems

Mathematics Program Learning Outcomes:

A Lewis University Mathematics Graduate will be able to:

(1) Model the essential components of a problem mathematically so that it can be solved through mathematical means.
(2) Solve problems through the appropriate technology.
(3) Write clear and concise formal proofs of mathematical theorems, and evaluate the validity of proofs written by others.
(4) Solve mathematics problems as a member of a group and in front of a group.
(5) Describe and leverage the interdependency of different areas of mathematics, the connections between mathematics and other disciplines, and the historical context for the development of mathematical ideas.

Of these, this course addresses program learning outcomes 1, 4, and 5, all at a reinforced level.

Lewis University Baccalaureate Characteristics:

A Lewis University undergraduate education is marked by the following characteristics:

(1) **Essential Skills:** The baccalaureate graduate of Lewis University will read, write, speak, calculate, and use technology at a demonstrated level of proficiency.
(2) **Major Approaches to Knowledge:** The baccalaureate graduate of Lewis University will understand the major approaches to knowledge.
(3) **Faith, Religion and Spirituality:** The baccalaureate graduate of Lewis University will understand the place of faith, religion, and spirituality in the search for truth and meaning.
(4) **Moral and Ethical Decision-Making:** The baccalaureate graduate of Lewis University will understand and prepare for moral and ethical decision-making.
(5) **Responsible Citizenship:** The baccalaureate graduate of Lewis University will become an informed, involved, and responsible citizen of a diverse yet interconnected national and global community through a grounding in economic, political, social, and historical influences that are inherent in shaping, developing, and advancing nations and the world.
(6) **Critical Thinking:** The baccalaureate graduate of Lewis University will think critically and creatively.
(7) **Lifelong Learning:** The baccalaureate graduate of Lewis University will possess the knowledge, skills, and dispositions to enter or advance a career, or to begin graduate study.

This course addresses Baccalaureate Characteristics 1, 6, and 7 all at a reinforced level. It provides students essential skills in mathematics, abstraction, and algorithmic thinking. It reinforces quantitative and empirical approaches to knowledge. In asking students to reduce problems to their important components and charting a path for solving them, it fosters critical thinking. Finally, in casting mathematics as a dynamic and evolving influencer of our collective future, it encourages curiosity and lifelong learning.

Student Learning Outcome	**Math Program**	**Baccalaureate Characteristic**	**Demonstrated by:**
Understand and identify properties of graphs	1	6, 7	homework, exams, class work,
Use graphs to model and solve mathematical problems	1,4,5	1,6,7	homework, exams, class work, final group project
Use combinatorics to solve mathematical problems	1,4,5	1,6,7	homework, exams, class work, final group project

This, That, Tips, and Suggestions...

Here are some suggestions that may help you in the class.

Situation: I am a student that learn best from writing everything down and I wish the lecture notes weren't so typed out.
Suggestion: Writing is the best way for me to remember things too! I want to encourage you to write if it helps you. Using lecture notes keep me somewhat organized and helps us get through all of the material (of which there is a ton)! Also, my handwriting is horrible so having some words typed is good for you. My suggestion is to rewrite your lecture notes into a notebook. Rewriting notes is a wonderful study tool and I encourage you to do this if you don't think the partially filled out notes allow you to learn best.

Situation: I can't see the document camera.
Suggestion: I am used to higher quality document cameras and am waiting to get a better one. I have horrible board handwriting so I don't like to write on the board for the majority of class. I do write on the board sometimes. Also I like the fact that the document camera allows me to face the class, so hopefully I can keep tabs on how the class is doing. My suggestion for you is to move to a different seat if you can't see the board. And also please let me know if you are having issues and I can try to refocus/adjust the document camera.

Practice, Practice, Practice: The most common suggestion from past students is to do ALL the homework. Learning mathematics is like any activity and requires that you practice in order to become proficient.

Note: This syllabus is subject to change. If there is a change, it will be discussed in class or through an email and will be updated in Blackboard.

Class Notation

- $\mathbb{R}$ represents all real numbers (so no imaginary numbers)
- $\mathbb{Z}$ represents integer numbers
- $\mathbb{Q}$ represents rational numbers [numbers that can be written as a fraction $\frac{p}{q}$ where p,q are integers]
- { } represents a set. For example $\{x$; such that x is an integer $\}$=Z
- $\in$ means "contained in". For example, $5 \in \Re$
- $\notin$ means "not contained in". For example $\pi \notin Z$.
- s.t. means "such that"
- "|" or ":" mean "such that" in a set setting. For example $\{x|x \in Z\}= \{x : x \in Z\} =$ $\{x$; such that x is an integer $\} = Z$
- $\therefore$ means "therefore".
- $\exists$ means "there exists"
- $\forall$ means "for all"
- $\Rightarrow$ means "this implies"
- $\iff$ or "iff" means "if and only if"
- Pf means "Proof"

Applied Combinatorics and Graph Theory Tentative Schedule Spring 2019

This schedule is subject to change, but you will be notified of any changes ahead of time.

Agenda	Special Notes
Week 1: 1/14 Intro to Graphs and Graph Models	
Week 2: 1/21 Graph Definitions and Properties	HW 1 due
Week 3: 1/28 Degrees	HW 2 due
Week 4: 2/4 Isomorphisms	HW 3 due
Week 5: 2/11 Trees	HW 4 due
Week 6: 2/18 Traversability **Mastery Exam 1**	**Mastery Exam 1 on Thursday**
Week 7: 2/25 Flex	HW 5 due
Week 8: 3/4 Digraphs ***Retesting Week 1***	HW 6 due ***Retesting Week 1*** **Topic for Group Presentation Due Thursday**
Spring Break: 3/11 **No class this week!**	
Week 9: 3/18 Network Algorithms	HW 7 due
Week 10: 3/25 Additional Graph Theory Topics	HW 8 due
Week 11: 4/1 Additional Graph Theory Topics **Mastery Exam 2**	**Mastery Exam 2 on Thursday**
Week 12: 4/8 Introduction to Combinatorics (multiplicative and additive counting principles, permutations and combinations, and the pigeonhole principle)	***Retesting week 2 Thurs-next Wed*** ***Outline of paper & talk due Thursday (in class)***
Week 13: 4/15 Generating Functions ***Retesting week 2***	No class on Thursday due to Easter Break ***Retesting week 2*** **HW 9 due Tuesday**
Week 14: 4/22 Recurrence relations	***Draft of Paper due Tuesday*** HW 10 due
Week 15: 4/29 Flex **Mastery Exam 3**	HW 11 due **Mastery Exam 3 on Tuesday**
Week 16: 5/6 Group Presentations during our final exam block: Thursday, 10:30 a.m.- 12:30 Final Testing Week	***Final Testing week*** ***Final Paper due Thursday*** ***Presentations on Thursday***

1.1 Group Project Description:

See next page!

Graph Theory and Combinatorics Applications Final Project

Overview: You will select a topic pertaining to the course, read additional material pertaining to the topic, write a short paper summarizing the main ideas and then give a 20 minute presentation to the class on your topic.

Goals: So far this semester, we've seen a number of both theoretical and applied results from graph theory. This is your chance to either explore one of those topics in more depth or to explore a more advanced topic. You will also gain experience reading journal articles from mathematics or another discipline and presenting technical information.

Parameters:

- The choice of topic must be made in consultation with Dr. Harsy or Dr. Meyer, some ideas are listed below. The topics below are very broad; you'll want to quickly narrow to a single idea.

- Your paper and presentation must make significant use of at least one journal article, must include at least one application to a field outside graph theory, and must include at least one significant graph theoretic result (though not necessarily its proof). You do not necessarily need to understand (or even) read the entire journal article.

- You may consult online sources such as Wikipedia, though the substance of your paper and presentation must come from other sources.

- You must carefully cite every source you use for your paper or presentation, using an inline reference to a bibliography/works cited page (you can use MLA, APA, or the LaTeX bibtex format). I can show you how to easily do this using Google Scholar and LaTeX. You may also include a "Works Consulted" list for sources which did not have a direct impact on your final product but which were helpful in getting you there. A source you glance at only briefly need not be placed in the Works Consulted list.

  Citing our sources is a way of giving thanks to those whose work has influenced you and enables later scholars to retrace your intellectual journey. Additionally, it enables you and others to distinguish your contributions (perhaps only expositional!) from those of others.

- Your paper should be 5 – 10 pages long, though significant use of figures may make it longer. The target audience can be either your classmates or someone working in another field who may benefit from graph theoretic concepts. The grading rubric for the paper is attached.

- Your presentation should be 20 minutes long and is aimed at your classmates. It is likely a condensed version of part of your paper. You may like to use slides to enable rapid dissemination of figures or background ideas. The grading rubric for the presentation is attached.

Deadlines:

<u>Thursday, March. 7</u>: Final decision on project topic. Have 2 back-ups in case of multiple groups wanting the same topic.

<u>Thursday, April 11:</u> Outline of both paper and talk due in class.

<u>Tuesday, April 23.</u> Draft of paper due.

<u>Thursday, May 9:</u> In class presentations (to be scheduled later) 10:30am-12:30pm

<u>Friday, May. 10:</u> Final version of paper due.

Presentation Grading Rubric:

(30%) Significant graph theoretic content beyond what was done in class.
(15%) Effective contextualization of the mathematics. Does the speaker motivate the ideas introduced? Does the speaker make connections with other subjects?
(25%) Clarity of presentation. Is the presentation understandable by the class? Is all relevant notation and terminology defined? Is effective use made of examples?
(20%) Preparation. Are visual aids well-designed? Is effective use made of the board? Has the talk been practiced?
(5%) Group Work -were you a good group member?
(5%) Student Audience Evaluation -was the talk interesting and easy to follow?

Paper Grading Rubric:

(30%) Organization. Is the organization logical? Is there both flow and structure?
(15%) Motivation. Is clear why the reader should be interested in the topic of the paper?
(35%) Depth. Does the paper make significant use of at least one journal article? Does the paper explain connections to a subject outside of graph theory? Does the paper contain significant mathematics.
(15%) Mechanics. Are grammar, punctuation, spelling (mostly) correct? Are citations correct? Are quotation marks used appropriately?
(5%) Group Work -were you a good group member?

Ideas for paper topics

You may suggest your own topic! But here are some ideas.

Note: You may not base your project on any project/paper/assignment/material from another course, though you are welcome to use the project to delve deeper into connections with another course.

- Independent Sets and Graph Homomorphisms.
  As an undergraduate Yufei Zhao solved a prominent open conjecture in graph theory concerning the number of independent sets in a d-regular graph. In a recent *Monthly* article, Zhao explains his solution and gives lots of references for understanding the significance of the problem.
- Algorithms. Code some of the more difficult algorithms from the course and investigate ways they've been developed or optimized by others.
- Random Graphs. A random graph is created by choosing a vertex set and then assigning edges between vertices according to some probability distribution. Investigate how the randomness interacts with various graph theoretic properties or explore how random graphs are used in mathematical modelling.
- Colorful Variations. Explore variations of the 4-color problem.
- Game Theory and the Chromatic Number. Coloring the vertices of a graph can be viewed in game theoretic terms and some of the theorems we've learned about can be reproven from that perspective.
- Graphs and Knot Theory. In knot theory, the Kauffman bracket is an exceptionally simple idea with profound consequences. It is calculated using recursion relationships similar to those in graph theory. Similarly, other numbers associated to graphs can be turned into knot invariants.
- Chromatic Number and Scheduling In this project you'll investigate scheduling algorithms related to the chromatic number of a graph.
- Percolation. Percolation is an idea from statistical physics in which quantities at one vertex are redistributed to other vertices. There are close connections to physics, chemistry, and probability theory.
- Small-World Networks. Many social networks are modelled using "small-world networks". You'll investigate what these networks are and what their properties are. You could perhaps begin with an article on the network of interactions between the characters in *Game of Thrones*.
- Random Walks on Graphs A random walk in a graph is a created by moving from one vertex to an adjacent vertex with some probability. One basic question is "What's the probability of returning to the original vertex?"
- Graphs and Games: Investigate how graphs are used to study board games (typically as Markov Chains)
- Constantine's Problem: A classic problem about how to distribute legions to maintain effective control over the empire. Fun and accessible.
- Modeling DNA Self-Assembly. Graphs can be used to help design nanostructures. Joe Ellis-Monaghan has some papers which use tile-based assembly and graph theory results to help model this. Note if you are one of Dr. Harsy's researchers, you cannot choose this topic.
- Traveling Salesman Problem: Given a collection of cities and the cost of travel between each pair of them, find the cheapest way of visiting all of the cities

and returning to your starting point. Represent the situation as an undirected weighted graph. This is one of the most intensively studied problems in graph theory and has yet to be solved; however, there is progress to report as well as solutions for specific cases.

- Spanning trees or Matrix Representation of a Graph: Spanning Trees: Compare and contrast algorithms for finding spanning trees of a graph and present/prove a way to count the spanning trees using the Laplacian matrix of a graph. Kirchoff's Matrix-Tree Theorem will be useful) Matrix Representation of a Graph: We use powers of the adjacency matrix of a graph G to compute the number of walks between pairs of vertices in G. The Laplacian matrix is another matrix uniquely associated to G that can be used to compute the number of spanning trees of G. Present a proof of Kirchoff's Matrix-Tree Theorem and explore the connection with linear algebra.
- Dijkstra's Algorithm: Using this algorithm, one can find the shortest path between two vertices. Explore applications to different transportation problems.
- Greedy Algorithms: A greedy algorithm involves making the best optimal choice at each local stage with the goal of finding a globally optimal solution. Choose one or two greedy algorithms to study and look into problems which do have solutions under the given greedy algorithms.
- Have your own idea? Share your idea with Dr. Harsy or Dr. Meyer!

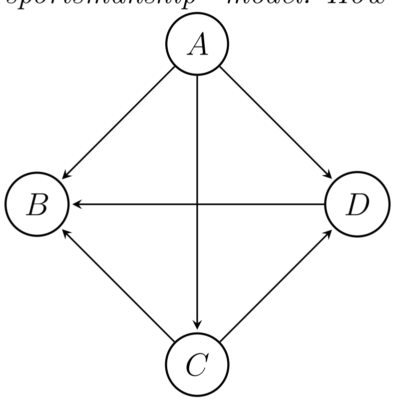
We may add other potential project ideas and will update you via Bb if we do!

2 Introduction to Graph Theory

2.1 Why Study Graphs?

Many problems can be modeled and solved with the help of graphs, directed graphs, and pseudo-graphs. We can model networks (social, website, circuit, etc.), job assignments and scheduling, chemical storage, traffic flow, spell checkers, athletic events, nanostructure design, surveillance, and more. In this section, we give a few examples of how we can use graphs to help with a variety of problems. These problems and examples will continue to come up throughout the course as we learn new graph or combinatorial ideas.

Example 2.1. *You can use graphs to model outcomes of sporting events like this "Poor sportsmanship" model. How would you rank these teams?*



Example 2.2. *Dr. Harsy uses graphs in her research modeling DNA self-assembly:*

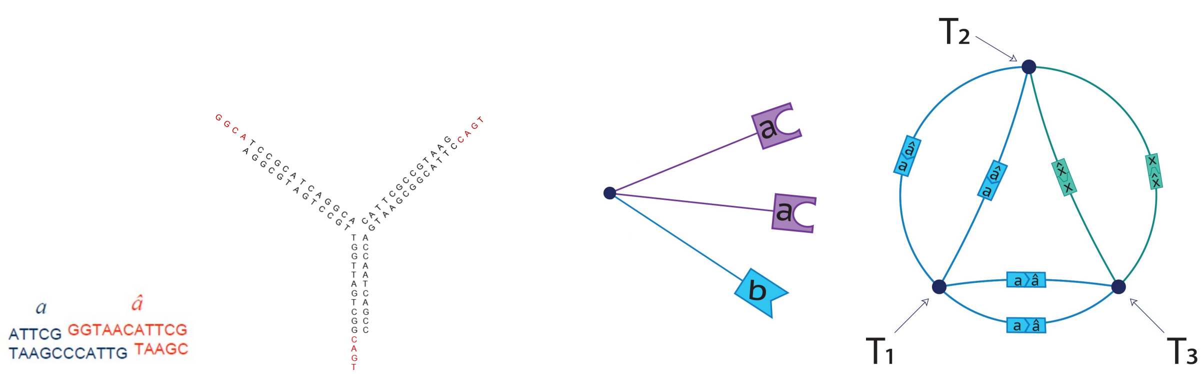


Figure 1: Representing the complementary cohesive end types (left), *3*-armed branched junction molecule (center left) with example tile representation (center right) Complete Graph(right)

Example 2.3. [2] *The Fantastic Math Kitties have started their own publishing company, The Cat's Meow. They have hired 10 editors in the various areas of Naps, Strings, Birdwatching, Milk, etc. These editors have been divided into 7 committees to meet each Friday at 3 potential time periods to discus specific topics of interest to the company, advertising, securing reviewers, finances, etc. Some pairs of committees cannot meet during the same period because more than one editor may be on both committees. The editors are split in the following ways for the committees:* $c_1 = \{1, 2, 3\}, c_2 = \{1, 3, 4, 5\}, c_3 = \{2, 5, 6, 7\}, c_4 = \{4, 5, 8, 9\}, c_5 = \{2, 6, 7\}, c_6 = \{8, 9, 10\}, c_7 = \{1, 3, 9, 10\}$. *Use a picture or a graph to help organize and model this problem.*

[2] Example adopted from *A first course in Graph Theory*

Example 2.4. [3] *The figure below shows the traffic lanes at the intersection of two busy streets. When a vehicle approaches this intersection,it could be in one of 9 lanes: L1, L2, ..., L9. We want to program the traffic light so that they tell the drivers when to proceed through the intersection in a way that will not result in a wreck. For example cars in L1 and L7 should not drive through the intersection at the same time, but L1 and L5 could! Represent this as a graph, G, with* $V(G) = \{L1, L2, ..., L9\}$ *and having the vertices joined by an edge if the vehicles in these two lanes* ***cannot*** *safely enter the intersection at the same time.*

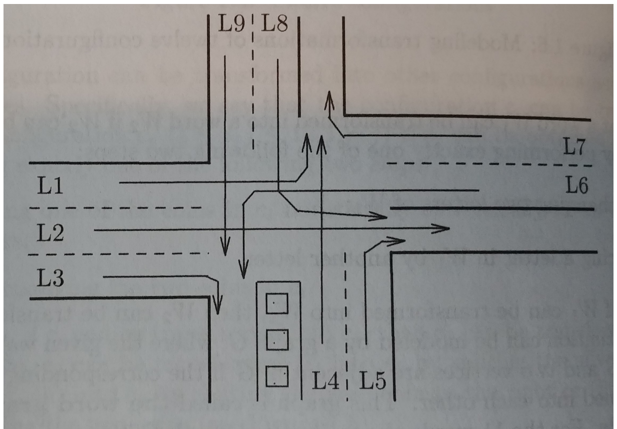


Figure 1.9: Traffic lanes at street intersections

[3]Example from *A first course in Graph Theory*

Example 2.5. [4] *A spelling checker looks at each word X (represented in a computer as a binary number) in a document and tries to match X with some word in its dictionary, which typically contains close to 100,000 words. To understand how this checking works, we consider the simplified problem of matching an unknown letter X with one of the 26 letters in the English alphabet. In the spirit of the strategy humans use to home in on the page in a dictionary where a given word appears, the computer search procedure would first compare the unknown letter X with M, to determine whether* $X \leq M$ *or* $X > M$*. The answer to this comparison locates X in the first 13 letters of the alphabet or the second 13 letters, thus cutting the number of possible letters for X in half. This strategy of cutting the possible matches in half can be continued with as many comparisons as needed to home in on Xs letter. Continuing in this manner we can determine what letter X is. Model this process using a graph.*

For reference: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Note: For our original spelling-checker problem, a word processor would use a similar, but larger, tree of comparisons. With just 12 rounds of comparisons, it could reduce the number of possible matches for an unknown word X from 100,000 down to 25, about the number of words in a column

[4] Example from *Applied Combinatorics*

of a page in a dictionary. (Once a list was reduced to about 25 possibilities, a computer search for X would usually run linearly down that list, just as a human would.)

Example 2.6. *Suppose Figure 2.1 below represents a section of a citys street map. We want to position police officers at corners (vertices) so that they can keep every block (edge) under surveillance -that is, every edge should have police officers at (at least) one of its end vertices. What is the smallest number of police officers that can do this job?*[5]

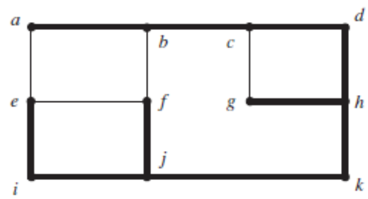


Figure 2: Figure from from *Applied Combinatorics*

[5]Example from *Applied Combinatorics*

Example 2.7. *Suppose psychological studies of a group of people determine which members of the group can influence the thinking of others in the group. We can make a graph with a vertex for each person and a directed edge* (a, b) *whenever person* a *influences* b*. The figure below models such a situation. What is a minimal subset of people who can spread an idea through to the whole group, either directly or by influencing someone who will influence someone else, and so forth. In graph-theoretic terms, we want a minimal subset of vertices with directed paths to all other vertices.*[6]

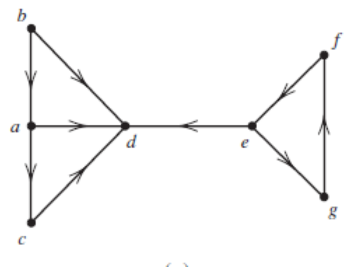


Figure 3: Figure from from *Applied Combinatorics*

[6]Example from *Applied Combinatorics*

2.2 ICE 1: Graphical Models

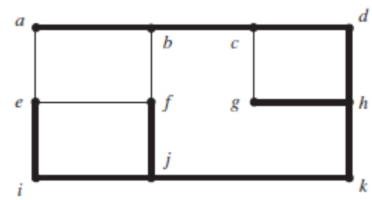


Figure 4: Figure from *Applied Combinatorics*

1. Suppose the graph above represents a network of telephone lines. We are interested in the networks vulnerability to accidental disruption. We want to identify sets of those lines and switching centers that must stay in service to avoid disconnecting the network.
   a) Is there a telephone line (edge) whose removal will disconnect the telephone network (graph)?

   b) Is there a vertex whose removal disconnects the graph?

   c) Is there any pair of edges whose removal disconnects the graph?

   d) Find a minimal set of edges needed to link together the 11 vertices in Figure 2.2.

2. Answer the question from Ex 2.6.

3. Answer the question from Ex 2.7.

.

2.3 Basic Graph Theory Terms

Definition 2.1. *Considered in an abstract setting, a **graph** $G = (V, E)$ is defined in terms of a set of vertices, V, and a set of edges, E, joining different pairs of vertices. We denote the number of vertices and edges of G respectively by $|V(G)|$ and $|E(G)|$.*
*$V(G)$ is the **vertex set of G** and $E(G)$ is the **edge set of G**.*

Although we could have infinite graphs with infinite vertices, most of this class will focus only on finite graphs. Sometimes vertices are called *points* or *nodes* and edges are called *lines*.

Example 2.8. *What is $V(G), |V(G)|, E(G), |E(G)|$ for the graph below?*

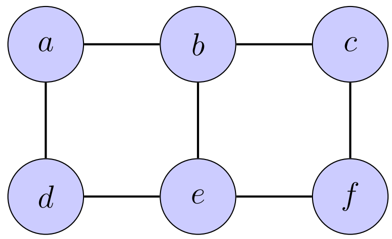


Figure 5: Example of a graph

Definition 2.2. *If two vertices are connected by an edge, we say these vertices are __________.*

So in the example above, a and b are adjacent, but a and f are not.

And if $e = uv$ is an edge of G, vertices u and v are adjacent. In fact we say u and v are **neighbors** and are **joined** by e.

Definition 2.3. *If two edges share the same vertex, they are called __________ edges.*

Definition 2.4. *The __________ of a given vertex is the number of edges incident to that vertex.*

So for the graph in Figure 5, vertex a has degree __________ since edges (a, b) and (a, d) are incident edges.

Definition 2.5. *The **order of a graph G** is given by the number of vertices.*
*The **size of a graph G** is given by the number of edges of G.*

Note: Some textbooks use size for the order of a graph (as does Dr. H -oops).

Example: The order of the graph in Figure 5 is __________. The size is __________.

Example 2.9. *The **Trivial Graph** is a graph with 1 vertex.*

So a **non-trivial** graph must have at least __________ vertices.

Example 2.10. *Draw a graph with 1 vertex and one edge.*

This is called a __________. Some textbooks allow loops in graphs, but your textbook only allows them in "pseudo-graphs." Loops do come up in applications so it is important to know what they are.

We also have a variation of graphs that our textbook calls, "Multigraphs."

Definition 2.6. *A **multigraph,** M, consists of a finite, non-empty vertex set and edge set in which every two vertices of M are joined by a finite number of edges (possibly zero).*

This means we can have something called, **parallel edges** in multigraphs which are multiple edges between the same vertices.

Pseudo-graphs can have parallel edges and loops (edges joining a vertex to itself).

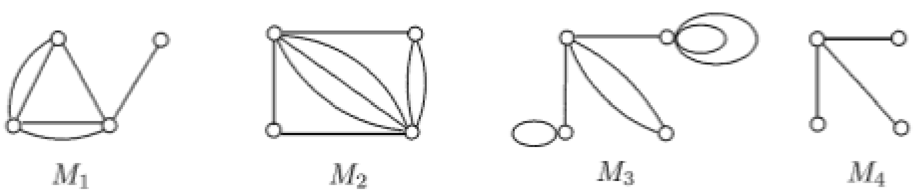


Figure 6: Example of a graph

Example 2.11. *Which of diagrams in Figure 6 are graphs?*

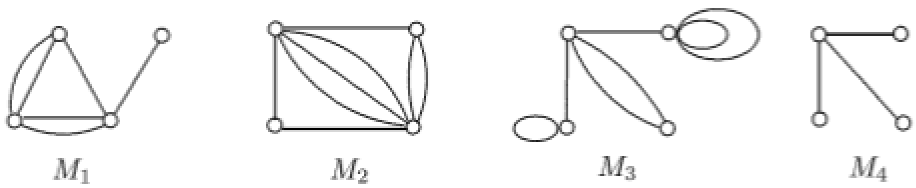
(a) M1 & M4

(b) M1, M2, & M4

(c) M4

(d) all of the diagrams

(e) none of the diagrams



Example 2.12. *Which of diagrams in Figure 6 are multigraphs?*

(a) M1 & M4

(b) M1, M2, & M4

(c) M4

(d) all of the diagrams

(e) none of the diagrams

Example 2.13. *Which of diagrams in Figure 6 are pseudographs?*

(a) M1 & M4

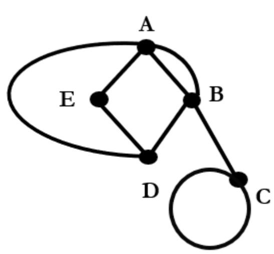
(b) M1, M2, & M4

(c) M4

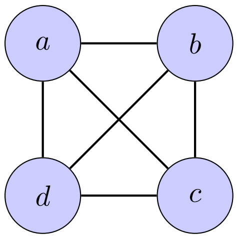
(d) all of the diagrams

(e) none of the diagrams

Example 2.14. *What are the degrees of each of the vertices of the graph below? Are they odd or even?*



Example 2.15. *The graph below has...*



(a) 6 edges, 4 vertices (exactly 2 of which are odd)

(b) 4 edges, 6 vertices (all of which are odd)

(c) 6 edges, 4 vertices (all of which are odd)

(d) 4 edges, 4 vertices (exactly 2 of which are odd)

Example 2.16. *All vertices of a graph could be odd.*

(a) True and I am very confident!

(b) True, but I am not very confident.

(c) False and I am very confident!

(d) False, but I am not very confident.

Example 2.17. *All vertices of a graph could be even.*

(a) True and I am very confident!

(b) True, but I am not very confident.

(c) False and I am very confident!

(d) False, but I am not very confident.

(e) I love math!

2.3.1 Paths, Walks, Trails, and more

Often we want to explore various ways in which we can *move around* in a graph especially if our vertices are representing locations. These next few definitions introduce common terminology about moving about a graph.

Definition 2.7. *A* __________ *W is a sequence of vertices with each pair of consecutive vertices in P joined by an edge. We can denote this as $W = (x_1, x_2, ..., x_n)$ or $W = x_1 - x_2 - x_3 - ... - x_n$.*

Definition 2.8. *If a walk starts and ends at the same vertex, we say it is a* ***closed walk*** *otherwise it is an* ***open walk****.*

The ***length*** *of a walk is the number of edges in the walk.*

Definition 2.9. *A* __________ *P is a sequence of distinct (no repeated) vertices with each pair of consecutive vertices in P joined by an edge. We can denote path as $P = (x_1, x_2, ..., x_n)$ or sometimes $P = x_1 - x_2 - x_3 - ... - x_n$.*

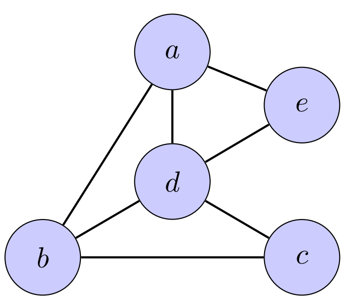
Definition 2.10. *A* __________ *T is a walk without any repeated edges. We can denote path as $T = (x_1, x_2, ..., x_n)$ or sometimes $T = x_1 - x_2 - x_3 - ... - x_n$.*

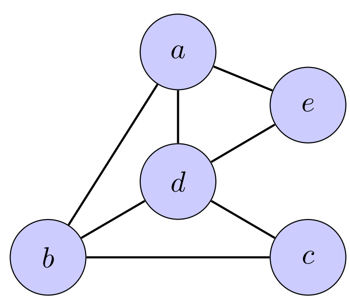
Definition 2.11. *A* ***circuit*** *in a graph G is a closed trail of length 3 or more.*

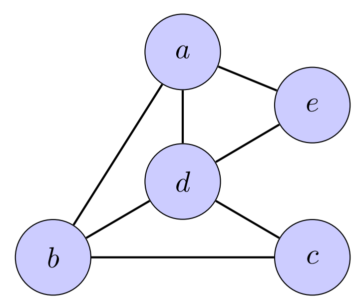
This means a circuit starts and ends at the same vertex but repeats no edges.

Definition 2.12. *A circuit that does not repeat any vertex except the first and last is called a* __________*. A k-cycle is a cycle of length k.*

Examples:







Definition 2.13. *We say a graph G is* __________ *if there is a path between every pair of vertices.*

Note: The Trivial graph is connected since each vertex has a path connecting to itself.

We will talk more about connected graphs later.

2.4 ICE 2: Intro to Graphs

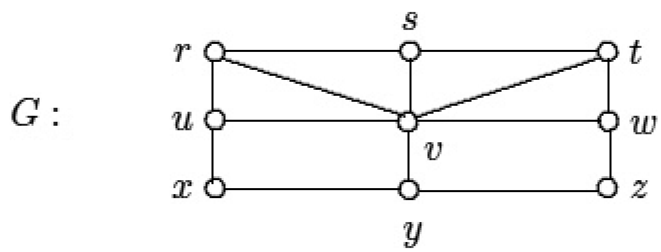


Figure 7:

1. Referring to the graph above, give an example of each or explain why such an example doesn't exist.

   (a) an x-y walk of length 6

   (b) an x-z path of length 3

   (c) a v-w trail that is not a v-w path.

2. Draw a graph with 4 vertices and 5 edges.

3. a) Let $S = \{2, 3, 4, 7, 11, 13\}$ Construct the multigraph M whose vertex set is S and where ij is an edge for **distinct** elements i and j in S whenever $i + j \in S$.

   b) Answer the same problem with this rule: i and j are joined by two edges if **both** $i+j \in S$ and $|i - j| \in S$

4. Draw a graph or pseudograph with 4 vertices (all odd) and 5 edges.

5. Draw a graph with 4 vertices (all even) and 5 edges (no loops).

2.5 Families of Graphs

There are many families of graphs.

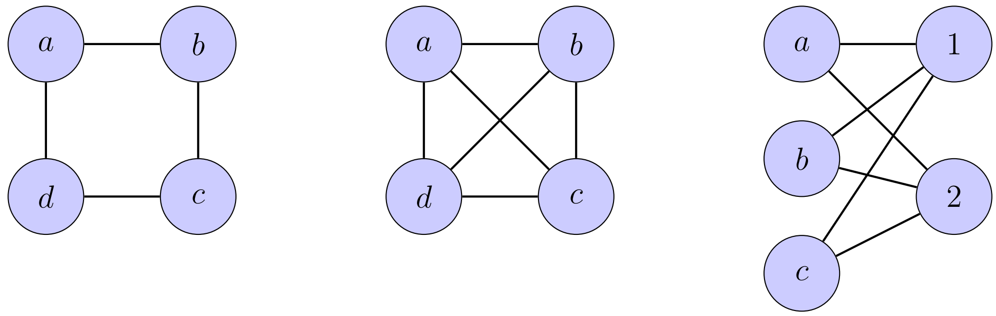


Figure 8: C_4 (left), K_4 (center), and $K_{3,2}$ (right)

2.5.1 Path Graphs

A graph is called a **path graph** if the vertices of G of order n can be labeled $v_1, v_2, ...v_n$ so that its edges are $v_1v_2, v_2v_3, v_3v_4, ..., v_{n-1}v_n$. We denote path graphs as $P_1, P_2, P_3, ...$

2.5.2 Cycle Graphs

A graph is called a **cycle graph** if the vertices of G of order $n \geq 3$ can be labeled $v_1, v_2, ...v_n$ so that its edges are $v_1v_2, v_2v_3, v_3v_4, ..., v_{n-1}v_n$, AND v_nv_1. We denote cycles graphs as $C_1, C_2, C_3, ...$

2.5.3 Complete Graphs

A complete graphs of order n are graphs in which every pair of vertices is connected by an edge. In other words, every two distinct vertices are adjacent. We denote complete graphs as $K_1, K_2, K_3, \ldots$ and these graphs have the maximum possible *size* for a graph with n vertices.

Question: What is the size of K_n?

Depending on the circumstance, sometimes it is helpful to change how one represents a graph. For example, the Complete Graph on 4 Vertices, K_4, can be represented by both graphs in the Figure 9 and, in some cases, it may be nicer to have a representation of K_4 without intersecting edges.

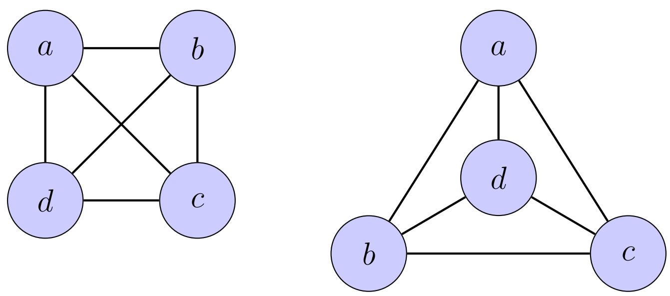


Figure 9: Two representations of K_4.

Example 2.18. *Can you draw some different representations of path graphs, cycle graphs, and complete graphs?*

2.5.4 Bipartite Graphs

Some graphs are classified by rules which have their vertex sets partitioned in particular ways. We call graphs in which the vertices can be partitioned into two sets A and B (called partition sets) in which every vertex in A connects with every vertex in B, but no vertex in A connects to another vertex in A, and no vertex in B connects with another vertex in B, **bipartite graphs.**

Examples:

If G is a bipartite graph with partition sets A and B and if every vertex in A connects with with every vertex in B and every vertex in B connects with every vertex in A, say G is a **complete bipartite graph.** If A has n vertices and B has m vertices, we denote G as $K_{n,m}$.

Example 2.19. *Is C_4 a bipartite graph?*

(a) Yes and I am very confident!

(b) Yes, but I am not very confident.

(c) No and I am very confident!

(d) No, but I am not very confident.

(e) Does this graph come with snacks?

Example 2.20. *Is C_5 a bipartite graph?*

(a) Yes and I am very confident!

(b) Yes, but I am not very confident.

(c) No and I am very confident!

(d) No, but I am not very confident.

(e) Is it time to go home yet?

Theorem 2.1. *A nontrivial graph G is a bipartite graph if and only if G contains no odd cycles.*

Proof in Textbook. We will prove the statement: If a nontrivial graph G is a bipartite graph then G contains no odd cycles. by proving the contrapositive statement:

2.5.5 K-partite Graphs

We can generalize bi-partite graphs for k partition sets instead of 2.
Examples:

2.5.6 Star Graphs

If G is a complete bipartite graph in which one of the partition sets has order 1, G is called a **star graph**.
Examples:

There are many other bipartite graph families including Mongolian Tent Graphs, Book Graphs, Cross-Prism Graphs, Crown Graphs, and more...

2.6 Creating New Graphs

There are several ways to create a new graph from a given graphs. Here are a few:

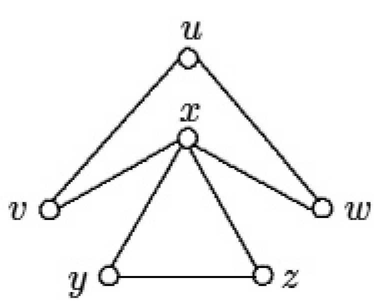
2.6.1 Union Graph

Given graphs G and H, the **union graph** of G and H, denoted, $G \cup H$, is the new graph whose vertex set is $V(G) \cup V(H)$ and whose edge set is $E(G) \cup E(H)$.
Example: $G = 2C_4 \cup P_3 \cup K_4$:

2.6.2 Complement Graph

Given a graph G, the __________ of G, denoted $\bar{G}$, is a graph whose vertex set is $V(G)$ for which each distinct vertex $u, v \in V(G)$, edge uv is in $\bar{G}$ if and only if uv is NOT an edge of G.

Example:



Example 2.21. *Draw* $\bar{C_3}$.

Example 2.22. *What is* $\bar{K_n}$?

2.6.3 Join Graph

Given graphs G and H, the **join** $G + H$ consists of $G \cup H$ and all edges joining a vertex of G and a vertex of H.

Example 2.23. *Draw* $C_4 + P_1$.

2.6.4 Cartesian Product of Graphs

Given graphs G and H, the **Cartesian Graph Product** $G \times H$ (sometimes denoted $G\Box H$) is the graph with vertex set $V(G \times H) = V(G) \times V(H)$. That is every vertex is an ordered pair (u, v) where $u \in V(G)$ and $v \in V(H)$. We also have the following edge property: distinct vertices (u, v) and (x, y) are adjacent in $G \times H$ if either
1) $u = x$ and $vy \in E(H)$ or
2) $v = y$ and $uxy \in E(G)$.

Example 2.24. *Draw* $P_2 \times P_3$.

2.6.5 Hypercubes Graphs

Example 2.25. *Draw* $C_4 \times K_2$

The graph above is sometimes denoted Q_3 and is called a **3-cube.** We can generalize this in the following way.
The graphs Q_n are called **n-cubes** or **hypercubes** and they are defined recursively in the following way:

$Q_1 = K_2$ and $Q_n = Q_{n-1} \times K_2$.

Examples:

These hypercubes can also be defined as the graphs whose vertex set is the set of ordered n-tuples of 0s and 1s (called n-bit strings) and where two vertices are adjacent if their ordered n-tuples differ in exactly one coordinate.

2.7 Digraphs

Remember the "Poor Sportsmanship Graph"? This was an example of a **directed graph** or **digraph.**

Definition 2.14. *A* ***digraph*** *or* ***directed graph*** *D is a finite nonempty set of vertices connected together, where all the edges are directed from one vertex to another. This means our Edge set can be represented by <u>ordered pairs</u> of distinct vertices which are called* ***arcs or directed edges.***

Examples:

Terminology: Given an edge (u, v) in $E(D)$, we say u is adjacent **to** v and v is adjacent **from** u. We say u and v are both incident with edge (u, v).

Note: We could have both edges (u, v) and (v, u) in a digraph. But if **at most** one of (u, v) or (v, u) is in the digraph for each pair of distinct vertices of the digraph, we say the digraph is an **oriented graph.**

Example 2.26. *Suppose Team A beats teams B and C. Team B beats Team D. Team D beats A and C. And C beats B. Draw the directed graph representing the Poor Sportsmanship model AND list the Edges of the digraph as ordered pairs. Is this graph oriented?*

2.8 ICE 3: Classes of Graphs

1. Draw $P_3 \cup P_2$

2. Draw $P_3 + P_2$

3. Is $P_3 \times P_2$ the same as $P_2 \times P_3$ from our notes?

4. What is $P_2 \times P_2$?

5. What is $\bar{K}_n + P_1$

6. What is $\bar{K}_3 + \bar{P}_2$

3 Degrees

There are many numbers which can describe a graph. We have already discussed order and size and degree. Often this isn't enough to describe the full structure of the graph (as we will see in our next section), but knowing the degree of each vertex is a big help for determining information about a given graph.

Recall, the **degree** of a given vertex is the number of edges incident to that vertex.

Definition 3.1. *A vertex of degree 0 is called an* __________ *vertex and a vertex of degree 1 is an* __________ *vertex or* __________.

Definition 3.2. *We denote the minimum degree of a graph G, by* __________
We denote the maximum degree of a graph G, by __________

Example 3.1. *Given a graph of order* n *what are the upper and lower bounds for* $\delta(G)$ *and* $\Delta(G)$*?*

(a) $0 \leq \delta(G) \leq \Delta(G) \leq n$

(b) $0 \leq \delta(G) \leq \Delta(G) \leq n-1$

(c) $1 \leq \delta(G) \leq \Delta(G) \leq n$

(d) $1 \leq \delta(G) \leq \Delta(G) \leq n$

(e) $0 \leq \delta(G) \leq \Delta(G) \leq n+1$

Example 3.2. *Given a graph of order* n *what are the upper and lower bounds for any vertex?*

Example 3.3. *Draw 4 different graphs (not all from the same family). Count the number of edges and vertices for each graph and count the sum of the degrees of the vertices.*

Question: Does there seem to be a relationship between any of these numbers?

Example 3.4. *If you could draw a graph with exactly one odd vertex, what would the sum of all the degrees be?*

Theorem 3.1 (The First Theorem of Graph Theory). *If G is a graph of size m, then*

$$\sum_{v \in V(G)} deg\ v = \underline{\hspace{3cm}}$$

Proof:

Example 3.5. *Suppose we want to construct a graph with 20 edges and have every vertex of degree 4. How many vertices must the graph have?*

Example 3.6. *Suppose a certain graph G has order 14 and size 27. The degree of each vertex is 3,4, or 5. There are six vertices of degree 4. How many vertices of G have degree 3 and how many have degree 5?*

What are some other ways we could have attempted to solve this problem?

Corollary 3.1. *Every graph has an even number of odd vertices.*

Proof:

Theorem 3.2. *Let G be a graph of order n. If for every two nonadjacent vertices u and v of G, deg $u+$ deg $v \geq n-1$, then G is connected and the diam(G)≤ 2.*

Proof: See text.

Corollary 3.2. *If G is a graph of order n with $\delta(G) \geq \dfrac{n-1}{2}$, then G is connected.*

Proof:

Example 3.7. *True or False: A graph of order 7 such that $\delta(G) \geq 3$ is connected.*

(a) True and I am very confident!

(b) True, but I am not very confident.

(c) False and I am very confident!

(d) False, but I am not very confident.

3.1 ICE 4: Degrees

1. Can a graph have nine vertices of which 4 have degree 2, three have degree 3 and two have degree 4?

   (a) True and I am very confident!

   (b) True, but I am not very confident.

   (c) False and I am very confident!

   (d) False, but I am not very confident.

2. Must the number of people at a party who do not know an odd number of other people be even?

3. Must the number of people ever born who had (have) an odd number of brothers and sisters be even?

4. Must the number of families in Alaska with an odd number of children be even?

5. Is it possible to have a group of seven people such that each person knows exactly three other people in the group?

6. How many vertices will the following graphs have if they contain:

   (a) 12 edges and all vertices of degree 3.

   (b) 21 edges, three vertices of degree 4, and the other vertices of degree 3.

   (c) 24 edges and all vertices of the same degree.

7. Consider a group of people at a party. Suppose some guests shook hands with some of the other guests. If we asked everyone at the party how many people they shook hands with and added them all up, what would be the total number of hands shook?

3.2 Regular Graph

Definition 3.3. *If the vertices of a graph G all have the same degree, we say G is __________. If all vertices of G are of degree k, we say G is ____________________.*

Examples:

Example 3.8. *Could you have a k-regular graph of order n if k and n are odd?*

(a) True and I am very confident!

(b) True, but I am not very confident.

(c) False and I am very confident!

(d) False, but I am not very confident.

Theorem 3.3. *Let k and n be integers with $0 \leq k \leq n-1$ Then there exists a k-regular graph of order n if and only if at least one of r and n is even.*

Proof: See text.

3.3 Degree Sequences

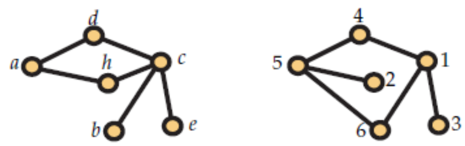
Often we don't have regular graphs, thus it makes sense that we may want to describe our graphs by the degrees the vertices in the graph have.

Definition 3.4. *If the degrees of the vertices of a graph G are listed in a sequence s, then s is called a **degree sequence** of G.*

Definition 3.5. *A finite sequence of nonnegative integers are said to be __________ if it is the degree sequence of some graph.*

Note: There are a lot of algorithms and applications which use this which we will get to later in the course.

Example 3.9. *What is the degree sequence of the graphs below:*



Example 3.10. *Is the degree sequence:* $s_1 = 3, 3, 2, 2, 1, 1$ *graphical?*

(a) Yes and I am very confident!

(b) Yes, but I am not very confident.

(c) No and I am very confident!

(d) No, but I am not very confident.

Example 3.11. *Is the degree sequence:* $s_2 = 6, 5, 5, 4, 3, 3, 3, 2, 2$ *graphical?*

(a) Yes and I am very confident!

(b) Yes, but I am not very confident.

(c) No and I am very confident!

(d) No, but I am not very confident.

Example 3.12. *Is the degree sequence:* $s_3 = 7, 6, 4, 4, 3, 3, 3$ *graphical?*

(a) Yes and I am very confident!

(b) Yes, but I am not very confident.

(c) No and I am very confident!

(d) No, but I am not very confident.

Example 3.13. *Is the degree sequence:* $s_3 = 3, 3, 3, 1$ *graphical?*

(a) Yes and I am very confident!

(b) Yes, but I am not very confident.

(c) No and I am very confident!

(d) No, but I am not very confident.

Theorem 3.4 (Havel-Hakimi Theorem)**.** *A non-increasing sequence $s = d_1, d_2, ..., d_n$ where $n \geq 2$ of non-negative integers where $d_1 \geq 1$ is graphical if and only if the sequence is graphical:*

$$s_1 : d_2 - 1, d_3 - 1, ..., d_{d_1+1}, d_{d_1+2}, ..., d_n$$

Proof: See Text.

Example 3.14. *Is the sequence graphical:* $s : 7, 7, 4, 3, 3, 3, 2, 1$

Example 3.15. *Is the sequence graphical:* $s : 5, 3, 3, 3, 3, 2, 2, 2, 1$

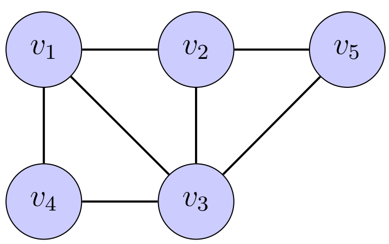
3.4 Matrices

We can describe a graph using $V(G)$ and $E(G)$, but we can also describe a graph using matrices which is especially nice when using computers.

Definition 3.6. *The* ***adjacency matrix*** *of G is the $n \times n$ matrix $A = [a_{ij}]$ such that*
$a_{ij} = \begin{cases} 1 \textit{ if } v_iv_j \in E(G) \\ 0 \textit{ otherwise} \end{cases}$

Definition 3.7. *The* ***incidence matrix*** *of G is the $n \times m$ matrix $B = [b_{ij}]$ such that*
$b_{ij} = \begin{cases} 1 \textit{ if } v_i \textit{ is incident with } e_j \\ 0 \textit{ otherwise} \end{cases}$

Example 3.16. *Find the adjacency and incidence matrices of the graph below.*

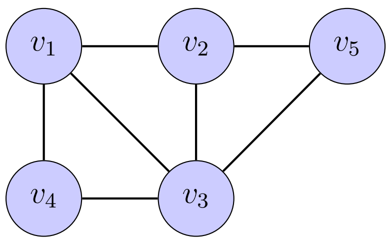


What are some observations/properties of these two matrices?

It turns out power of these matrices also have information!

Theorem 3.5. *Let G be a graph with vertex set $V(G) = \{v_1, v_2, ..., v_n\}$ and adjacency matrix $A = [a_{ij}]$. Then the entry $a_{ij}^{(k)}$ in row i and column j of A^k is the number of distinct $v_i - v_j$ walks of length k in G.*

Note: This means $a_{ii}^{(2)}$ =deg v_i and $a_{ii}^{(3)}$ is twice the number of triangles in G that contain v_i.
Proof: Uses Induction, but we won't go over it because it involves linear algebra which not everyone has taken.



$$A^2 = \begin{bmatrix} 3 & 1 & 1 & 2 & 1 \\ 1 & 2 & 2 & 1 & 1 \\ 1 & 2 & 2 & 1 & 1 \\ 2 & 1 & 1 & 4 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

$$A^3 = \begin{bmatrix} 4 & 5 & 5 & 6 & 2 \\ 5 & 2 & 2 & 6 & 1 \\ 5 & 2 & 2 & 6 & 1 \\ 6 & 6 & 6 & 4 & 4 \\ 2 & 1 & 1 & 4 & 0 \end{bmatrix}$$

4 Isomorphic Graphs

So far we have been dancing around the following question:
"How can we tell if two graphs are really the same graph, but are just drawn differently and with different names for the vertices?"

We have noticed that graphs are essentially the same graph even if they aren't drawn the same way. We call graphs that have the same number of vertices and identical connections (but may look different) **isomorphic.**

Example:

Who cares? Here are some examples where the isomorphism question comes into play:
"Researchers working with organic compounds build up large dictionaries of compounds that they have previously analyzed. When a new compound is found, they want to know if it is already in the dictionary. Large dictionaries can have many compounds with the same molecular formula but differing in their structure as graphs (and possibly in other ways). Then one must test the new compound to see if its graph-theoretic structure is the same as the structure of one of the known compounds with the same formula (and the same in other ways)that is, whether the new compound is graph-theoretically isomorphic to one of a set of known compounds.

A similar problem arises in designing efficient integrated circuitry for a computer. If the design problem has already been solved for an isomorphic circuit (or if a piece of the new network is isomorphic to a previously designed circuit), then valuable savings in time and money are possible."[7]

Informal definition of isomorphic graphs: Two graphs are **isomorphic** if one can be "stretched" or "shifted" to look like the other without adding, breaking, or removing any parts.

Note: You can think of the vertices as steel balls and the edges as rubber bands. We assume balls will remain in whatever position we place them and that the rubber bands never break. Using this idea, two graphs are isomorphic if we can move the balls around to make one graph look like the other.

Definition 4.1. ***Formal definition of isomorphic graphs**: Graphs G and H are **isomorphic** if there exists a bijective function $f : G \to H$ that preserves adjacencies of the vertices of G.*

In other words, f is injective (or __________) and surjective (or __________) and if vertices x and y in G are adjacent, so are $f(x)$ and $f(y)$ in H.

[7] Tucker's *Applied Combinatorics*

Definition 4.2. *We say that the function $f : X \to Y$ is* ***injective*** *if the following property is true: Whenever $f(x_1) = f(x_2)$ it must be true that* ________ = ________.

Notice, in Figure 10, we see that the transformation represented on the left is 1-1, but the transformation represented on the right is not because both v_1 and v_2 map to w_4, but $v_1 \neq v_2$.

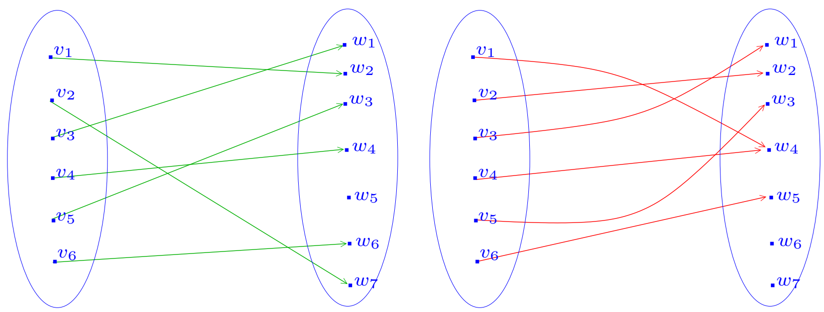


Figure 10:

Process to show Injectivity: Start with two arbitrary elements in the range of f and suppose they are equal, that is suppose $f(u) = f(v)$ and work backwards to see if $u = v$. In other words, can we find two possible input values that map to the same output? If yes, f is NOT injective.

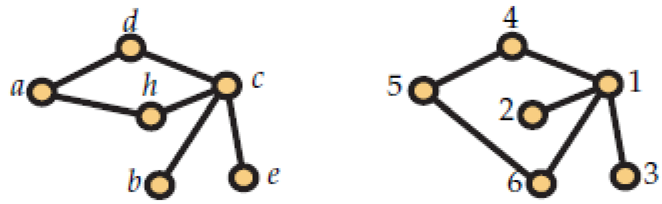
Definition 4.3. *We say that the function $f : X \to Y$ is* ***surjective*** *if every element in Y is mapped to. That is, if $y \in Y$, then there exists a $x \in X$ so that* ________ = ________.

Idea: Dr. Harsy is going to do her absolute worst to present you with the most obscure element of the codomain. Are you certain that this function will be able to map to this element no matter what Dr. Harsy presents you with?

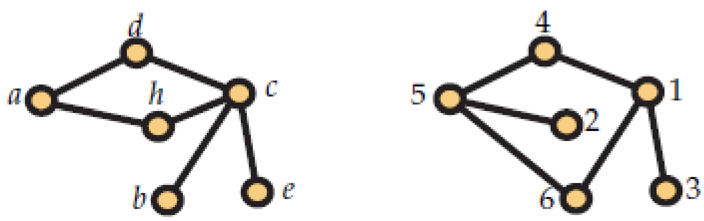
Process to show surjectivity: Pick an arbitrary element in co-domain, find an element that maps to it.

Example 4.1. *Consider the function $f : \mathbb{R} \to \mathbb{R}$ where $f(x) = x^2$. Note that this is neither injective nor onto. Why?*

Example 4.2. *Create a bijective map between these two graphs in order to show these graphs are isomorphic.* [8]



Example 4.3. *Are these graphs isomorphic?*



Example 4.4. *There are 11 non-isomorphics graphs of order 4. Let's find them!*

[8] image from Tucker's *Applied Combinatorics*

Example 4.5. *If two graphs have the same size and order, they are isomorphic.*

(a) True and I am very confident!

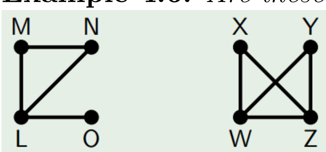
(b) True, but I am not very confident.

(c) False and I am very confident!

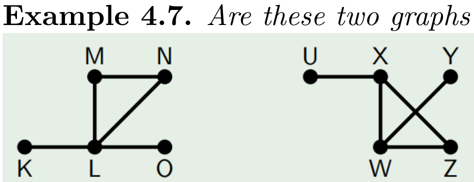
(d) False, but I am not very confident.

(e) Snack time?

Example 4.6. *Are these two graphs isomorphic?*



Example 4.7. *Are these two graphs isomorphic?*



4.1 Useful Theorems to help with the Isomorphism Question

Determining whether two graphs are isomorphic is one of the fundamental and notoriously challenging questions in the field of graph theory. Luckily we have some tools (theorems) which can help us!

4.1.1 Degree Sequences

Theorem 4.1. *If two graphs are isomorphic, they have the same degree sequence.*

Proof:

Example 4.8. *Is the converse of this statement true, "If two graphs have the same degree sequence are the graphs isomorphic?" In other words, if two graphs have the same size, order, and degree sequence, must they be isomorphic?*

(a) Yes and I am very confident!

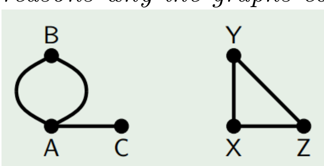
(b) Yes, but I am not very confident.

(c) No and I am very confident!

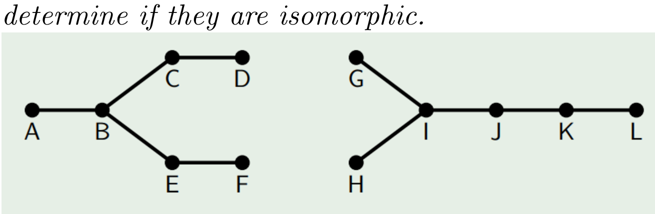
(d) No, but I am not very confident.

(e) More snacks?

Example 4.9. *For the graphs below, give reasons why the graphs could be isomorphic and give reasons why the graphs could not be isomorphic.*



Example 4.10. *Consider the two graphs below. What is the degree sequence of each graph and determine if they are isomorphic.*



4.1.2 Subgraphs

Definition 4.4. *A subgraph G' of a graph G is a graph formed by a subset of vertices and edges of G. We write $G' \subseteq G$.*

Definition 4.5. *If a subgraph G' of a graph G has the <u>same</u> vertex set as G, we say G' is a ______________ ________ of G.*

Definition 4.6. *We say F is an ________ subgraph of G if whenever u and v are vertices of G and uv is an edge of G, then uv is an edge of F.*

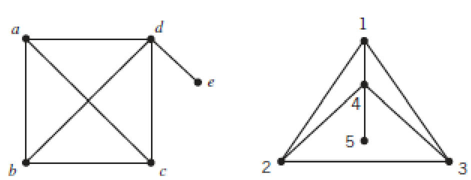
Examples:

Theorem 4.2. *If two graphs are isomorphic, then subgraphs formed by corresponding vertices and edges must be isomorphic.*

Proof: Omitted.

Subgraphs can be used to test for isomorphism in the following way. If a graph G has a set of six vertices forming a chordless circuit of length 6 (chordless means there are no other edges between these six vertices except the six edges forming the circuit), then any graph isomorphic to G must also have a set of six vertices forming such a chordless 6-circuit.

Example 4.11. *Use a subgraph argument to determine whether the graphs below are isomorphic:*



4.2 Other Theorems

Theorem 4.3. *Two graphs are isomorphic if and only if their complements are isomorphic.*

Proof: Omitted.

Example:

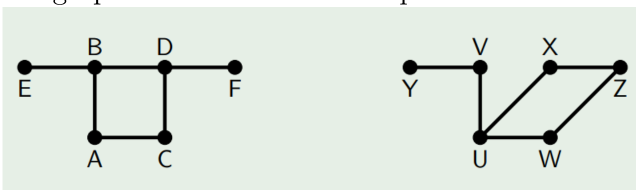
Theorem 4.4. *If G and H are isomorphic then G is bipartite if and only if H is bipartite*

Theorem 4.5. *If G and H are isomorphic then G is connected if and only if H is connected*

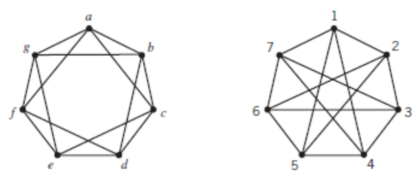
Definition 4.7. *Two Digraphs D and D' are* ***isomorphic*** *if there exists a one to one correspondence $f : V(D) \to V(D')$ such that $(u, v) \in E(D)$ if and only if $(f(u), f(v)) \in E(D')$.*

4.3 Ice 5: Isomorphisms

1. For the graphs below, give reasons why the graphs could be isomorphic and give reasons why the graphs could not be isomorphic.

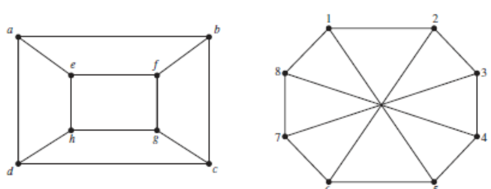


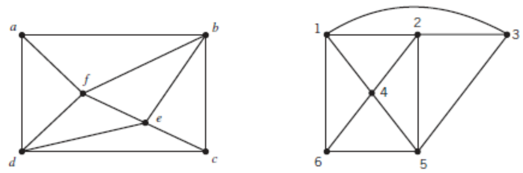
2. Use the complement of these two graphs to determine if the graphs below are isomorphic.

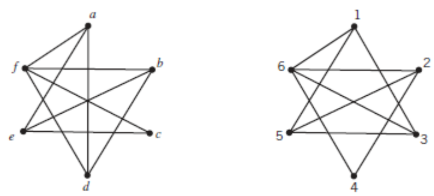


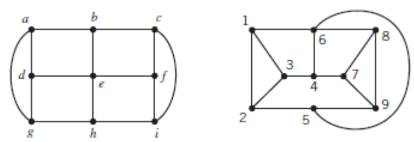
3. Determine whether the following graphs are isomorphic. Explain your reasoning or state the isomorphism.

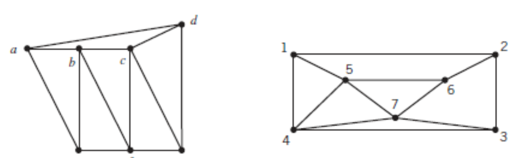
(a)




a
b
f
e
d
c
1
2
3
4
6
5
(b)


a
f
b
e
c
d
1
6
2
5
3
4
(c)


a
b
c
d
e
f
g
h
i
1
6
8
3
7
4
2
5
9
(d)


d
a
b
c
e
f
g
1
2
5
6
7
4
3
(e)

5 Trees

The most commonly used graph is called a Tree. They are very useful for organizing information and search procedures (directories in a UNIX operating systems, word sorting, computer networks and parallel computing). We will also revisit these graphs after we discuss Euler and Hamilton Circuits.
There are several ways to define a tree graph.

Definition 5.1. *Tree Definition 1: A* ***Tree Graph*** *is a connected graph with no circuits.*

Definition 5.2. *Tree Definition 2: A* ***Tree Graph*** *is a graph with a designated vertex called a* ***root*** *such that there is a unique path from the root to any other vertex in the tree.*

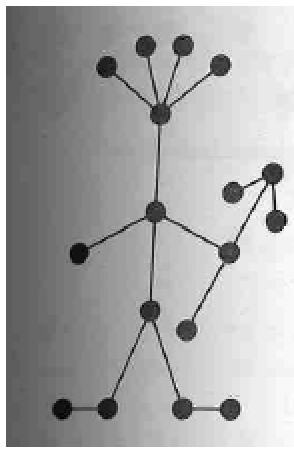


Figure 11: Example of a tree.

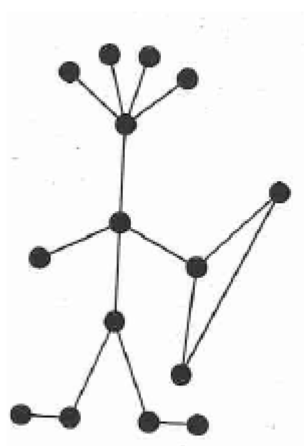


Figure 12: Is this graph a tree?

We say a Tree Graph is __________ if any vertex can be the root of the graph. If we have a directed tree graph we call the graph a __________ **tree**.

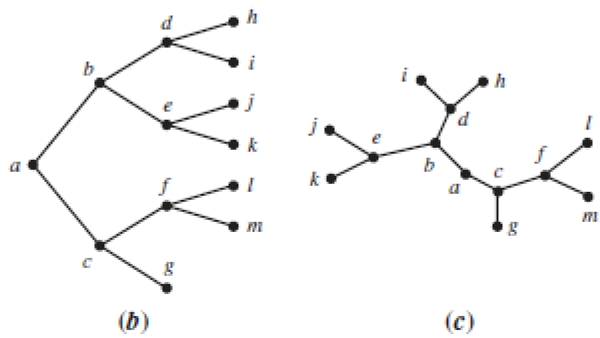


Figure 13: Two pictures of the same graph (Picture from *Applied Combinatorics*).

We usually draw a rooted tree with the root at the top of the figure. We say the root is at level 0. The vertices directly below the root are at level 1, and so on. For any non-root vertex, x, in a rooted tree the **parent** of x is the vertex y with edge (y, x) into x. The **children** of x are the vertices z with directed edge from x to z. Two vertices with the same parents are called **siblings.** Vertices of trees with no children are called **leaves**. If every internal vertex of a rooted tree has m children, we call the tree an **m-ary tree.**

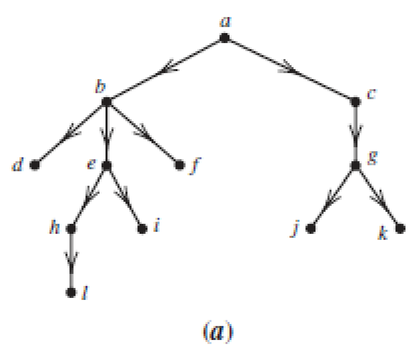


Figure 14: Rooted Trees (Picture from *Applied Combinatorics.*).

Definition 5.3. *A graph which has no circuits is called* __________.

So you can also define a tree as a connected acyclic graph. We call acyclic graphs __________.

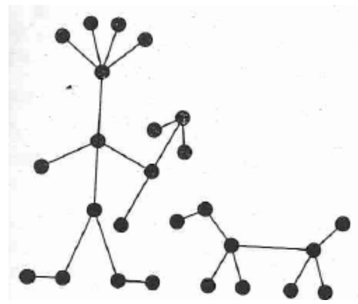


Figure 15: Example of a forest.

Theorem 5.1. *Let T be a connected graph. Then the following statements are equivalent.*
a) T has no circuits.
b) Let a be any vertex in T. Then for any other vertex x in T, there is a unique path P_x between a and x.
c) There is a unique path between any pair of distinct vertices x, y in T.
d) T is minimally connected, in the sense that the removal of any edge of T will disconnect T.

Proof:

Theorem 5.2. *A tree with n vertices has $n-1$ edges.*

Proof:

Theorem 5.3. *Let T be an m-ary tree with n vertices, of which i vertices are internal, then $n = mi + 1$.*

Proof: Homework

Corollary 5.1. *Let T be an m-ary tree with n vertices consisting of i internal vertices and l leaves. If we know one of the parameters n, i, or l, then we can find the other parameters by the following formulas:*

a) Given i, then $l = (m-1)i + 1$ and $n = mi + 1$
b) Given l, then $i = \dfrac{l-1}{m-1}$ and $n = \dfrac{ml-1}{m-1}$
c) Given n, then $l = \dfrac{(m-1)n+1}{m}$ and $i = \dfrac{n-1}{m}$

Example 5.1. *Suppose 63 people sign up for a tennis tournament, how many matches will be played in the tournament?*

Definition 5.4. *The __________ of a rooted tree is the length of the longest path from the root. In other words, it is the largest level number of any vertex.*

Definition 5.5. *A rooted tree of height h is called __________ if all leaves are at levels h and $h-1$.*

Balanced trees are "good" trees. Why is it important for a tennis tournaments tree to be balanced?

Theorem 5.4. *Let T be an m-ary tree of height h with l leaves. Then,*
(a) $l \leq m^h$, and if all leaves are at height h, $l = m^h$.
(b) $h \geq \lceil \log_m l \rceil$, and if the tree is balanced, $h = \lceil \log_m l \rceil$.

Proof of a)

Determining the number of leaves and height of more complex search trees is a major concern in the field of computer science called analysis of algorithms. Below is formula for the number of different undirected trees on n labeled vertices. This formula was first proved by Cayley in 1889.

Theorem 5.5. *There are n^{n-2} different undirected trees on n labels.*

Proof: Omitted.

5.1 Search Trees and Spanning Trees

Trees are used to find solutions to problems that involve a sequence of choices, whether hunting through a graph for a particular vertex (which may represent something like the cheapest solution). Many classic graph theory problems which involve isomorphisms, Hamilton circuits, minimal colorings, and optimal graphical modeling use tree-based searching for computerized solutions. By letting the sequential choices be internal vertices in a rooted tree and the solutions and representing "dead ends" by leaves, trees help us better organize and solve such problems. The main concern we have when using trees to solve these problems is that we want to design it so the search is exhaustive -that is we want to make sure we have checked ALL possible options.

The field of Optimization Research uses very large trees and special "pruning" algorithms to solve their problems. We won't go into any major research questions, but the ideas from this section may show up later when we discuss network algorithms and the traveling salesman problem.

Recall, A spanning subgraph of G is a subgraph that contains all __________ of G. Thus the following definition makes a lot of sense:

Definition 5.6. *A __________ __________ of G is a tree containing all vertices of G.*

Note, unlike spanning subgraphs, if a graph is not connected, we will not be able to construct a spanning tree.

There are 2 traditional ways to construct a spanning tree:
Creating a depth-first spanning tree:

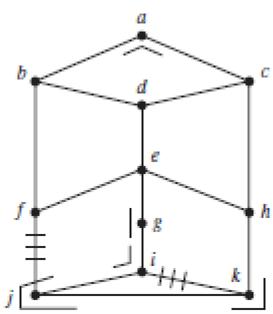
1. Pick some vertex as the root and begin building a path from the root composed of edges of the graph.
2. Continue this path until it cannot go any further without repeating a vertex already in the tree.
3. We will have a leaf at the vertex where this path is forced to stop.
4. Now backtrack to the parent of the leaf and try to build a path from the parent, say vertex y, in another direction.
5. Continue until all possible paths from this parent (y) and children have been built.
6. Now backtrack to find the parent of vertex y and repeat this process.
7. Stop when we have come back to the root and have checked all other possible paths from the root.

Creating a breadth-first spanning tree:

1. Pick some vertex as the root, say x and put in all edges leaving x (along with the vertices at the ends of these edges) in the tree.
2. Add to the tree the edges leaving the vertices adjacent from x, unless such an edge goes to a vertex already in the tree.
3. Continue to build the tree in this level-by-level manner.

Note: If we can create a spanning tree using either of these methods, this means the graph is connected. So this process could be used to check whether a graph is connected.

Example 5.2. *Find a depth-first spanning trees for the graph below:*



Example 5.3. *Find a breadth-first spanning trees of the graph above.*

Example 5.4. *Use Spanning Trees to determine whether the graph with the following adjacency matrix is connected:*

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	0	1	1	1	0	1	0	0
x_2	1	0	1	0	1	0	1	0
x_3	1	1	0	0	0	0	0	0
x_4	1	0	0	0	0	0	1	0
x_5	0	1	0	0	0	0	1	0
x_6	1	0	0	0	0	0	0	0
x_7	0	1	0	1	1	0	0	1
x_8	0	0	0	0	0	0	1	0

Proposition 5.1. *Show that in an n-vertex graph, a set of n-1 edges that form no circuits is a spanning tree.*

Proof:

Proposition 5.2. *A breadth-first spanning tree consists of shortest paths from the root to every other vertex in the graph.*

Proof: Omitted.

Example 5.5. *Suppose we are given three pitchers of water, of sizes 10 quarts, 7 quarts, and 4 quarts. Initially the 10-quart pitcher is full and the other two empty.We can pour water from one pitcher into another, pouring until the receiving pitcher is full or the pouring pitcher is empty. Is there a way to pour among pitchers to obtain exactly 2 quarts in the 7- or 4-quart pitcher? If so, find a minimal sequence of pourings to get 2 quarts in the 7-quart or 4-quart pitcher.*

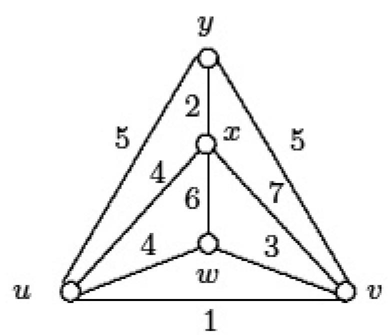
5.1.1 Minimal Spanning Tree Algorithms for Weighted Trees

There are several algorithms for finding a minimum spanning tree. We will talk about 2 of them.

Kruskal's Algorithm:
Given a connected weighted graph G, we can construct a spanning tree T of G in the following way:

1. Start with the vertex set of G.
2. Have the first edge e_1 be any edge of G of minimum weight.
3. For e_2 select any remaining edge of G with minimal weight.
4. For e_3 select any remaining edge of minimum weight that does not produce a cycle.
5. Continue until a spanning tree is produced.

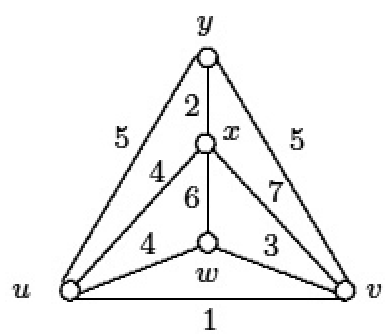
Example 5.6. *Use Kruskal's Algorithm to create a spanning tree for the following graph:*



Prim's Algorithm:
Given a connected weighted graph G, we can construct a spanning tree T of G in the following way:

1. Start with the vertex set of G.
2. Pick an arbitrary vertex, v, of G.
3. For e_1 select an edge incident with v with minimal weight.
4. For e_2, e_3, etc. select incident edges of minimum weight among the edges having exactly one of its vertices incident with an edge already selected.
5. Continue until a spanning tree is produced.

Example 5.7. *Use Prim's Algorithm to create a spanning tree for the following graph:*



5.2 Ice 6- Trees

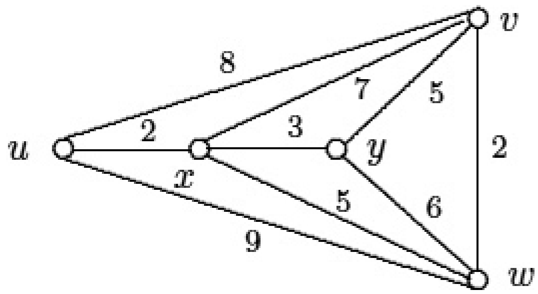
1. Suppose a telephone chain is set up among 100 parents to warn of a school closing. It is activated by a designated parent who calls a chosen set of three parents. Each of these three parents calls given sets of three other parents, and so on. How many parents will have to make calls? Repeat the problem for a telephone tree of 200 parents.

2. A tree T has order 13 with degrees of its vertices only be 1,2, and 5. If T has exactly 3 vertices of degree 2, how many end-vertices does it have?

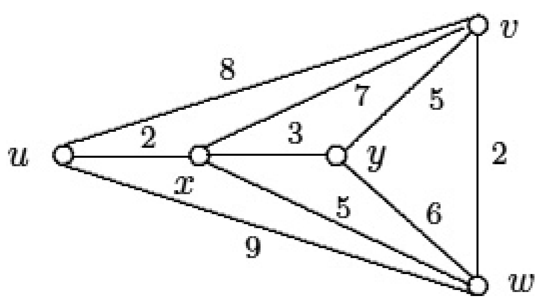
3. Find a depth-first spanning tree for K_6.

4. Find a breadth-first spanning tree for K_6.

5. Use Kruskal's Algorithm to create a spanning tree for the following graph:



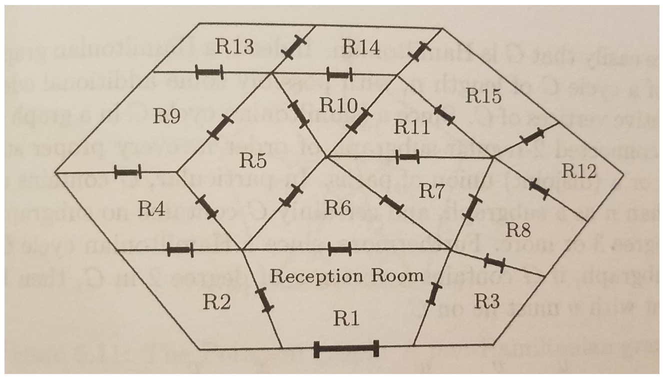
6. Use Prim's Algorithm to create a spanning tree for the following graph:



6 Traversability

6.1 Hamiltonian Graphs

The figure below shows a diagram of an art museum that is divided into 15 exhibition rooms. At the end of the day, a security officer enters the reception room by the front door and checks each exhibition room to make certain that everything is in order. Can the officer make a round trip to visit each room only once and return to the reception room?

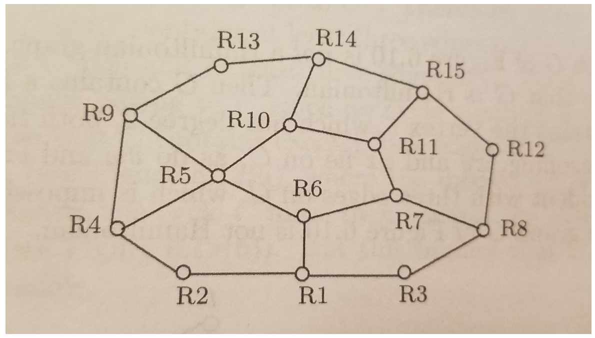


How could we turn this situation into a graph? What does the question become?

Definition 6.1. *A cycle in a graph G that contains every vertex of G is called a* ***Hamiltonian cycle*** *of G. A* ***Hamiltonian graph*** *is a graph that contains a Hamiltonian cycle. A path in G that contains every vertex of G is called a* ***Hamiltonian path****.*

Example 6.1. *List some families of graphs which are Hamiltonian.*

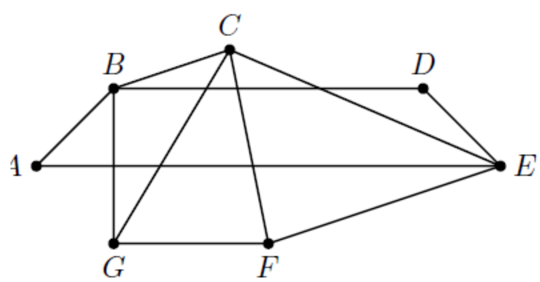
Example 6.2. *The graph below is the art museum situation from the first page.*



Is G Hamiltonian? Does G have a Hamiltonian path?

Question: If G contains a Hamiltonian cycle, must G contain a Hamiltonian path?

Example 6.3. *Show that the following graph is not Hamiltonian.*



6.2 Activity: Constructing Disjoint Hamiltonian Cycles of K_{2n+1}

Recall a **partition** of a nonempty set A is a collection of nonempty subsets of A such that every element of A belongs to exactly one of these subsets. In other words, a partition of A is a collection of pairwise disjoint nonempty subsets of A whose union is A.

Definition 6.2. *A graph G is said to be* ***decomposable*** *into the subgraphs H_1, $H_2, \dots H_k$ if $\{E(H_1), E(H_2), \dots, E(H_k)\}$ is a partition of $E(G)$. Such a partition produces a* ***decomposition*** *of G.*

Example 6.4. *We can decompose G below into a cycle and a path.*

See handout for activity.

Theorem 6.1. *The complete graph K_{2n+1} has a decomposition into n disjoint Hamiltonian cycles.*

Proof Idea. The algorithm that proves this was our class activity. $\square$

Theorem 6.2. *The complete graph K_{2n} has a decomposition into n disjoint Hamiltonian paths and a decomposition into $n-1$ disjoint Hamiltonian cycles and a matching.*

(We will study matchings on a graph at a later time.)

Proof Idea. Run the algorithm from above and delete vertex $2n+1$. $\square$

Example 6.5. *If 17 students dine together at a circular table during a conference, and if each night each student sits next to a pair of different students, how many days can the conference last?*

Definition 6.3. *A graph G that is not connected is called* ***disconnected****. A connected subgraph of G that is not a proper subgraph of any other connected subgraph of G is a* ***component*** *of G. The number of components of G is denoted $k(G)$.*

Thus a graph G is connected if and only if _____________.

Theorem 6.3. *If G is a Hamiltonian graph, then for every nonempty proper set S of vertices of G,*

$$k(G - S) \leq |S|.$$

Proof:

The above theorem provides a _____________ condition for G to be Hamiltonian. Thus we can use the contrapositive to prove G is not Hamiltonian.

Contrapositive:

Example 6.6. *Show the graph below is non-Hamiltonian.*

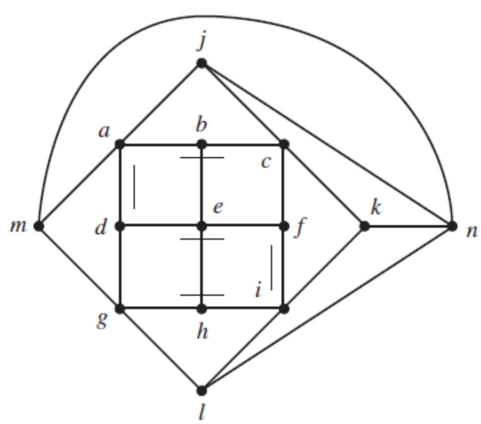


Figure 2.6

The following result provides a *sufficient* condition for a graph to be Hamiltonian.

Theorem 6.4. *Let G be a graph of order $n \geq 3$. If*

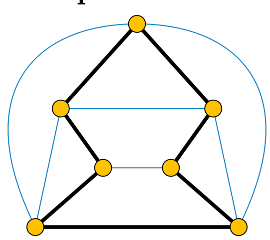
$$\deg u + \deg v \geq n$$

for each pair u, v of nonadjacent vertices of G, then G is Hamiltonian.

Proof Sketch.

$\square$

Example 6.7. *Show the following graph is Hamiltonian using the Theorem above.*



Corollary 6.1. *Let G be a graph of order $n \geq 3$. If $\deg v \geq n/2$ for each vertex v of G, then G is Hamiltonian.*

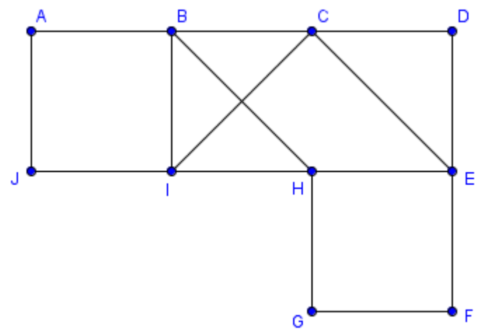
Proof.

$\square$

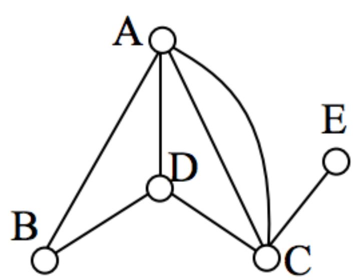
Conclusion: Although Eulerian circuits and Hamiltonian cycles seem to have similar definitions, it is now clear they are dramatically different. Determining whether G is Euclidean is straightforward; however, determining whether G is Hamiltonian can be **extremely difficult** because there is no useful characterization of Hamiltonian graphs to this day.

6.3 ICE 7: Eulerian and Hamiltonian Graphs

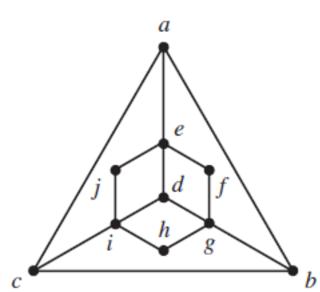
1. Determine if the following graph is Eulerian, Hamiltonian, both, or neither. Support your answer.

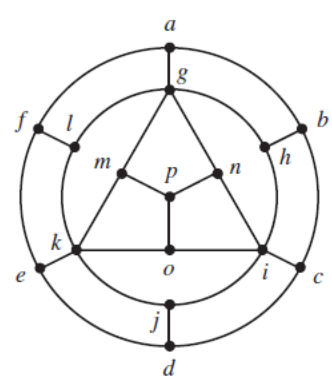


2. Find an Eulerian trail and a Hamiltonian path in the following graph.

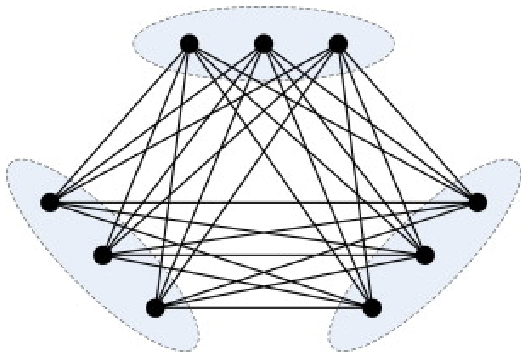


3. Prove that the following graphs are not Hamiltonian.



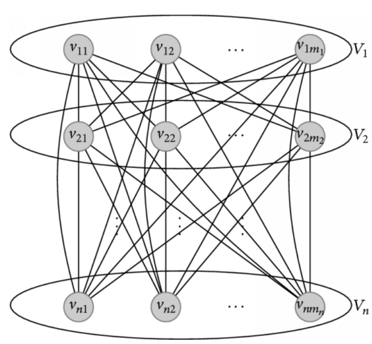


4. Show the following **complete tripartite** graph, $K_{3,3,3}$ is Hamiltonian.



5. When is $K_{\ell,m,n}$ for $\ell, m, n \in \mathbb{Z}$ Hamiltonian? Provide some families of $K_{\ell,m,n}$ which are and are not Hamiltonian.

6. Try to answer question Question 5 for any multipartite graph.

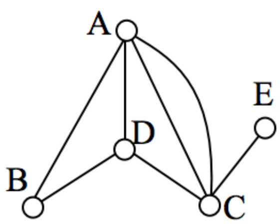


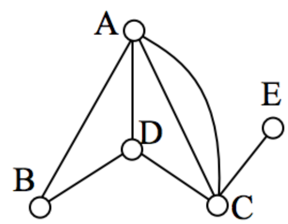
7 Digraphs

Recall a **digraph**, also known as a **directed graph**, D consists of a vertex set V and an edge set E. The ordered pairs of E are **arcs** or **directed edges**. If for each distinct $u, v \in V$, at most one of (u, v) and (v, u) is a directed edge of D, then D is an **oriented graph**. Here, D is referred to as an **orientation** of G.

Definition 7.1. *If $(u, v) \in E(D)$, then u* **is adjacent to** *v and v* **is adjacent from** *u. The number of vertices to which a vertex v is adjacent is the* **outdegree of** *v and is denoted $\text{od}(v)$. The number of vertices from which v is adjacent is the* **indegree of** *v, denoted $\text{id}(v)$.*

Example 7.1. *Draw two different orientations of the following graph. Then find the indegrees and outdegrees for each orientation.*



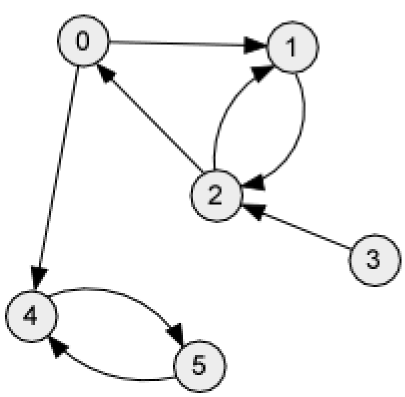


Theorem 7.1 (The First Theorem of Digraph Theory)**.** *If D is a digraph of size m with $V(D) = \{v_1, v_2, \ldots, v_n\}$, then*

$$\sum_{i=1}^{n} \text{od}(v_i) = \sum_{i=1}^{n} \text{id}(v_i) = m.$$

Proof. Each $(u, v) \in E(D)$ contributes to ______________ and ______________. □

Example 7.2. *Consider the following digraph to find the following:*



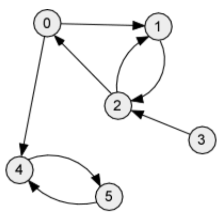
1. **(directed)** $2-4$ **walk** *of length 6:*
2. **(directed) trail** *of length 6:*
3. **(directed) path** *of length 4:*
4. **(directed) cycle** *of length 3:*
5. **(directed) circuit** *of length 5:*

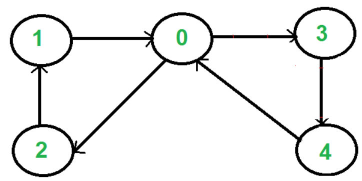
The **underlying graph** G of a digraph D is obtained by replacing each arc (u, v) or the pair $(u, v), (v, u)$ by the edge uv. Draw the underlying graph from the digraph above.

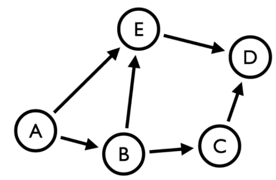
7.1 Strong Digraphs

Definition 7.2. *A digraph D is* **connected** *(or* **weakly connected***) if the underlying digraph of D is connected. A digraph is* **strong** *(or* **strongly connected***) if D contains both a $u-v$ path and a $v-u$ path for every pair u, v of distinct vertices of D.*

Example 7.3. *Classify each of the digraphs below as* **connected**, **strong**, *or neither.*







Theorem 7.2. *If a digraph D contains a $u-v$ walk of length ℓ, then D contains a $u-v$ path of length at most ℓ.*

Proof Idea. Refer to Example 7.2 □

Theorem 7.3. *A digraph D is strong if and only if D contains a closed spanning walk.*

Proof.

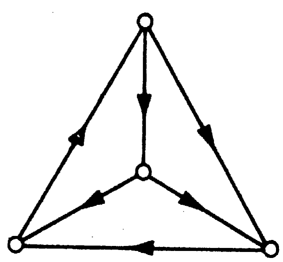
□

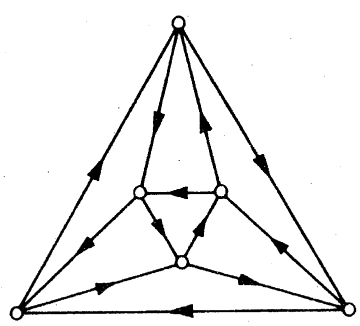
Example 7.4. *Two people A and B play a game on the complete graph K_4. These two people alternate assigning a direction to an edge, beginning with A. It is the goal of A to produce a strong orientation of K_4 while it is the goal of B to stop A from doing this. If both players play perfectly, who will win the game? On which turn will the game end?*

Come up with a variation of this game and determine the outcome.

Definition 7.3. *An* **Eulerian circuit** *in a (strong) digraph D is a circuit containing every arc of D. If a D contains an Eulerian circuit, then D is an* ____________________.

Example 7.5. *Determine if the following graphs are Eulerian. Can you come up with a characterization of Eulerian digraphs?*





Theorem 7.4. *A nontrivial connected digraph D is Eulerian if and only if* $\mathrm{od}(v) = \mathrm{id}(v)$ *for every vertex v of D.*

Proof Idea.

$\square$

Theorem 7.5. *A nontrivial connected graph G has a strong orientation if and only if G contains no bridges.*

Proof.

$\square$

7.2 Tournaments

Definition 7.4. *A* **tournament** *is an orientation of a complete graph. Thus a* **tournament** *can be defined as a digraph such that for every pair of distinct vertices u, v, exactly one of $(u, v), (v, u) \in E(D)$. A tournament T models a round robin tournament. The vertices of T are the ______________, and (u, v) is an arc in T if ____________________________________.*

Example 7.6. *The top 8 male tennis players compete in the ATP world finals at the end of each year. The players are split into two groups of 4 where they complete a round robin stage. The top 2 players in each group advance to a semifinal round. Last year one of the groups consisted of players: Cilic, Djokovic, Isner, and Zverev.*

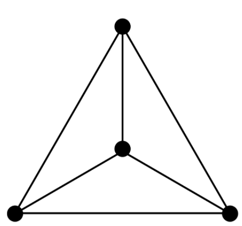
1. *Draw the corresponding tournament from the results shown below.*

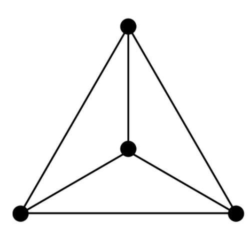
Winner	*Loser*	*Score*
Zverev	*Cilic*	*7-6(5), 7-6(1)*
Djokovic	*Isner*	*6-4, 6-3*
Djokovic	*Zverev*	*6-4, 6-1*
Cilic	*Isner*	*6-7(2), 6-3, 6-4*
Zverev	*Isner*	*7-6(5), 6-3*
Djokovic	*Cilic*	*7-6(7), 6-2*

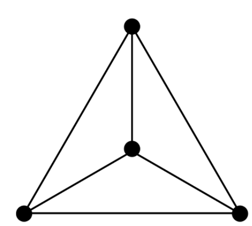
2. *Use the digraph to determine which two players advanced. In general, which team will win the tournament?*

***Zverev ends up beating Djokovic in the final round.*

3. *There are* 4 *non-isomorphic tournaments of order* 4*. Draw the other* 3*.*







4. *In each tournament of order 4 above, indicate the winning vertex (if one exists).*

Definition 7.5. *A tournament T is* **transitive** *if whenever (u, v) and (v, w) are arcs of T, then (u, w) is also an arc of T.*

Example 7.7. *Which order 4 tournaments from above are transitive?*

Example 7.8. *Draw a transitive tournament of order 3 and a transitive of order 5. In general, how many transitive tournaments of order n are there?*

Remark 7.1. *If T is a transitive tournament of order n and i is an integer with $0 \leq i \leq n - 1$,*

____.

Theorem 7.6. *A tournament T is transitive if and only if T contains no cycles.*

Proof.

□

Definition 7.6. *Let u, v be vertices of a digraph D. The* **distance** $\vec{d}(u, v)$ **from** u **to** v *is the length of a shortest $u - v$ path in D. A $u - v$ path of length $\vec{d}(u, v)$ is a $u - v$* **geodesic**.

Remark 7.2. *In order for $\vec{d}(u, v)$ to be defined for every pair of vertices $u, v \in V(D)$, then D must be* _______________.

Theorem 7.7. *If u is a vertex of maximum outdegree in a tournament T, then $\vec{d}(u, v) \leq 2$ for every vertex v of T.*

Proof.

□

Theorem 7.8. *Every tournament contains a Hamiltonian path.*

Proof.

□

Theorem 7.9. *A nontrivial tournament T is Hamiltonian if and only if T is strong.*

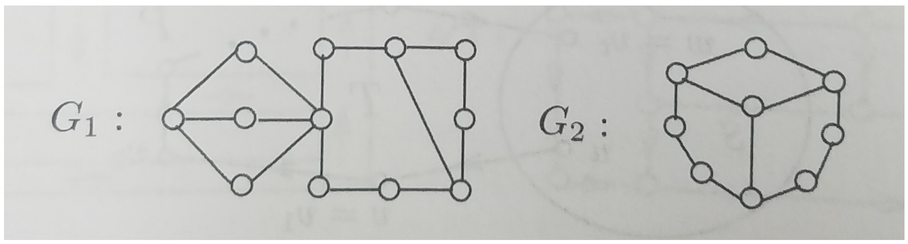
Proof. See book.

□

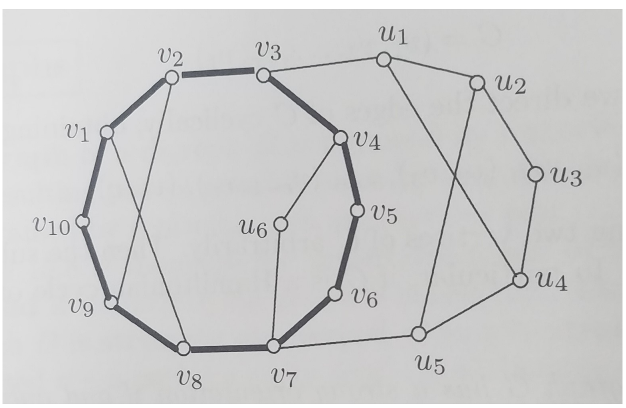
7.3 ICE 8: Digraphs and Tournaments

1. How many orientations does a simple graph G of order n have?

2. Show that each of the graphs G_1 and G_2 is orientable by assigning a direction to each edge so that the resulting digraph is strong.



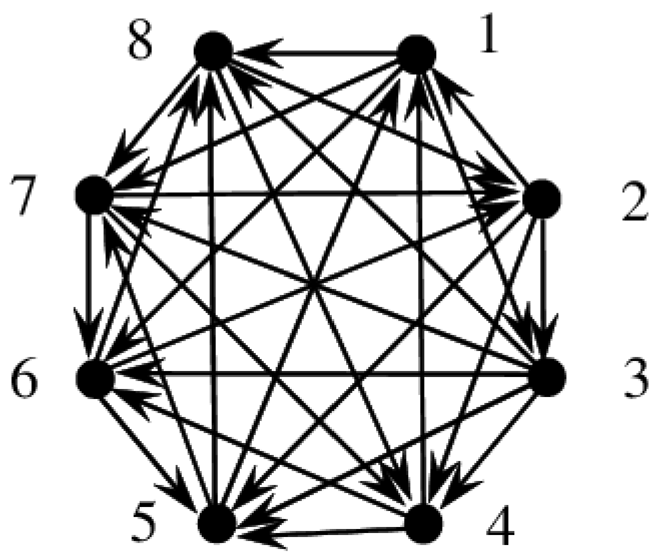
3. Use the proof of the converse from Theorem 7.5 to find a strong orientation of the graph below. Start by directing the cycle $C'' = (v_1, v_2, \ldots, v_{10}, v_1)$.



4. Is it possible to have a tournament with 6 teams where 3 of the teams win 3 games and the other 3 teams win 2 games?

5. Let u and v be distinct vertices in a tournament such that $\vec{d}(u, v)$ and $\vec{d}(v, u)$ are defined. Show that $\vec{d}(u, v) \neq \vec{d}(v, u)$.

6. Consider the tournament below to answer the following questions.



(a) Which team wins this tournament?

(b) Is this digraph strong? Explain your reasoning.

(c) Is this tournament transitive? Explain your reasoning.

(d) Is this digraph Eulerian? Explain your reasoning.

(e) Is this digraph Hamiltonian? Explain your reasoning.

8 Network Algorithms

Previously we studied the problem of finding a minimum spanning tree using *Kruskal's* and *Prim's* Algorithms. In this chapter, we continue the discussion of algorithms for the solutions of the important network optimization problems: shortest paths and maximum flows. By a **network** we mean a graph with a non-negative integer $k(e)$ assigned to each edge e. This integer will typically represent the *length* or *cost* or *capacity* of an edge in various units.

8.1 Shortest Path

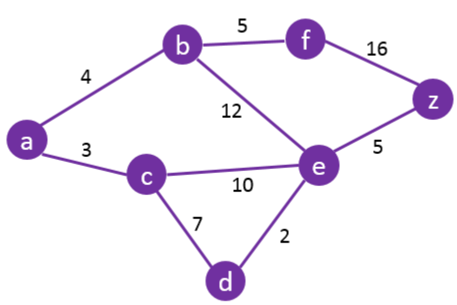
The following algorithm gives shortest paths from a given vertex a to all other vertices. Let $k(e)$ denoted the length of edge e. Let the variable m be a *distance counter*.

For increasing values of m, the algorithm labels vertices whose minimum distance from vertex a is m. The label on each vertex v is of the form $(-, -)$, where the first blank will be the previous vertex on the shortest path from a to v, and the second blank will be the length of the shortest path from a to v.

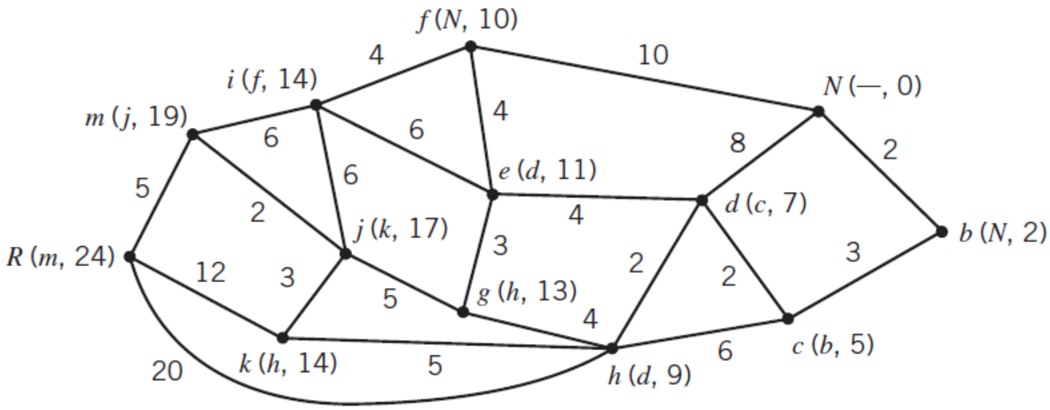
Shortest Path Algorithm (due to Dijkstra):

1. Set $m = 1$ and label vertex a with $(-, 0)$.
2. Check each edge $e = (p, q)$ from some labeled vertex p to some unlabeled vertex q. Suppose p's labels are $(r, d(p))$. If $d(p) + k(e) = m$, label q with (p, m).
3. If all vertices are not yet labeled, increment m by 1 and go to Step 2. Otherwise go to Step 4. If we are only interested in a shortest path to z, then we go to Step 4 when z is labeled.
4. For any vertex y, a shortest path from a to y has length $d(y)$, the second label of y. Such a path may be found by backtracking from y (using the first labels).

Example 8.1. *Find the shortest path from a to z using the shortest path algorithm.*



Example 8.2. *Suppose you wish to find a shortest path from point N (Niagara Falls) to point R (Reno) in the road network shown below. You apply the shortest path algorithm to get the labels shown below. What is the shortest path?*



Example 8.3. *Use the shortest path algorithm to find the shortest path between vertex c and vertex m in the previous example.*

8.2 Network Flows

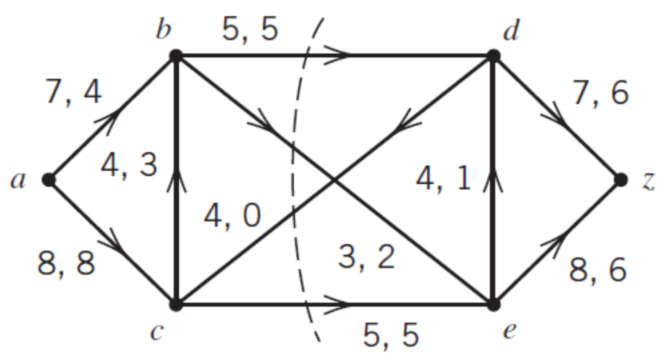
In this section, all networks are direced. We interpret the integer $k(e)$ associated with edge e as a capacity. Our goal is to maximize a *flow* between a *source* vertex and a *sink* vertex such that the flow in each edge does not exceed that edge's capacity.

Definition 8.1. *An $a - z$* **flow** *in a directed network N is an integer-valued function $f(e)$ defined on each edge e, $f(e)$* **is the flow in** *e, together with a* **source** *vertex a and a* **sink** *vertex z satisfying the following conditions:*

1. $0 \leq f(e) \leq k(e)$
2. *For every non-source, non-sink vertex x,*
$$\sum_{e \in In(x)} f(e) = \sum_{e \in Out(x)} f(e).$$
$In(x)$, $Out(x)$ denote the sets of edges directed into and out from vertex x, respectively.
3. $f(e) = 0$ *if* $e \in In(a)$ *or* $e \in Out(z)$.

Definition 8.2. *The* **value** *of the $a - z$ flow $f(e)$, denoted $|f|$, equals the sum of the flow in edges coming out of a.*

Example 8.4. *A sample single-source, single-sink network with flow.*



Definition 8.3. *Let $(P, \overline{P})$ denote the set of all edges (x, y) with vertex $x \in P$ and $y \in \overline{P}$. The set $(P, \overline{P})$ is a* **cut**. *Further we call $(P, \overline{P})$ an* **a - z cut** *if $a \in P$ and $z \in \overline{P}$.*
The **capacity** *$k(P, \overline{P})$ of the cut $(P, \overline{P})$ is*

$$k(P, \overline{P}) = \sum_{e \in (P, \overline{P})} k(e).$$

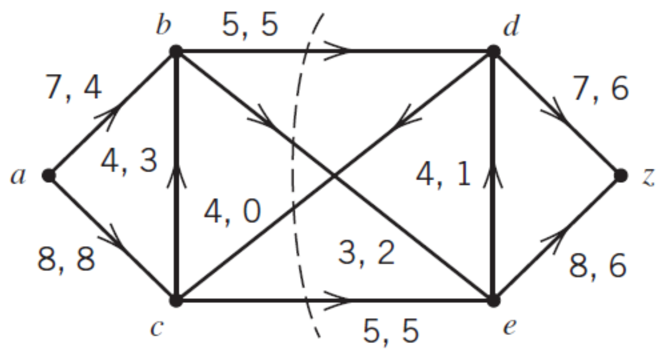
Find the capacity of the cut displayed in Example 8.4.

Theorem 8.1. *For any $a - z$ flow $f(e)$ in a network N, the flow out of a equals the flow into z.*

Proof Idea.

$\square$

Example 8.5. *What is the maximum value of any flow in the network below?*



Remark 8.1. *All flow from a to z must cross each $a - z$ cut. Thus, the combined capacity of the edges in any $a - z$ cut is an ____________________________ on how much flow can get from a to z.*

Theorem 8.2. *For any $a - z$ flow f and any $a - z$ cut $(P, \overline{P})$ in a network N, $|f| \leq k(P, \overline{P})$.*

Proof Idea.

$\square$

Remark 8.2. *The* **conservation of flow** *at a vertex (condition 2. from Definition 8.1) can be extended to any set of vertices P not containing a or z:*

$$\sum_{e\in(\overline{P},P)} f(e) = \sum_{e\in(P,\overline{P})} f(e).$$

That is, the flow into P equals the flow out of P.

Corollary 8.1. *For any $a-z$ flow f and any $a-z$ cut $(P,\overline{P})$ in a network N, $|f| = k(P,\overline{P})$ if and only if*

1. *For each $e \in (\overline{P},P), f(e) = 0$.*
2. *For each $e \in (P,\overline{P}), f(e) = k(e)$.*

Further, when $|f| = k(P,\overline{P})$, f is a maximum flow and $(P,\overline{P})$ is an $a-z$ cut of minimum capacity.

Proof.

□

Flow Maximizing Algorithm Attempt # 1

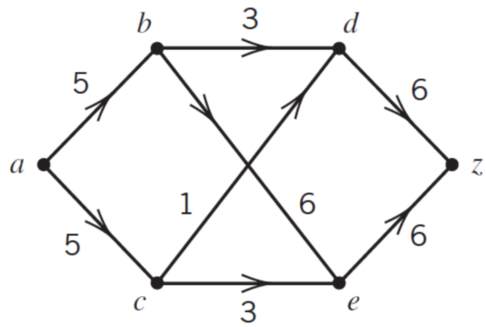
Definition 8.4. *An $a-z$* **unit flow** *f_L along an $a-z$ path L is defined as $f_L(e) = 1$ if e is in L and 0 if e is not in L. We call an edge* **unsaturated** *whenever the present flow does not equal the capacity. The* **slack** *$s(e)$ of an edge e in flow f is $s(e) = k(e) - f(e)$.*

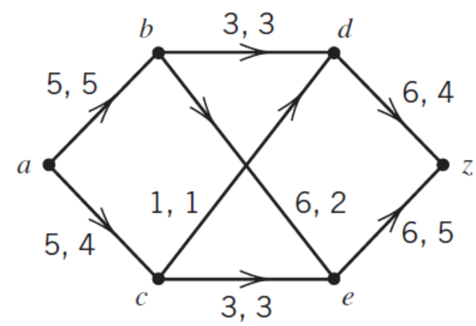
Remark 8.3. *All normal flows can be decomposed into a sum of $a-z$ unit flows. Then to build an $a-z$ flow, we can build up the flow as much as possible by successively adding $a-z$ flows together while being sure not to exceed any edge's capacity.*

Strategy: If s is the minimum slack among the edges in the $a-z$ unit flow f_L, we can add:

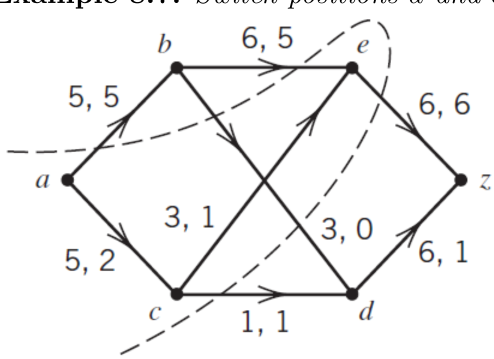
If $f_1, f_2, \ldots, f_m$ are $a-z$ unit flows, then:

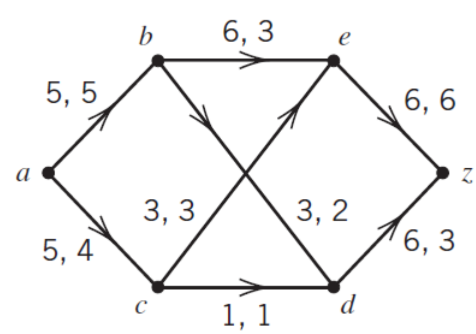
Example 8.6. *The capacities are shown for each arc in the network. Find a maximum flow.*





Example 8.7. *Switch positions d and e, and let's try the same strategy this same network.*





Augmenting Flow Algorithm

1. Give the vertex a the label $(-, \infty)$.

2. Call the vertex being scanned p with second label $\Delta(p)$. Initially, $p = a$.

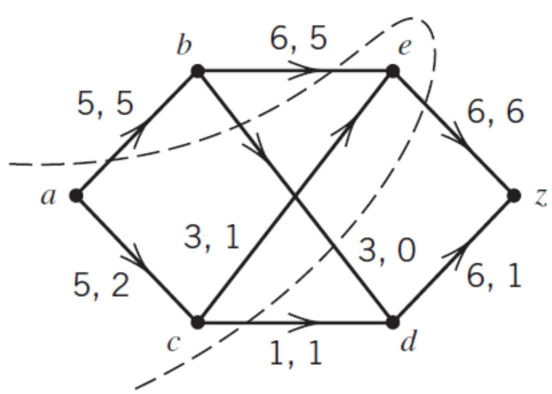
   (a) Check each incoming edge $e = (q, p)$. If $f(e) > 0$ and q is unlabeled, then label q with $[p, \Delta(q)]$, where $\Delta(q) = \min[\Delta(p), f(e)]$.

   (b) Check each outgoing edge $e = (p, q)$. If $s(e) = k(e) - f(e) > 0$ and q is unlabeled, then label q with $[p^+, \Delta(q)]$, where $\Delta(q) = \min[\Delta(p), s(e)]$.

3. If z has been labeled, go to Step 4. Otherwise choose another labeled vertex to be scanned (which was not previously scanned) and go to Step 2. If there are no more labeled vertices to scan, let P be the set of labeled vertices, and now $(P, \overline{P})$ is a saturated $a - z$ cut. Moreover, $|f| = k(P, \overline{P})$, and thus f is maximum.

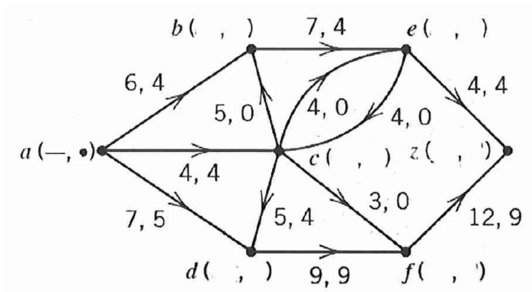
4. Find an $a - z$ chain K of slack edges by backtracking from z as in the shortest path algorithm. Then an $a - z$ flow chain f_K along K of $\Delta(z)$ units is the desired augmenting flow. Increase the flow in the edges of K by $\Delta(z)$ units (decrease flow if edge is backward directed in K).

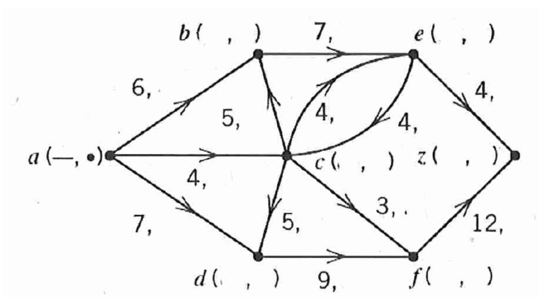
Example 8.8. *Apply the augmenting flow algorithm to the previous example.*

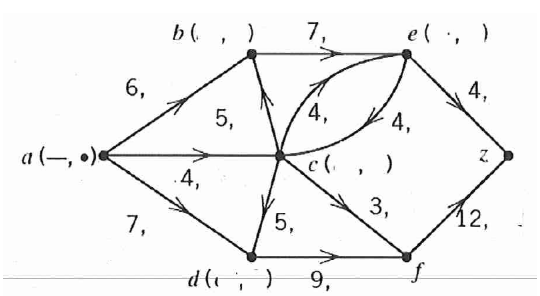


Remark 8.4. *If the algorithm is executed by a computer, it start with zero flow. When we implement the algorithm by hand, start with a flow by inspection to speed up the process.*

Example 8.9. *Apply the augmenting flow algorithm to the figure below given the initial flow. Find a maximum $a-z$ flow and a minimum capacity $a-z$ cut.*





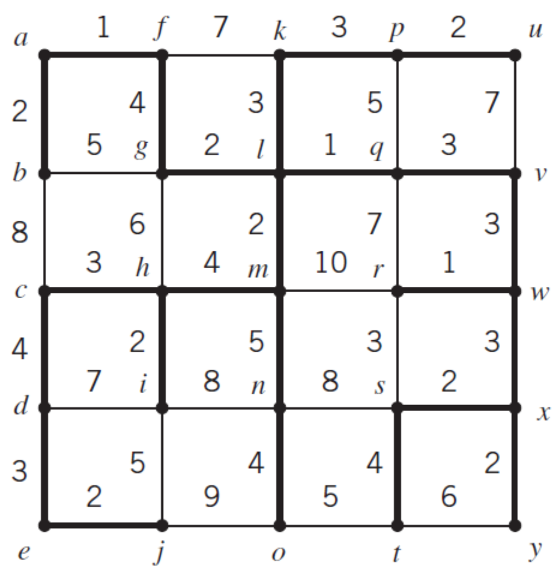


Theorem 8.3. *For any $a - z$ flow f, a finite number of applications of the augmenting flow algorithm yields a maximum flow. Moreover, if P is the set of vertices labeled during the final (unsuccessful) application of the algorithm, then $(P, \overline{P})$ is a minimum $a - z$ cut.*

Corollary 8.2 (Max Flow - Min Cut Theorem)**.** *In any directed flow network, the value of a maximum $a - z$ flow is equal to the capacity of a minimum $a - z$ cut.*

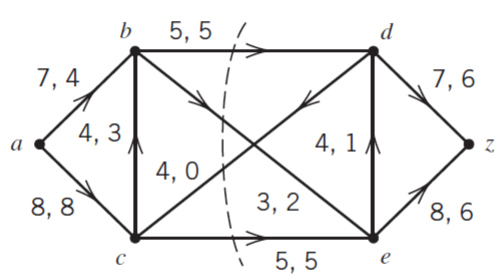
8.3 ICE 9: Network Algorithms

1. The network below shows the time it takes to walk along blocks in a city. The number on the edge uv is the time (number of minutes) it takes to walk from intersection u to intersection v.

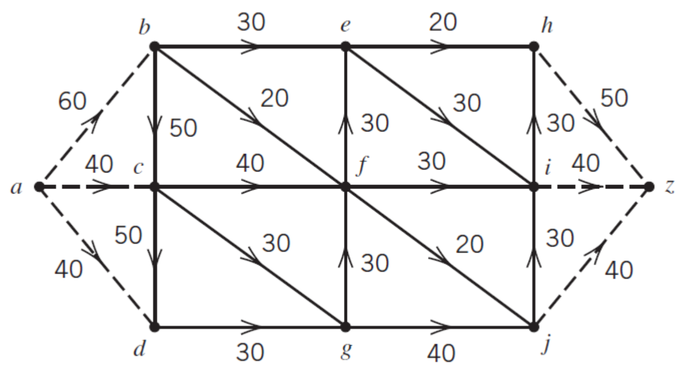


Use the shortest path algorithm to find the quickest path between intersections a and y. How long will it take to walk?

2. Apply the augmenting flow algorithm to the flow in the figure below.



3. Consider the network of solid edges below. The vertices b, c, and d can supply up to 60, 40, and 40 units of flow, respectively. Vertices h, i, and j have flow demands of 50, 40, and 40 units, respetively. The sources could be factories and sinks warehouses. Or perhaps the sources are oil refineries and the sinks oil-truck distribution centers.



(a) Can we meet all the demands? Find such a flow upon inspection, if possible.

(b) Now suppose the vertices b, c, d have unlimited supplies. How much flow can be sent to the set $\{h, i, j\}$? Explain your new model.

9 Combinatorics

9.1 Basic Counting Principles

Combinatorics is a branch of mathematics dealing with counting different outcomes of a some task called an activity with an observable outcome (called an experiment).

Principle 9.1 (Addition Principle)**.** *If there are r_1 different objects in the first set, r_2 different objects in the second set, ..., and r_m different objects in the m^{th} set, and if the different sets are **disjoint**, then the number of ways to select an object from one of the m sets is $r_1 + r_2 + ... + r_m$*

Principle 9.2 (Multiplication Principle)**.** *If a procedure can be broken into m successive (ordered) stages, with n_1 different outcomes in the first stage, n_2 different outcomes in the second stage, ..., and n_m different outcomes in the m^{th} stage. If the number of outcomes at each stage is **independent of the choices** in previous stages and if the composite outcomes are all distinct, then the total procedure has $n_1 \times n_2 \times n_3 \times ... \times n_m$ different composite outcomes.*

Principle 9.3 (Subtraction Principle)**.** *If $S \subset U$, then $S^c := \overline{S} = U\ \ S = \{x \in U : x \notin S\}$ is called the **complement** of S in U. The number of objects in S, $|S| = |U| - |\overline{S}|$*

Principle 9.4 (Division Principle)**.** *Let S be a finite set partitioned into k parts such that each part contains the same number of objects, say n, then the number of parts in the partition is $k = \dfrac{|S|}{n}$*

Example 9.1. *What are the total possible outcomes when you flip a coin 5 times?*

Example 9.2. *Eva is deciding on what canned wet food she will have for dinner. The box has 3 different flavors of fish and 2 different flavors which are chicken based. How many choices does she have?*

Example 9.3. *Baby Soraya has 10 baby bodysuits, 9 pairs of baby pants, 3 jackets, and 6 baby headbands. How many different outfit combinations does Soraya have?*

Example 9.4. *Suppose you roll two dice, one red and one green.*
a) How many different outcomes are there?

b) How many outcomes are there in which there are no doubles?

c) What is the probability that there are no doubles rolled?

Example 9.5. *A soccer team is numbering their jerseys by ironing on 1's, 2,'s, 6,'s, 7's, and 9's. How many 2-digit numbers can be made if*

1. *repetition is allowed?*

2. *repetition is NOT allowed?*

3. *Only odd 2-digit numbers can be used and repetition is NOT allowed?*

4. *In which the number 9 must be used and repetition is NOT allowed?*

Example 9.6. *A computer password needs to consist of a string of 6 symbols taken from the digits* $0, 1, 2, ..., 9$ *and lowercase letters* $a, b, c, ..., z$*. How many computer passwords have a repeated symbol?*

Example 9.7. *Suppose 11 people consisting of 5 men, 2 women, and 4 kids are going to a Lewis Volleyball game and have 11 seats in a row.*

a) How many ways there to seat them if there are no restrictions?

b) If the grown-ups sit together and the kids sit together?

Example 9.8. *Given 10 different English books, six different French books, and four different German books, (a) How many ways are there to select one book?*

(b) How many ways are there to select three books, one of each language?

(c) How many ways are there to make a row of three books in which exactly one language is missing (the order of the three books makes a difference)?

.

9.1.1 Ice -Counting Principles

2 sided!

1. How many 5 digit codes are there if

   (a) repetition is not allowed?

   (b) repetition is allowed?

   (c) the code number must start with a 2 and end with a 9 and repetition is not allowed?

   (d) The code number must be odd but cannot have any 5's in it and repetition is not allowed?

2. How many numbers of any length less than 500 can be formed from 2, 4, 7, and 9 if repetition of digits is allowed?

3. Suppose we have 3 different colored chips, Red, Blue, and White.

   (a) How many ways are there to grab 1 chip?

   (b) How many ways are there to grab one chip and then a second chip without replacement?

   (c) How many ways are there to grab a chip, then a second, then a third without replacement?

   (d) How many ways are there to grab 2 of the 3 chips at once?

   (e) How many ways are there to grab all 3 chips at once?

   (f) How many ways are there to put the chips in a circle?

9.1.2 Ice -Does Order Matter

For the following situations, determine where order matters.

1. Lewis has 3 soccer scholarships of $1,000. How many ways are there to award these after they've narrowed the candidates down to 6 athletes?

2. Lewis has 3 soccer scholarships of $1,000, $2,000, and $3,000. How many ways are there to award these after they've narrowed the candidates down to 6 athletes?

3. How many ways are there for 10 soccer teams to end the season in first, second, or third place?

4. After being dealt 5 cards, how many ways can someone arrange the cards in their hands?

5. How many different groups of 4 can be formed from 6 cats?

6. How many different 5-card poker hands are there?

7. How many different 9 player batting line ups are possible on a softball team with 13 players on the roster?

8. How many different outcomes are there when you flip 3 identical nickels at the same time?

9. How many different outcomes are there when you flip a penny 3 times in a row?

10. How many different 2 card blackjack hands are there (first card dealt down and second dealt face up)?

11. How many ways are there to grab 3 marbles one at a time out of a bag with 6 red marbles, 5 green marbles, and 2 blue marbles?

12. How many ways are there to grab 3 marbles one at a time out of a bag with 6 red marbles, 5 green marbles, and 2 blue marbles so that the first is red, the second is green, and the third is blue?

13. How many different 3 piece ski outfits could you come up with if you have 2 jackets, 3 hats, and 4 pairs of gloves?

14. How many different words can be formed using all of the letters in BIKE? (nonsense words are ok)?

9.2 Permutations and Combinations

Definition 9.1. *A permutation of n distinct objects is an arrangement, or ordering, of the n objects. An r-permutation of n distinct objects is an arrangement using r of the n objects, denoted* $P(n, r) =$

If $r > n$, $P(n, r) =$

$P(n, 1) =$

Example 9.9. *What is $P(n, n)$?*

Example 9.10. *If there are 9 cats in a cutest cat competition, how many ways are there for them to finish in 1st, 2nd, and third? (What if Archer is guaranteed to with first?)*

Example 9.11. *What is the probability that a 4-digit campus telephone number has one or more repeated digits?*

Example 9.12. *Suppose Archer has 3 kitty friends over to play poker. How many ways are there for the 4 to sit at Archer's round poker table?*

Definition 9.2 (Circular Permutations)**.** *To place n objects in a circle, there are* __________ *number of ways.*

Definition 9.3. *An r-combination of n distinct objects is an unordered selection, or subset, of r out of the n objects, denoted $C(n, r)$ and we say, "n choose r."*

We can derive the formula for $C(n, r)$:

Example 9.13. *From a standard deck of playing cards. How many 10-card hands are there?*

Example 9.14. *Suppose 11 people consisting of 5 men, 2 women, and 4 kids only have 6 tickets for a Blackhawks game. How many ways are there to choose who gets to go to the game if*

a) there are no restrictions?

b) 2 men, 1 women, and 3 kids get to go?

c) exactly 3 men get to go?

d) at least 2 kids get to go?

Theorem 9.1 (Permutations of Objects with some being identical). *If there are n objects r_1 of type 1, r_2 of type 2, ... , and r_m of type m, where $r_1 + r_2 + ... + r_m = n$, then the number of arrangements of these n objects, denoted $P(n; r_1, r_2, ..., r_m)$, is $C(n, r_1) \cdot C(n - r_1, r_2) \cdot C(n - r_1 - r_2, r_3) \cdot ...C(n - r_1 - ...r_{m-1}, r_m) = \dfrac{n!}{r_1! \cdot r_2! \cdot ...r_m!}$*

Example 9.15. *How many different ways are there to select 6 cats from 3 different breads of cats (say Siamese, the Archer line, and the Eva Line)?*

Theorem 9.2. *The number of selections with repetition of r objects chosen from n types of objects is $C(r + n - 1, r)$.*

Example 9.16. *How many ways are there to fill a box with a dozen doughnuts chosen from five different varieties with the requirement that at least one doughnut of each variety is picked?*

Example 9.17. *How many different full house hands are there in a standard deck of cards?*

.

9.2.1 Ice -Permutations and Combinations

1. Suppose you have a kickball team which has 12 kids, 7 boys and 5 girls.

   (a) Suppose you have chosen 3 kids to rest while the others are playing. How many ways are there to assign your remaining 9 players to either play the outfield (3 positions) or the infield (6) where their actual positions are not important only infield or outfield?

   (b) Suppose you have chosen 3 kids to play outfield (left, center, and right). How many ways can you assign these 3 selected for these outfield positions?

   (c) Suppose you have chosen 9 kids to play, 5 of which are girls. How many ways can you assign the 9 players to the infield if exactly 3 of the boys must play infield?

   (d) How many different 9 person batting line ups are there?

   (e) How many different 9 person line ups are there if it must alternate first a girl, then a boy, then a girl, etc.?

turn over!

2. From a group of 5 Maine Coon cats, 4 Siamese cats, and 3 Ragdoll cats, Eva needs to make a committee of 6 cats. In how many ways can this be done with:

   (a) there are no restrictions?

   (b) 3 Maine Coons, 2 Siamese, and 1 Ragdoll?

   (c) exactly 2 Ragdolls?

   (d) at least 4 are Maine Coons?

3. How many different 2 pair hands are there in a standard deck of cards?

4. How many ways are there to distribute six different books among 13 children if no child gets more than one book?
   (a) C(13, 6)
   (b) 13^6
   (c) C(13, 6)6!
   (d) None of the above.
   (e) Yay! Dr. Harsy is back!

9.3 Pigeon Hole Principle

The Pigeon Hole Principle (sometimes called Dirichlet drawer principle or the shoebox principle) basically says if a lot of pigeons fly into not as many pigeonholes, then at least one hole will have at least two pigeons.

Theorem 9.3 (Pigeon Hole Principle: Simple Form)**.** *If $n+1$ objects are distributed into n boxes, then at least one box contains two or more objects.*

Proof:

Example 9.18. *Among 13 people there are 2 who have their birthdays in the same month.*

Example 9.19. *Suppose we have n married couples. How many of the $2n$ people must be selected to guarantee that a married couple is selected?*

Abstract Algebra Tie-In:
The Pigeon Hole Principle is used to prove the Chinese Remainder Theorem:

Let m, n be relatively prime[9] positive integers and let $a, b \in \mathbb{Z}$ where $0 \leq a \leq m-1$ and $0 \leq b \leq n-1$. Then there is a positive $x \in \mathbb{Z}$ such that x can be written in the form $x = pm + a$ and $x = qn + b$ for some $p, q \in \mathbb{Z}$.

It's proof uses a similar method to the following example:

[9] Recall we say m, n are *relatively prime* if their greatest common divisor is 1

Example 9.20. *Given m integers $a_1, a_2, ..., a_m$ there exits $k, l \in \mathbb{Z}$ with $0 \leq k < l \leq m$ such that $a_{k+1} + a_{k+2} + ... + a_l$ is divisible by m. (That is, there exists consecutive a's in the sequence $a_1, a_2, ..., a_m$ whose sum is divisible by m.)*

Example: Let $m = 7$, and integers be 2,4,6,3,5,5, and 6.

9.4 ICE- More Counting and Pigeon Hole Principle

1. How many ways to select a subset of eight doughnuts from three types of doughnuts if at most two doughnuts of the first type can be chosen?

2. How many ways are there to have a collection of eight fruits from a large pile of identical oranges, apples, bananas, peaches, and pears if the collection should include exactly two different kinds of fruits?

3. Show that if $n + 1$ integers are chosen from the set $\{1, 2, ..., 2n\}$ then there are always two which differ by 1.

.

10 Generating Functions

10.1 Generating Function Models

Generating functions can be used to solve combinatorial problems with special constraints in selection and arrangement problems with repetition.

Suppose a_r is the number of ways to select r objects in a certain procedure. Then $g(x)$ is a ____________ ________ for a_r if

$$g(x) = a_0 + a_1x + a_2x^2 + ... + a_rx^r + ... + a_nx^n$$

If $g(x)$ has infinitely many terms, it is called a ________ ________.

Recall from Calculus II,

$$(x+1)^n = 1 + \binom{n}{1}x + \binom{n}{2}x^2 + ... + \binom{n}{r}x^r + ... + \binom{n}{n}x^n$$

Thus $g(x) = (x+1)^n$ is the generating function for $a_r = C(n,r)$, the number of ways to select r things from a set of n objects.

Example:
$(1+x)(1+x)(1+x) = 1 \cdot 1 \cdot 1 + 1 \cdot 1 \cdot x + 1x1 + 1xx + +x11 + x1x + xx1 + xxx$

To find the coefficient of x^r is the number of different products with r x's and $(3-r)$ 1's.
Example: Find x^2's coefficient:

Note we could replace 1 with x^0.

This is great when we are only counting one thing. But for this section, we will mostly be discussing multiplying polynomial factors... fun stuff right?

Note that $(1+x)(1+x)(1+x) = 111 + 11x + 1x1 + 1xx + +x11 + x1x + xx1 + xxx$
could be written as $x^{e_1}x^{e_2}x^{e_3}$ where $0 \leq e_i \leq 1$.

And $(1+x+x^2+x^3)^3$ could be written as $x^{e_1}x^{e_2}x^{e_3}$ where $0 \leq e_i \leq 3$.

Example: The coefficient of x^5 in $(1+x+x^2)^4$ is the number of formal products $x^{e_1}x^{e_2}x^{e_3}x^{e_4}$ where $0 \le e_i \le 2$ equaling x^5 and this is the same as the number of integer solutions of

This is equivalent to asking how many ways to select 5 objects from a collection of 4 types with at most 2 objects in each type (or distributing 5 identical objects into 4 distinct boxes with at most 3 objects in each box).

Example 10.1. *Find the generating function for the number of ways to select r balls from 3 green, 3 white, 3 blue, and 3 gold balls.*

Example 10.2. *Find a generating function for the number of ways to select r cats from 5 Siamese, 5 Maine Coons, 3 Tabbies, and 3 Calicos.*

Example 10.3. *Find a generating function for the number of ways for Soraya to select r outfits from 5 onesies, 4 pants, 4 jackets, 3 pairs of socks, and 3 headbands. Note this has the implied constraint that there must be at least one of each type.*

Example 10.4. *Find a generating function for a_r , the number of ways to distribute r identical objects into five distinct boxes with an even number of objects not exceeding 10 in the first two boxes and between three and five in the other boxes.*

Note if there isn't an upper bound, we have a power series.

.

10.1.1 ICE -Generating Function Models

1. Build a generating function for a_r the number of r selections from

   (a) 5 red, 5 black, and 4 white balls.

   (b) Five jelly beans, five licorice sticks, eight lollipops with at least one of each type of candy.

   (c) Six types of light bulbs with an odd number of the first and second types. Hint: There is no upper bound for the number of light bulbs.

2. Use a generating function for modeling the number of distributions of 16 chocolate bunny rabbits into four Easter baskets with at least three rabbits in each basket. Which coefficient do we want?

3. Find a generating function for a_k , the number of k-combinations of n types of objects with an even number of the first type, an odd number of the second type, and any amount of the other types.

4. Find a generating function for the number of integers between 0 and 999,999 whose sum of digits is r. Hint: 6 digits total with values between 0 and 9.

10.2 Calculating Coefficients of Generating Functions

Recall we can find coefficients of products of polynomials using __________ __________

That is long and tedious so instead we are going to use algebra to rewrite products in a way so that we can use Table 10.2 to determine coefficients of generating functions.

Step 1: Use algebra to rewrite your function to one of (or a product of) the following functions: $(1+x)^n, (1-x^m)^n$, or $(1-x)^{-n}$.

Step 2: Use Table 10.2 and polynomial multiplication to determine the desired coefficient.

(1) $\dfrac{1-x^{m+1}}{1-x} = 1 + x + x^2 + \cdots + x^m$

(2) $\dfrac{1}{1-x} = 1 + x + x^2 + \cdots$

(3) $(1+x)^n = 1 + \binom{n}{1}x + \binom{n}{2}x^2 + \cdots + \binom{n}{r}x^r + \cdots + \binom{n}{n}x^n$

(4) $(1-x^m)^n = 1 - \binom{n}{1}x^m + \binom{n}{2}x^{2m} + \cdots + (-1)^k\binom{n}{k}x^{km} + \cdots + (-1)^n\binom{n}{n}x^{nm}$

(5) $\dfrac{1}{(1-x)^n} = 1 + \binom{1+n-1}{1}x + \binom{2+n-1}{2}x^2 + \cdots + \binom{r+n-1}{r}x^r + \cdots$

(6) If $h(x) = f(x)g(x)$, where $f(x) = a_0 + a_1x + a_2x^2 + \cdots$ and $g(x) = b_0 + b_1x + b_2x^2 + \cdots$, then

$$h(x) = a_0b_0 + (a_1b_0 + a_0b_1)x + (a_2b_0 + a_1b_1 + a_0b_2)x^2 + \cdots$$
$$+ (a_rb_0 + a_{r-1}b_1 + a_{r-2}b_2 + \cdots + a_0b_r)x^r + \cdots$$

Figure 16: Table from Tucker's *Applied Combinatorics*

Example 10.5. *Find the coefficient of x^{16} in $(x^2 + x^3 + x^4 + ...)^5$. What is the coefficient of x^r? What does $(x^2 + x^3 + x^4 + ...)^5$ represent combinatorially?*

Example 10.6. *Use generating functions to find the number of ways to collect \$15 from 20 distinct people if each of the first 19 people can give a dollar (or nothing) and the twentieth person can give either \$1 or \$5 (or nothing).*

Example 10.7. *How many ways are there to select 25 toys from seven types of toys with between two and six of each type?*

Example 10.8. *How many ways are there to distribute 25 identical balls into seven distinct boxes if the first box can have no more than 10 balls but any number can go into each of the other six boxes?*

10.2.1 Coefficients of Generating Functions

1. Find the Coefficient of x^{11} in

   (a) $x^2(1-x)^{-10}$

   (b) $\dfrac{x^2 - 3x}{(1-x)^4}$

   (c) $\dfrac{(1-x^2)^5}{(1-x)^5}$

2. Find the coefficient of x^{10} in $(1 + x + x^2 + x^3 + ...)^n$.

3. Use generating functions to find the number of ways to select 10 balls from a large pile of red, white, and blue balls if the selection has at least two balls of each color. State the generating function.

4. Use generating functions to find the number of ways to select 10 balls from a large pile of red, white, and blue balls if the selection has at most two red balls. State the generating function.

5. Find the coefficient of x^r in $(x^5 + x^6 + x^7 + ...)^8$.

6. Find the coefficient of x^{32} in $(x^3 + x^4 + x^5 + x^6 + x^7)^7$.