MAT 118

Mathematical Thinking

Time and Place: 12:40-2:00 Monday and Friday, Javits 111

Lecturer: Sorin Popescu

- Office: 4-119 Math Tower
- Office Hours:
  - Monday 4:00-5:00 in Math Tower 4-119
  - Tuesday 2:20-3:20 in Physics A-127
  - or by appointment
- Phone: (631) 632-8358
- email: `sorin@math.sunysb.edu`

Recitation 1: 9:25-10:20 Monday, SBU 231 (James Sugrim)
Recitation 2: 3:20-4:15 Wednesday, Physics P-117 (S. Sadalge)
Recitation 3: 9:25-10:20 Monday, SBU 226 (S. Sadalge)

TAs: James Sugrim and Sandesh Sadalge

- Office Hours: (Sandesh Sadalge) Wed 2:15pm-3:15pm, Th 4:00-5:00 in MLC, Physics A-127
- Office Hours: (James Sugrim) Mo 6:30pm-8:30pm in MLC, Physics A-127

Exam dates:

- midterm exam 1: Monday, October 7 (in class). **Information concerning Midterm #1** [**PDF**]
- midterm exam 2: Monday, November 11 (in class). **Information concerning Midterm #2** [**PDF**]
- final exam: Friday, December 20, 11:00-1:30, Dance Studio. **Information concerning the Final Exam**

See the Stony Brook academic calendars, at `http://ws.cc.sunysb.edu/registrar/acadcal.htm` for other important dates.

Login on **Blackboard** to access the **Discussion Board** for this class.
(To access it, once in Blackboard, click on the **Communication** tab and then click on **Discussion Board**.)

Math learning center: The Math Learning Center (MLC), located in Room A-125/7 in the Physics Building, is an important resource. It is staffed most days and some evenings by mathematics tutors (professors and advanced students); your lecturer and recitation instructor will hold at least one office hour there. For more information and a schedule, consult the MLC web site.

Students with disabilities: If you have a physical, psychological, medical, or learning disability that may impact on your ability to carry out assigned course work, you are strongly urged to contact the staff in the Disabled Student Services (DSS) office: Room 133 in the Humanities Building; 632-6748v/TDD. The DSS office will review your concerns and determine, with you, what accommodations are necessary and appropriate. All information and documentation of disability is confidential. Arrangements should be made early in the semester (before the first exam) so that your needs can be accommodated.

MAT 118

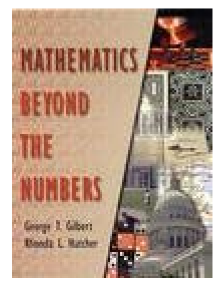
Mathematical Thinking

Syllabus | **Homework** | **Course policies**

The text for this course is

Mathematics Beyond the Numbers, by George T. Gilbert and Rhonda L. Hatcher, Wiley 2000

This introductory text includes a wide selection of interesting mathematical topics and real world applications. As prequisities an understanding of high school level mathematics is assumed. We will discuss as many of the following topics as time permits.



1. Voting Methods
   - Plurality and Runoff Methods
   - Borda's method: A Scoring System
   - Head-to-Head Comparisons
   - Aproval Voting
   - The Search for a Ideal Voting System
2. The mathematics of money
   - Powers, Roots and Logarithms
   - Simple Interest
   - Compound Interest
   - The Rewards of Systematic Savings
   - Amortized Loans
3. Probability
   - Elementary Probability
   - Odds
   - The Addition Rule
   - Counting Techniques
4. Paths and Networks
   - Eulerian Paths and Circuits on Graphs
   - The travelling salesman problem
5. Tilings and Polyhedra
   - Polygons
   - Regular Tilings
   - Polyhedra

You may find also useful the suite of Java applets (designed by John Lindsay Orr) available on the John Wiley & Sons website to suppport the textbook *Mathematics Beyond the Numbers*. Click here for a guide for using these java applets.

MAT 118

Mathematical Thinking

Syllabus | **Homework** | Course policies

Comment: Notice that odd problems have brief answers in the back of the book. Use these to check your work. Make sure you show and explain your work on all homework problems. Just writing the answer, as they do in the back of the book, is often nowhere near a complete solution.

HW#1 (Due 9/17 or 9/18 in Recitation) 1.1 # 2,3,12,13,15,21,22,25 and 1.2 # 1,9,12

HW#2 (Due 9/23 or 9/25 in Recitation) 1.3 # 2,10d,12,11d,13,14,18,19 and 1.4 # 5

HW#3 (Due 9/30 or 10/2 in Recitation) 1.4 #7, 1.5 # 3,4 and 3.1 # 1,2,7,9,11,13,21.
Extra Credit (worth 10 points): If I told you that the logarithm of my social security number is between 9 and 10, would you believe me?

HW#4 (Due 10/7 or 10/9 in Recitation) 3.2 #3, 7, 19, 23 and 3.3 #25, 27, 29, 32, 36, 41.
Extra Credit (worth 10 points): 3.3 #53

HW#5 (Due 10/14 or 10/16 in Recitation) 3.4 #3, 13, 15, 17, 22 and 3.5 #1, 5, 11, 14, 17

HW#6 (Due 10/21 or 10/23 in Recitation) 3.5 #24, 25, 35, 37

HW#7 (Due 10/28 or 10/30 in Recitation) 4.1 #1, 2, 9, 11, 13, 17, 19, 20, and 4.2 #1, 4, 9, 14
(In all problems on HW#7, you must explicitly describe sample spaces and events)
Extra Credit (worth 10 points): 4.1 #33

HW#8 (Due 11/4 or 11/6 in Recitation) 4.3 #8, 10, 4.4 #6, 8, 14, 15, 25, 4.5 # 1, 7, 31 and 4.6 #6, 10, 17

HW#9 (Due 11/11 or 11/13 in Recitation) 6.1 #2, 4, 6, 8, 9, 18, 21, 24, 27, 30

HW#10 (Due 11/18 or 11/20 in Recitation) 6.2 # 1, 2, 6, 7, 8, 9, 13, 15, 19

HW#* No homework due for 11/25 or 11/27 (Thanksgiving Break)

HW#11 (Due 12/2 or 12/4 in Recitation) 6.2 #20, 25, 6.3 #3, 4, 18, 21, 7.1 #26, and 7.2 #6, 8, 14

HW#12 (Due 12/9 or 12/11 in Recitation) 7.2 #11, 15, 20 and 7.3 #2, 13, 19, 24, 30, 32, 33

HW#11 and **HW#12** are for **extra credit**.

Sorin Popescu

2002-12-2

MAT 118

Mathematical Thinking

Homework: You can't learn mathematics without doing mathematics. Each week a Homework assignment will be posted on the web. It will be due in recitation the following week. You may work together on homework, but write up your own solutions in your own words. Late homework will **NOT** be accepted!

Quizzes: Most Fridays will begin with a short (10 minute) quiz, which should be straightforward for anyone who is keeping up with the homework and reading assignments.

Exams: We will have two midterm exams and a final exam. The dates are specified on the top level web page. No make-ups are allowed except if pre-arranged and in case of emergencies (medical, jury duty, etc)

Recitations: In addition to three hours of lectures each week, you will attend a one-hour recitation. The recitation instructor will help you with difficult homework problems, and will sometimes prepare exercises and activities that review past material and/or anticipate new material.

Grade: The various graded materials in this class form a total of 1000 points. Your grade will be determined solely based on your final point total. The minimal score needed to earn each grade is as follows:

900	840	780	730	680	630	580	540	500	460	0
A	A-	B+	B	B-	C+	C	C-	D+	D	F

Here is a breakdown of where the 1000 points come from:

7 quizzes (worth 20 points each)	140 points
10 homeworks (worth 25 points each)	250 points
Midterm I	180 points
Midterm II	180 points
Final exam	250 points

Extra Credit: Extra credit opportunities will periodically be posted on the course web page

Math 118 -- Useful information concerning Midterm #1

The test will be a 60-minute in-class exam held on Monday, October 7. You may not use any notes or books. Please come to class a few minutes early, so that you don't forfeit any of your 60 minutes.

Make up tests will be allowed only if pre-arranged. If you are sick, please email or call me in my office (632-8358) **BEFORE** the test.

The exam will cover all of Chapter 1 and most of Chapter 3.

The following is a partial list of concepts and "skills" you may need during the exam:

• Be able to explain how each of the following voting systems works: **Plurality**, **Runoff**, **Borda's Method** and **Approval Voting**. Be able to explain the advantages and disadvantages of each system.

• Be able to explain what a **Condorcet winner** means. Understand that not every election has a Condorcet winner!

. Given a table of **preference rankings**, be able to figure out which candidate would win each type of election, and which candidate (if any) is a Condorcet winner.

• Understand what it means for preference rankings to be **single-peaked,** with respect to a given ordering of the candidates. Given a table of preference rankings, you should be able to quickly determine whether they are single-peaked.

• Know **Duncan Black's Theorem**: "Any election in which the preference rankings are single-peaked (with respect to some ordering of the candidates) has a Condorcet winner."

• Be able to explain what **Pareto optimality** and **independence from irrelevant alternatives** mean. Know which of the voting systems we've studied (plurality, runoff, Borda) satisfies each of these properties. Give an example of a voting system that does not satisfy the Pareto optimality property.

• Know **Arrow's Impossibility Theorem**: Other than a dictatorship, no voting system based on preference rankings satisfies both the Pareto optimality and the independence from irrelevant alternatives properties.

• Know basic properties of **logarithms**. Use logarithms to solve equations like $5^t=20$, etc.

• Understand that **simple interest** means that interest is paid only on the amount originally deposited, which is called the **principal**, or **present value**. With simple interest, the **future value** is calculated by the formula $F=P(1+rt)$.

• Understand that in a **compound interest** account, interest is compounded at regular intervals, and is paid on the principal as well as on all previous interests payments credited to the account. If the annual interest rate is r and interest is compounded n times per year, then the **periodic interest rate** is r/n, and the future value of the account is given by the formula $F=P(1+r/n)^{nt}$ where t is the number of years that you wait.

• Know that the **APY** (Annual Percentage Yield) of an account is the actual percentage by which your balance increases each year, which depends on the annual interest rate and on the frequency of compounding. You may compare two investments by comparing their APYs. Know and understand the formula APY =

n

(1+r/n) -1.

• You do not need to memorize the **Systematic Savings Plan Formula** (pg 155), although if you are given the formula on the exam, you should know what each of the letters stands for (F,D,r,n,t), and you should be able to use the formula to solve problems similar to your homework problems. Also, understand that the formula tells you what answer you would get if you added up the first deposit plus the interest it earns plus the second deposit plus the interest it earns, etc.

• Know that an **amortized loan** is a loan that is paid back with equal payments at regular intervals. The lender uses the **Loan Formula** (pg 164) to calculate the dollar amount of each payment. You do not need to memorize the loan formula, but if you are given the formula on the exam, you should know what each of the letters stands for (F,R,r,n,t), and you should be able to use the formula to solve problems similar to the homework problems.

• Understand that the loan formula contrives the amount of each payment so that, if the lender were to deposit the payment checks as he received them, the amount in this account when the loan is finally paid off (as calculated by the systematic savings plan formula) equals the amount he could have made by investing the entire amount of the loan in the first place.

• Know that the **loan balance** of an amortized loan is the amount that one would need in order to pay off the entire loan today. Said differently, the loan balance is the principal that one would have to borrow today in order to yield the series of payments remaining to be paid.

Math 118 −− Useful information concerning Midterm #1

The test will be a 60−minute in−class exam held on Monday, October 7. You may not use any notes or books. Please come to class a few minutes early, so that you don't forfeit any of your 60 minutes.

Make up tests will be allowed only if pre−arranged. If you are sick, please email or call me in my office (632−8358) **BEFORE** the test.

The exam will cover all of Chapter 1 and most of Chapter 3.

The following is a partial list of concepts and "skills" you may need during the exam:

· Be able to explain how each of the following voting systems works: **Plurality**, **Runoff**, **Borda's Method** and **Approval Voting**. Be able to explain the advantages and disadvantages of each system.

· Be able to explain what a **Condorcet winner** means. Understand that not every election has a Condorcet winner!

. Given a table of **preference rankings**, be able to figure out which candidate would win each type of election, and which candidate (if any) is a Condorcet winner.

· Understand what it means for preference rankings to be **single−peaked,** with respect to a given ordering of the candidates. Given a table of preference rankings, you should be able to quickly determine whether they are single−peaked.

· Know **Duncan Black's Theorem**: "Any election in which the preference rankings are single−peaked (with respect to some ordering of the candidates) has a Condorcet winner."

· Be able to explain what **Pareto optimality** and **independence from irrelevant alternatives** mean. Know which of the voting systems we've studied (plurality, runoff, Borda) satisfies each of these properties. Give an example of a voting system that does not satisfy the Pareto optimality property.

· Know **Arrow's Impossibility Theorem**: Other than a dictatorship, no voting system based on preference rankings satisfies both the Pareto optimality and the independence from irrelevant alternatives properties.

· Know basic properties of **logarithms**. Use logarithms to solve equations like $5^t=20$, etc.

· Understand that **simple interest** means that interest is paid only on the amount originally deposited, which is called the **principal**, or **present value**. With simple interest, the **future value** is calculated by the formula F=P(1+rt).

· Understand that in a **compound interest** account, interest is compounded at regular intervals, and is paid on the principal as well as on all previous interests payments credited to the account. If the annual interest rate is *r* and interest is compounded n times per year, then the **periodic interest rate** is *r/n*, and the future value of the account is given by the formula $F=P(1+r/n)^{nt}$ where t is the number of years that you wait.

· Know that the **APY** (Annual Percentage Yield) of an account is the actual percentage by which your balance increases each year, which depends on the annual interest rate and on the frequency of compounding. You may compare two investments by comparing their APYs. Know and understand the formula APY = $(1+r/n)^n-1$.

· You do not need to memorize the **Systematic Savings Plan Formula** (pg 155), although if you are given the formula on the exam, you should know what each of the letters stands for (F,D,r,n,t), and you should be able to use the formula to solve problems similar to your homework problems. Also, understand that the formula tells you what answer you would get if you added up the first deposit plus the interest it earns plus the second deposit plus the interest it earns, etc.

· Know that an **amortized loan** is a loan that is paid back with equal payments at regular intervals. The lender uses the **Loan Formula** (pg 164) to calculate the dollar amount of each payment. You do not need to memorize the loan formula, but if you are given the formula on the exam, you should know what each of the letters stands for (F,R,r,n,t), and you should be able to use the formula to solve problems similar to the homework problems.

· Understand that the loan formula contrives the amount of each payment so that, if the lender were to deposit the payment checks as he received them, the amount in this account when the loan is finally paid off (as calculated by the systematic savings plan formula) equals the amount he could have made by investing the entire amount of the loan in the first place.

· Know that the **loan balance** of an amortized loan is the amount that one would need in order to pay off the entire loan today. Said differently, the loan balance is the principal that one would have to borrow today in order to yield the series of payments remaining to be paid.

Math 118 -- Useful information concerning Midterm #2

The test will be a 70-minute in-class exam held on Monday, November 11. You may not use any notes or books. You will need a calculator! Please come to class a few minutes early, so that you don't forfeit any of your 60 minutes.

Make up tests will be allowed only if pre-arranged. If you are ill, please email or call me in my office (632-8358) **BEFORE** the test.

The exam will cover sections 1,2(no house odds and fair bets),3,4,5 of chapter 4 and sections 1,2 (only till the Greedy Algorithm) of chapter 6.

The following is a partial list of concepts and "skills" you may need during the exam:

- A "**sample space**" (denoted **S**) is a list of all possible **outcomes** of an **experiment**. The favorable "**events**" (denoted **E**) form a subset of the sample space. For example, in the experiment of rolling a die, the sample space is **S**={1,2,3,4,5,6}, and the collection of favorable events that the die rolls an even number is **E**= {2,4,6}.

- The most important rule of probability is this: if all outcomes in a sample space are equally likely, then **P(E) = n(E)/n(S)**. Continuing the previous example, the probability of rolling an even number is 3/6.

- The probability that something does not happen is 1 – the probability that it does happen. For example, since there is a 1/6 chance that a die will roll "3", there must be a 5/6 chance that the die will roll something other than a "3". If you are having trouble calculating the probability that something happens, ask yourself whether it is easier to calculate the probability that the opposite thing happens!

- Two events are called **mutually exclusive** if they cannot both happen; in other words, if they are disjoint subsets of the sample space. For example, if I roll two dice, then the event that the dice sum to 6 and the even that the dice sum to 7 are mutually exclusive (they can't both happen at the same time).

- The **Addition Rule** states that, if A and B are events in a sample space, then **P(A or B) = P(A) + P(B) – P(A and B)**. Notice that when A and B are mutually exclusive, P(A and B) is zero, so we just have that **P(A or B) = P(A) + P(B)**.

- To find the chance that two **independent things** will both happen, you should **multiply** their chances. For example, what is the chance that a coin flips heads AND a die rolls 6? P(heads and 6) = P(heads).P(6) = (1/2).(1/6) = 1/12. In this example, the die roll and the coin flip are independent because they have nothing do with each other; knowing how the coin flip turns out doesn't affect the outcome of the die roll.

- Even when two things are dependent, you can still multiply their chances. The **multiplication rule**, which involves conditional probabilities, says that the chance that two things will both happen is the chance that the first thing will happen times the chance that the second thing will happen, given that the first happened. For example, if I take two cards from a deck, what's the chance they are both hearts? P(both are ♥) = P(first is ♥) . P(second is ♥ | first is ♥) = (13/52).(12/51) = 5.88%.

- In the last example, P(second is ♥ | first is ♥)= 12/51 is called a **conditional probability**, and means the chance that the second card is a heart given that the first card was a heart. More precisely, if A and B are any events in a sample space S, then the conditional probability is **P(B|A)=n(A and B)/n(A)**. Think of P(B|A) as telling you the chance that B is true given that you know A is true (so only the outcomes in A are relevant for figuring the probability, hence the formula). The events A and B are called **independent** if **P(B|A) = P(B)**, which says that “knowing A is true doesn’t change the odds at which you’re willing to bet that B is true”.

The precise version of the **multiplication rule** says that **P(A and B) = P(A) P(B|A)=P(B) P(A|B)**.

• The **Basic Counting Law** says that if there are M possible ways a first task can be performed and, after the first task is complete, there are N possible ways for a second task to be performed, then there are M.N possible ways for the two tasks to be performed in that order. For example, there are 52.51 = 2652 outcomes that the experiment of drawing two cards from a deck could turn out.

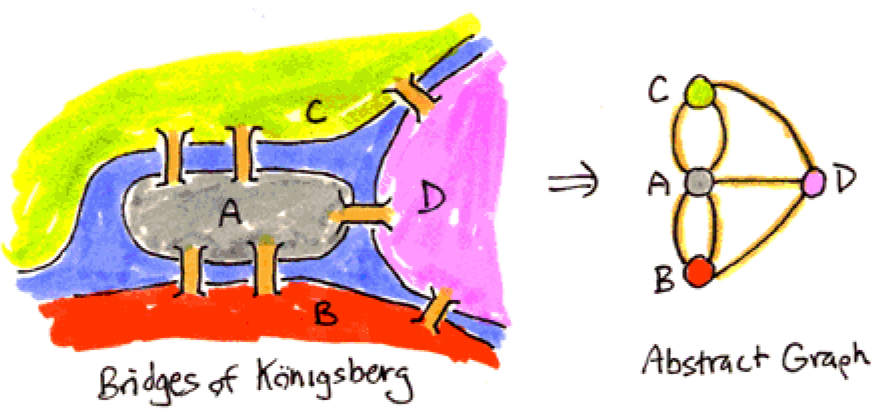
• **n!** denotes the product of all the numbers between 1 and n. There are **n!** possible ways to **permute** (rearrange) a list of n objects. For example, there are 5! = 120 words that can be spelled by rearranging the letters A,B,C,D,E.

• The number of **permutations of n objects taken k at a time** is denoted $_nP_k$, and is given by the formula $\mathbf{_nP_k = n!/(n-k)!}$ For example, $_5P_2 = 20$, which means that there are 20 different ways that you could form an <u>ordered</u> list of 2 of the 5 letters A,B,C,D,E. Can you list them all?

• The number of **combinations of n objects taken k at a time** is denoted $_nC_k$, or C_n^k, and is given by the formula $\mathbf{_nC_k = n!/k!(n-k)!}$ For example, $_5C_2 = 10$, which means that there are 10 different ways that you could form an <u>unordered</u> list of 2 of the 5 letters A,B,C,D,E. Can you list them all?

• In certain probability questions, when using the formula P(E)=n(E)/n(S), the sample space and favorable events are way too big to list. In these problems, it may still be a good idea to describe the sample space in words and begin to list the outcomes in it, as starting to list outcomes in the sample space may help you figure out how to use the combinations/permutation formulas above to calculate n(E) and n(S).

• An undirected **graph** G consists of a collection of **vertices** and of segments or arcs starting and ending at vertices which are called **edges**. An edge may connect two different edges or it may start and end at the same vertex; in this last case it is called a **loop**. Here is the graph G modeling Euler's problem whether or not it is possible to stroll around Königsberg in Prussia (nowadays Kaliningrad in Russia) crossing each of its bridges across the Pregel (nowadays called Pregolya) exactly once:

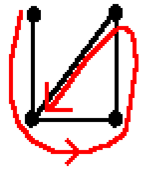


The above graph has four vertices (A,B,C,D). Vertices A and B are joined by two edges, vertices A and C are also joined by two edges, while A and D, B and D, C and D are each joined by only one edge. There are altogether 7 edges, each corresponding to a bridge over the Pregel.

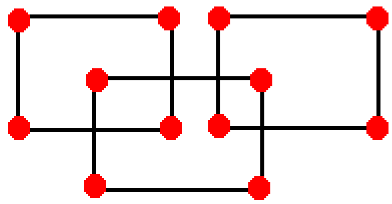
• The **degree** or **valency** of a vertex is the number of edges that end (or start) at that vertex (a loop at that vertex is counted twice). For instance the degree/valency of the vertex A in Euler's above graph is 5. The degree of vertex D is 3. A vertex whose degree is even is called **even**; a vertex whose degree is odd is called **odd**. For instance all vertices in Euler's graph are odd. Check this!

• Every graph has an even number of odd vertices!

• A **path** in a graph G is a finite sequence of edges in which any two consecutive edges are adjacent. For instance a path of length four:

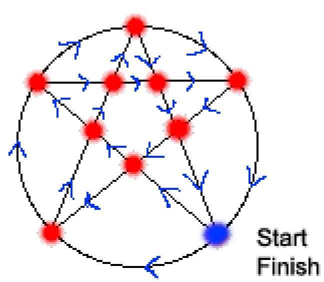


We say a path is a **circuit** if the initial vertex is also the finial vertex. A graph is **connected** if and only if there is a path between each pair of vertices. It is called **disconnected** otherwise. Is the Königsberg's bridges graph connected or not? The following graph is disconnected (it has three components):



• A path that runs exactly once over each edge of a graph is called an **Eulerian path**. An Eulerian path which is a circuit (that is it is closed) is called an **Eulerian circuit**. A graph that possesses an Eulerian circuit is called an **Eulerian graph** (named in honor of Euler's problem about Königsberg's bridges, which required to check if the bridges' graph is Eulerian or not).

• **Euler's theorem**: A graph has an Eulerian circuit if and only if it is connected and all its vertices are even. A graph has an Eulerian path if and only if it is connected and has either no odd vertices or exactly two odd vertices. If two of its vertices are odd, then any Eulerian path must begin at one these vertices and end at the other one. The Königsberg's bridges graph is not Eulerian since it has four odd vertices! Here is an example of an Eulerian circuit:



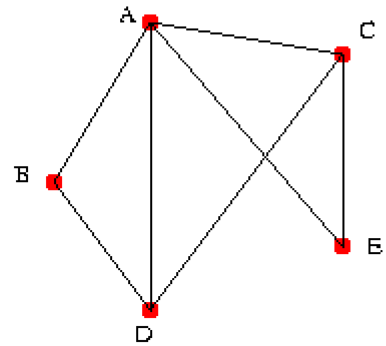
• A **bridge** in a graph is an edge whose removal disconnects the remaining graph. Eulerian graphs have no bridges! An **isolated vertex** in a graph is a a vertex with no edges connected to it.

• If G is an Eulerian (connected) graph, then the following algorithm (**Fleury's algorithm**) always produces an Eulerian circuit in G:
Start at any vertex of the graph and walk along the edges of the graph arbitrarily but subject to the following two rules:
1) Erase edges as they are traversed, and the isolated vertices resulted (if any)

2) Walk along a bridge only if there is no alternative.

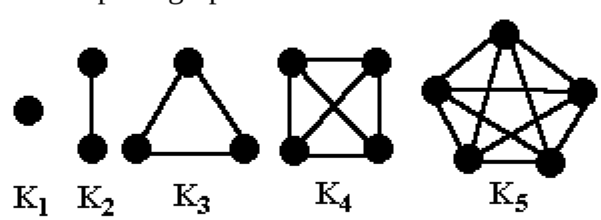
• If a graph does not have an Eulerian circuit or an Eulerian path, we might still be interested in knowing how it could be traveled with as few retraced edges as possible (starting and ending at the same vertex if interested in a circuit, or starting and ending at different vertices if just interested in a path). This process is called an **Eulerization** of the graph. Here are the guidelines mentioned in your textbook to achieve that:
- Circle all odd vertices and pair each of them with another odd vertex that is close to it
- For each pair of odd vertices, find a path with the fewest edges in the original graph connecting them and then duplicate all edges along this path.

• A connected graph G is said to be **Hamiltonian** if there exists a circuit (called a **Hamiltonian circuit**) passing through each vertex of G. Hamiltonian circuits, like Eulerian circuits, aim to traverse the "whole graph", but by whole they mean "visit each vertex once" (as opposed to Euler's "walk each edge once"). Here is one example of a Hamiltonian graph:

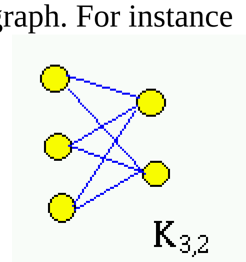


Can you find a Hamiltonian circuit in it ?

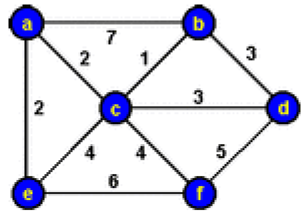
• A graph is called **simple** if there is at most one edge between any two of its vertices. A graph is called **complete** if there is exactly one edge between any two vertices. The complete graph with n vertices is denoted K_n. Here are the first five complete graphs:



A graph is called **complete bipartite**, and denoted $K_{n,m}$, if its vertices can be partitioned into two disjoint subsets A and B, one of size n and the other of size m, such that every vertex in A is connected to every vertex in B, and these form all edges of the graph. For instance



• A **weighted graph** is a graph for which one has associated a number (the **weight**) to each edge. For instance:



• The **Traveling salesman problem** is to find a Hamiltonian circuit in a complete weighted graph for which the sum of the weights of the edges is minimal. The sum of the weights of the edges of a circuit is called the **weight of the circuit**. More generally the Traveling salesman problem asks to find in a weighted (simple) graph a Hamiltonian circuit of least total weight.

• A **Brute Force Algorithm** approach to the Traveling salesman problem is: Check all circuits and select the one with the least total weight! In other words:
- Make a list of all possible Hamiltonian circuits of the graph
- For each Hamiltonian circuit, calculate its total weight by adding the weights of all the edges in the circuit.
- Find the circuit(s) with the least total weight.

Pros: Guaranteed to find the most (cost/weight)-efficient circuit! Cons: Enormous amount of work to carry out the algorithm!
(each increase in the number of vertices increases the work by a factor equal to the number of vertices in the graph)

• The **Nearest Neighbor Algorithm** is an approximate algorithm for the Traveling salesman problem in a complete weighted graph:
- Start at the vertex where the circuit is supposed to begin
- From the starting vertex, go to the vertex for which the corresponding edge has the smallest weight (= **nearest neighbor**).
- Build the circuit, one vertex at a time, by always going from a vertex to the nearest neighbor of that vertex from among the vertices that haven't been visited yet. If there is a tie choose any of the nearest vertices. Keep doing this until all the vertices have been visited.
- Once all the vertices have been visited, from the last vertex, return to the start.

Pros: Much less work then the brute force approach! Cons: It doesn't necessarily produce the optimal Hamilton circuit (so it is a compromise answer).

Math 118 −− Useful information concerning Midterm #2

The test will be a 70−minute in−class exam held on Monday, November 11. You may not use any notes or books. You will need a calculator! Please come to class a few minutes early, so that you don't forfeit any of your 60 minutes.

Make up tests will be allowed only if pre−arranged. If you are ill, please email or call me in my office (632−8358) **BEFORE** the test.

The exam will cover sections 1,2(no house odds and fair bets),3,4,5 of chapter 4 and sections 1,2 (only till the Greedy Algorithm) of chapter 6.

The following is a partial list of concepts and "skills" you may need during the exam:

· A "**sample space**" (denoted **S**) is a list of all possible **outcomes** of an **experiment**. The favorable "**events**" (denoted **E**) form a subset of the sample space. For example, in the experiment of rolling a die, the sample space is **S**={1,2,3,4,5,6}, and the collection of favorable events that the die rolls an even number is **E**={2,4,6}.

· The most important rule of probability is this: if all outcomes in a sample space are equally likely, then **P(E) = n(E)/n(S)**. Continuing the previous example, the probability of rolling an even number is 3/6.

· The probability that something does not happen is 1 – the probability that it does happen. For example, since there is a 1/6 chance that a die will roll "3", there must be a 5/6 chance that the die will roll something other than a "3". If you are having trouble calculating the probability that something happens, ask yourself whether it is easier to calculate the probability that the opposite thing happens!

· Two events are called **mutually exclusive** if they cannot both happen; in other words, if they are disjoint subsets of the sample space. For example, if I roll two dice, then the event that the dice sum to 6 and the even that the dice sum to 7 are mutually exclusive (they can't both happen at the same time).

· The **Addition Rule** states that, if A and B are events in a sample space, then **P(A or B) = P(A) + P(B) – P(A and B)**. Notice that when A and B are mutually exclusive, P(A and B) is zero, so we just have that **P(A or B) = P(A) + P(B)**.

· To find the chance that two **independent things** will both happen, you should **multiply** their chances. For example, what is the chance that a coin flips heads AND a die rolls 6? P(heads and 6) = P(heads).P(6) = (1/2).(1/6) = 1/12. In this example, the die roll and the coin flip are independent because they have nothing do with each other; knowing how the coin flip turns out doesn't affect the outcome of the die roll.

· Even when two things are dependent, you can still multiply their chances. The **multiplication rule**, which involves conditional probabilities, says that the chance that two things will both happen is the chance that the first thing will happen times the chance that the second thing will happen, given that the first happened. For example, if I take two cards from a deck, what's the chance they are both hearts? P(both are e) = P(first is e) . P(second is e | first is e) = (13/52).(12/51) = 5.88%.

· In the last example, P(second is e | first is e)= 12/51 is called a **conditional probability**, and means the chance that the second card is a heart given that the first card was a heart. More precisely, if A and B are any events in a sample space S, then the conditional probability is **P(B|A)=n(A and B)/n(A)**. Think of P(B|A) as telling you the chance that B is true given that you know A is true (so only the outcomes in A are relevant for figuring the probability, hence the formula). The events A and B are called **independent** if **P(B|A) = P(B)**, which says that "knowing A is true doesn't change the odds at which you're willing to bet that B is true".

The precise version of the **multiplication rule** says that **P(A and B) = P(A) P(B|A)=P(B) P(A|B)**.

· The **Basic Counting Law** says that if there are M possible ways a first task can be performed and, after the first task is complete, there are N possible ways for a second task to be performed, then there are M.N possible ways for the two tasks to be performed in that order. For example, there are 52.51 = 2652 outcomes that the experiment of drawing two cards from a deck could turn out.

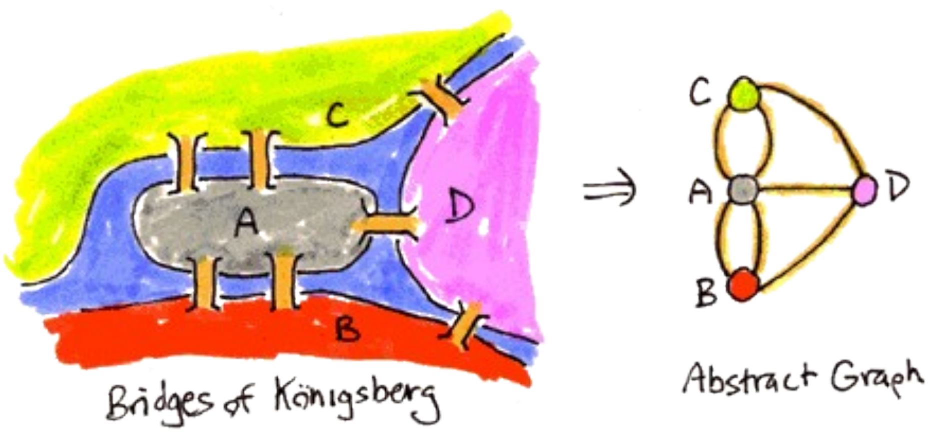
· **n!** denotes the product of all the numbers between 1 and n. There are **n!** possible ways to **permute** (rearrange) a list of n objects. For example, there are 5! = 120 words that can be spelled by rearranging the letters A,B,C,D,E.

· The number of **permutations of n objects taken k at a time** is denoted $_nP_k$, and is given by the formula $\mathbf{{}_nP_k = n!/(n-k)!}$ For example, $_5P_2$ = 20, which means that there are 20 different ways that you could form an ordered list of 2 of the 5 letters A,B,C,D,E. Can you list them all?

· The number of **combinations of n objects taken k at a time** is denoted $_nC_k$, or C_n^k, and is given by the formula $\mathbf{{}_nC_k = n!/k!(n-k)!}$ For example, $_5C_2$ = 10, which means that there are 10 different ways that you could form an unordered list of 2 of the 5 letters A,B,C,D,E. Can you list them all?

· In certain probability questions, when using the formula P(E)=n(E)/n(S), the sample space and favorable events are way too big to list. In these problems, it may still be a good idea to describe the sample space in words and begin to list the outcomes in it, as starting to list outcomes in the sample space may help you figure out how to use the combinations/permutation formulas above to calculate n(E) and n(S).

· An undirected **graph** G consists of a collection of **vertices** and of segments or arcs starting and ending at vertices which are called **edges**. An edge may connect two different edges or it may start and end at the same vertex; in this last case it is called a **loop**. Here is the graph G modeling Euler's problem whether or not it is possible to stroll around Königsberg in Prussia (nowadays Kaliningrad in Russia) crossing each of its bridges across the Pregel (nowadays called Pregolya) exactly once:

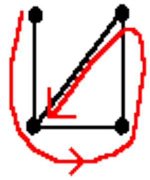


The above graph has four vertices (A,B,C,D). Vertices A and B are joined by two edges, vertices A and C are also joined by two edges, while A and D, B and D, C and D are each joined by only one edge. There are altogether 7 edges, each corresponding to a bridge over the Pregel.

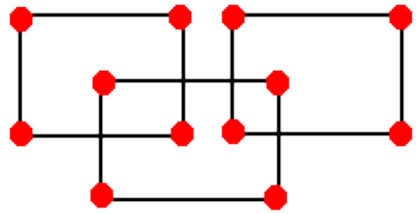
· The **degree** or **valency** of a vertex is the number of edges that end (or start) at that vertex (a loop at that vertex is counted twice). For instance the degree/valency of the vertex A in Euler's above graph is 5. The degree of vertex D is 3. A vertex whose degree is even is called **even**; a vertex whose degree is odd is called **odd**. For instance all vertices in Euler's graph are odd. Check this!

· Every graph has an even number of odd vertices!

· A **path** in a graph G is a finite sequence of edges in which any two consecutive edges are adjacent. For instance a path of length four:

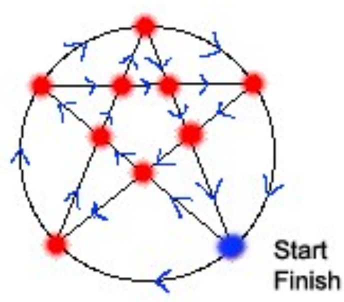


We say a path is a **circuit** if the initial vertex is also the finial vertex. A graph is **connected** if and only if there is a path between each pair of vertices. It is called **disconnected** otherwise. Is the Königsberg's bridges graph connected or not? The following graph is disconnected (it has three components):



· A path that runs exactly once over each edge of a graph is called an **Eulerian path**. An Eulerian path which is a circuit (that is it is closed) is called an **Eulerian circuit**. A graph that possesses an Eulerian circuit is called an **Eulerian graph** (named in honor of Euler's problem about Königsberg's bridges, which required to check if the bridges' graph is Eulerian or not).

· **Euler's theorem**: A graph has an Eulerian circuit if and only if it is connected and all its vertices are even. A graph has an Eulerian path if and only if it is connected and has either no odd vertices or exactly two odd vertices. If two of its vertices are odd, then any Eulerian path must begin at one these vertices and end at the other one. The Königsberg's bridges graph is not Eulerian since it has four odd vertices! Here is an example of an Eulerian circuit:



· A **bridge** in a graph is an edge whose removal disconnects the remaining graph. Eulerian graphs have no bridges! An **isolated vertex** in a graph is a a vertex with no edges connected to it.

· If G is an Eulerian (connected) graph, then the following algorithm (**Fleury's algorithm**) always produces an Eulerian circuit in G:
Start at any vertex of the graph and walk along the edges of the graph arbitrarily but subject to the following two rules:

1) Erase edges as they are traversed, and the isolated vertices resulted (if any)

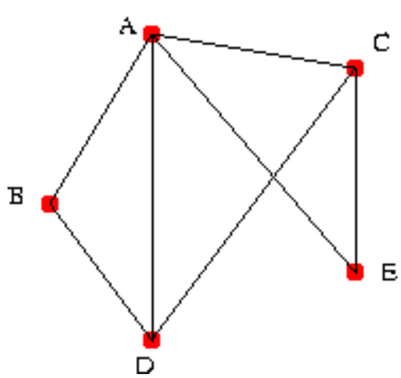
2) Walk along a bridge only if there is no alternative.

· If a graph does not have an Eulerian circuit or an Eulerian path, we might still be interested in knowing how it could be traveled with as few retraced edges as possible (starting and ending at the same vertex if interested in a circuit, or starting and ending at different vertices if just interested in a path). This process is called an **Eulerization** of the graph. Here are the guidelines mentioned in your textbook to achieve that:

− Circle all odd vertices and pair each of them with another odd vertex that is close to it

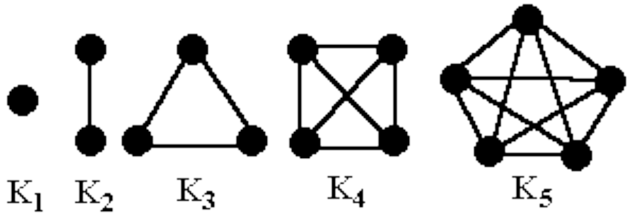
− For each pair of odd vertices, find a path with the fewest edges in the original graph connecting them and then duplicate all edges along this path.

· A connected graph G is said to be **Hamiltonian** if there exists a circuit (called a **Hamiltonian circuit**) passing through each vertex of G. Hamiltonian circuits, like Eulerian circuits, aim to traverse the "whole graph", but by whole they mean "visit each vertex once" (as opposed to Euler's "walk each edge once"). Here is one example of a Hamiltonian graph:

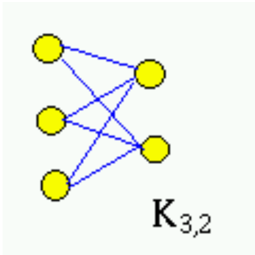


Can you find a Hamiltonian circuit in it ?

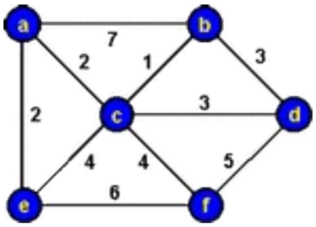
· A graph is called **simple** if there is at most one edge between any two of its vertices. A graph is called **complete** if there is exactly one edge between any two vertices. The complete graph with n vertices is denoted K_n. Here are the first five complete graphs:



A graph is called **complete bipartite**, and denoted $K_{n,m}$, if its vertices can be partitioned into two disjoint subsets A and B, one of size n and the other of size m, such that every vertex in A is connected to every vertex in B, and these form all edges of the graph. For instance



· A **weighted graph** is a graph for which one has associated a number (the **weight**) to each edge. For instance:



· The **Traveling salesman problem** is to find a Hamiltonian circuit in a complete weighted graph for which the sum of the weights of the edges is minimal. The sum of the weights of the edges of a circuit is called the **weight of the circuit**. More generally the Traveling salesman problem asks to find in a weighted (simple) graph a Hamiltonian circuit of least total weight.

· A **Brute Force Algorithm** approach to the Traveling salesman problem is: Check all circuits and select the one with the least total weight! In other words:

− Make a list of all possible Hamiltonian circuits of the graph

− For each Hamiltonian circuit, calculate its total weight by adding the weights of all the edges in the circuit.

− Find the circuit(s) with the least total weight.

Pros: Guaranteed to find the most (cost/weight)−efficient circuit! Cons: Enormous amount of work to carry out the algorithm!
(each increase in the number of vertices increases the work by a factor equal to the number of vertices in the graph)

· The **Nearest Neighbor Algorithm** is an approximate algorithm for the Traveling salesman problem in a complete weighted graph:

− Start at the vertex where the circuit is supposed to begin

− From the starting vertex, go to the vertex for which the corresponding edge has the smallest weight (= **nearest neighbor**).

− Build the circuit, one vertex at a time, by always going from a vertex to the nearest neighbor of that vertex from among the vertices that haven't been visited yet. If there is a tie choose any of the nearest vertices. Keep doing this until all the vertices have been visited.

− Once all the vertices have been visited, from the last vertex, return to the start.

Pros: Much less work then the brute force approach! Cons: It doesn't necessarily produce the optimal Hamilton circuit (so it is a compromise answer).

Math 118 -- Useful information concerning the Final Exam

The final exam will be held on Friday, December 20, 11:00am-1:30pm, Dance Studio and will be comprehensive.   You may not use any notes or books. You will need a calculator! Please come to class 10 minutes early, so that you don't forfeit any of your exam time. At the end of the exam, when you turn in your paper you must sign your name in the list provided and show a picture id.

You must take the final examination at the time scheduled by the university; in particular the exam will not be rescheduled because of travel arrangements. It is your responsibility to schedule travel appropriately!   If you are ill, please email or call me in my office (632-8358) **BEFORE** the test.

Please review all material covered in class, in particular you should review the info sheets for the past two midterms:

- Info sheet for Midterm 1
- Info sheet for Midterm 2

The following list of concepts and "skills" reflects material covered in class since Midterm #2:

• The **Traveling Salesman problem** is to find a Hamiltonian circuit in a complete weighted graph for which the sum of the weights of the edges is minimal. The sum of the weights of the edges of a circuit is called the **weight of the circuit**. More generally the Traveling salesman problem asks to find in a weighted (simple) graph a Hamiltonian circuit of least total weight.

• A **Brute Force Algorithm** approach to the Traveling salesman problem is: Check all circuits and select the one with the least total weight! In other words:
- Make a list of all possible Hamiltonian circuits of the graph
- For each Hamiltonian circuit, calculate its total weight by adding the weights of all the edges in the circuit.
- Find the circuit(s) with the least total weight.

Pros: Guaranteed to find the most (cost/weight)-efficient circuit! Cons: Enormous amount of work to carry out the algorithm!
(each increase in the number of vertices increases the work by a factor equal to the number of vertices in the graph)

• The **Nearest Neighbor Algorithm** is an approximate algorithm for the Traveling salesman problem in a complete weighted graph:
- Start at the vertex where the circuit is supposed to begin
- From the starting vertex, go to the vertex for which the corresponding edge has the smallest weight (= **nearest neighbor**).
- Build the circuit, one vertex at a time, by always going from a vertex to the nearest neighbor of that vertex from among the vertices that haven't been visited yet. If there is a tie choose any of the nearest vertices. Keep doing this until all the vertices have been visited.
- Once all the vertices have been visited, from the last vertex, return to the start.

Pros: Much less work then the brute force approach! Cons: It doesn't necessarily produce the optimal Hamilton circuit (so it is a compromise answer).

• The **Greedy Algorithm** is also an approximate algorithm for the Traveling salesman problem in a complete weighted graph:
- Begin by choosing the edge of least weight and marking this edge. If there is a tie choose any of the edges

with that weight.
- At every stage in the algorithm the next edge should be an unmarked edge of least weight unless it creates a circuit that does not visit every edge of the graph or unless it results in three marked edges coming out of the same vertex in the graph. Again in case of a tie choose any of the edges with that weight.
- The algorithm ends when the marked edges form a Hamiltonian circuit. The (approximate) solution provided by the algorithm is the circuit that starts at one of the vertices, travels along the marked edges in either (appropriate) direction and returns to the starting vertex.

Pros: Much less work then the brute force approach! Cons: It doesn't necessarily produce the optimal Hamilton circuit (so it is a compromise answer).

For some graphs the **Nearest Neighbor Algorithm** gives a better approximation than the **Greedy Algorithm**, for other graphs the **Greedy Algorithm** may produce a better solution. In some cases both algorithm lead to the same solution.

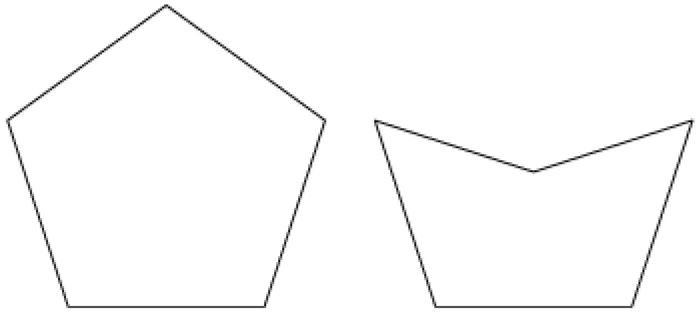
Check this web site for lots of resources on the **Traveling Salesman problem**

A **networking problem** in graph theory asks to efficiently connect all vertices of a (weighted) graph.
• A **subgraph** of a graph is any collection of its edges and vertices that themselves form a graph. In other words a **subgraph** is another graph lying within the original graph.
• Any graph that is connected and contains no circuits is called a **tree**. A subgraph which is a tree containing all vertices of the given graph is called a **spanning tree** of the graph.
• The **weight** of a (weighted) tree is the sum of the weights of its edges.
• A **minimum spanning tree** of a weighted graph is a spanning tree of minimum weight (among spanning trees).

Although no efficient algorithm is known for the **Traveling Salesman problem**, there are simple and efficient algorithms for finding a **minimum spanning tree** of a weighted graph.
• One such efficient algorithms for finding a **minimum spanning tree** of a weighted graph is **Prim's Algorithm** :
- Start at any vertex of the graph, and mark this one as the starting tree for the algorithm.
- From among those vertices not yet contained in the current tree, find the one that may be connected to a vertex in the current tree by an edge of the lest weight. Connect this vertex to the current tree by the edge of least weight connecting them. In case of a tie choose any one of them.
- Continue the previous step until all vertices are contained in the tree.

• A **(simple) polygon** is a closed connected plane figure consisting of a finite number of line segments each of whose endpoints is the endpoint of exactly one other line segment, and such such that there are no other points where two of the line segments intersect. Each of the line segments forming a polygon is called a **side**, each point where two sides meet is called a **vertex**. A **polygon** with n sides is called a ***n*-gon**. The word "polygon" derives from the Greek (poly) meaning "many" and (gonia) meaning "angle."

• A polygon is **convex** if it contains *all* the line segments connecting any pair of its points. A polygon that is not **convex** is called **concave**. Thus, for example, the left pentagon is convex, while the right (indented) pentagon is not! The right one is concave.

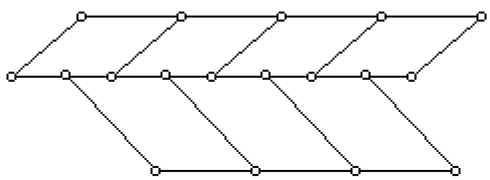


• If all sides and angles of a convex simple polygon are equivalent, the polygon is called **regular**. For example, the left pentagon in the above picture is a regular pentagon.

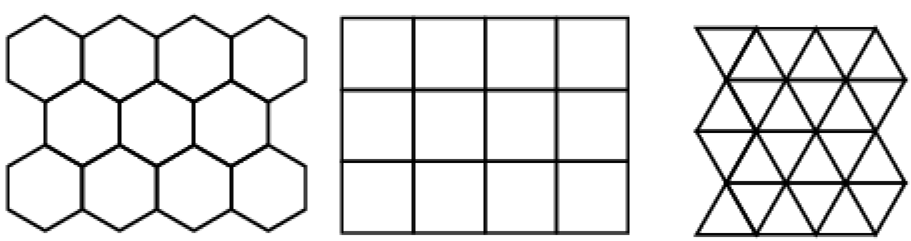
• The sum of the (interior) angles of any n-gon is $(n-2)180$ degrees.

• The measure of the angle of a regular n-gon is $180-360/n$ degree. In particular, an equilateral triangle has angles of 60 degrees, a square has angles of 90 degrees, a regular pentagon has angles of 108 degrees, a regular hexagon has angles of 120 degrees, and so on ...

• A (planar) **tiling** (or **tessellation**) is a plane-filling arrangement of plane polygonal figures (tiles). A **monohedral tiling** is a tiling in which all tiles are congruent. If the corners and sides of the polygonal tiles form all the vertices and edges of the tiling and vice versa, then the tiling is called **edge-to-edge**. Here is part of a tessellation that is not edge-to-edge:



• A (planar) **regular tiling** (or **regular tessellation**) is a an edge-to-edge monohedral tiling for which the tiles are regular polygons. The only **regular tiling** are the following three

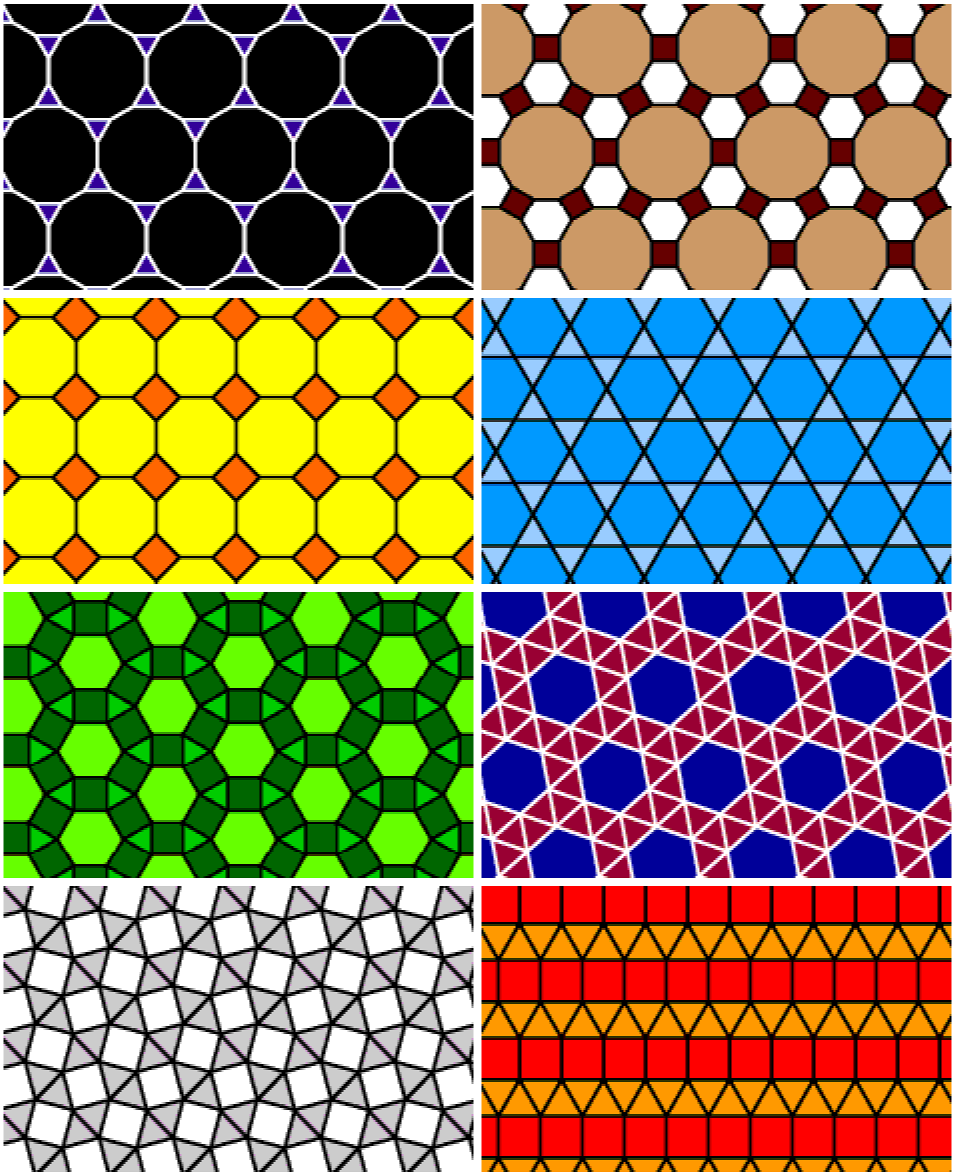


{6, 3} {4, 4} {3, 6}

That is the only regular tilings are those with regular hexagons, with squares and with equilateral triangles. There must be at least three polygons at each vertex. (Why?) There cannot be more than six. (Why?) There cannot be five. (Why?). What is the meaning of the pair of numbers beneath each of the above regular tilings?

• Edge -to-edge tessellations of the plane by two or more convex regular polygons such that the same polygons in the same order surround each polygon vertex are called **semiregular tessellations (tilings)**, or sometimes **Archimedean tessellations (tilings)**. In regular or semiregular tessellations all vertices have the same type. A vertex is said to have **(vertex) type** $n_1, n_2, ..., n_r$ if it is surrounded in cyclic (either clockwise, or counterclockwise) order by regular n-gons with n_1, n_2, ..., n_r edges.

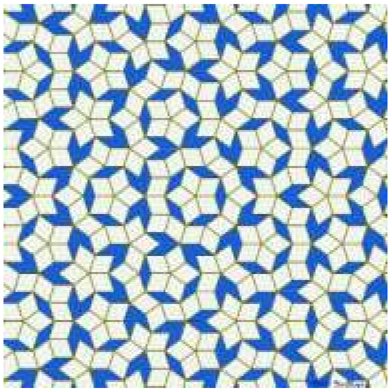
• There are 8 edge-to-edge **semiregular tessellations (tilings)**



Can you describe the vertex type of each of them ? The vertex type 4,6,12 is special, in that it has two different orientations. Labeling a vertex in the clockwise direction results in a different notation from labeling the vertex in the anti-clockwise direction. For the other tilings, direction makes no difference in the notation. In which of the above **semiregular tessellations (tilings)** can you find a vertex with clockwise type 4.6.12 and one with counterclockwise type 4.12.6. Notice that we need to use both orientations in order to tile the plane!

• There are many edge-to-edge monohedral tilings with non-regular polygons:
- Any triangle can tile the plane.
- Any (convex or concave) quadrilateral can tile the plane.
- There are **only** 3 classes of convex hexagons that can tile the plane.
- So far, there are 14 types of convex pentagons that can tile the plane. A complete answer is not known yet.
- A convex polygon with seven or more sides cannot tile the plane!

• All the above tilings are **periodic tilings**. If one draws a periodic tiling on a transparency, then it is possible to translate it, without rotating it, in at least two nonparallel directions, until the transparency exactly matches the original tiling everywhere! A **nonperiodic tiling** is a tiling in which there is no regular repetition of the pattern.
- It was once believed that if a set of one or more tiles could be used to construct a nonperiodic tiling, then the same set of tiles could be used in a different way to construct also a periodic tiling. In 1975 Roger Penrose found an aperiodic tiling of the plane with only two tiles. In its simplest form, it consists of 54 and 72 degree rhombi, with "matching rules" forcing the rhombi to line up against each other only in certain patterns. It can also be formed by tiles in the shape of (deformed) "kites" and "darts". Its main interest is that it has a five-fold symmetry impossible in periodic tilings; it also has been used to explain the structure of certain "quasicrystal" aloys of aluminium. Here is a **Penrose (type) tiling** :



• A **polyhedron** is a (closed) solid shape whose surface does not intersect itself, with polygonal sides called **faces**, that are arranged so that every edge of each face coincides entirely with an edge of another bordering face. (Poly means "many" and hedron means "flat surfaces".)

• A polyhedron is **convex** if it contains inside it *all* the line segments connecting any pair of points inside the polyhedron.

• For any convex polyhedron with, say, F faces, E edges and V vertices, **Euler's formula** says that

$$F-E+V=2$$

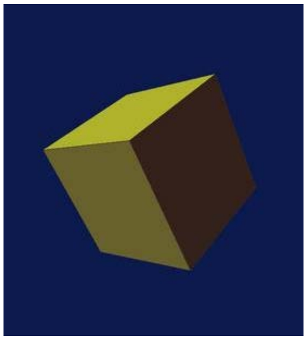
• A convex polyhedron is said to be **regular** if all its faces are regular polygons of the same size and shape, and moreover the number of edges meeting at a vertex is the same for all vertices.

• There are 5 regular polyhedra: the Tetrahedron, the Cube, the Octahedron, the Dodecahedron and the Icosahedron. These solids are also known as the **Platonic solids** since Plato in one of his books, the *Timaeus*, described how these solids relate to the four primordial elements and heaven.
- The regular polyehdra naturally fall into three groups, based on their symmetries and duals: The **Octahedron** and **Cube**, which are duals of each other, form one group, the **Dodecahedron** and **Icosahedron** form another group, while the **Tetrahedron** forms a third group on its own since it is its own dual.
- We have used in class Euler's formula to establish the following table containing their basic invariants:

Polyhedron	Faces	Edges per Face	Vertices	Edges at Vertex	Edges	Dual of...
Tetrahedron	4	3	4	3	6	Tetrahedron
Cube	6	4	8	3	12	Octahedron
Octahedron	8	3	6	4	12	Cube
Dodecahedron	12	5	20	3	30	Icosahedron

Icosahedron	20	3	12	5	30	Dodecahedron

- Here are images of the five Platonic solids

• A **semiregular polyhedron** is a convex polyhedron with all its faces regular polygons, all of its vertices of the same type, with at least two different kinds of regular polygons as faces, and with the additional symmetry that up to mirroring the polyhedron looks exactly the same from every vertex. Semiregular solids are also called **Archimedean solids**.
- There are only 13 **Archimedean solids**, (not counting two mirror images twice).
- Nine of the Archimedean solids can be obtained by truncation of a Platonic solid, and two further can be obtained by a second truncation. The remaining two solids, the snub cube and snub dodecahedron, are obtained by moving the faces of a cube and dodecahedron outward while giving each face a twist. Can you describe the solid obtained by truncating an Icosahedron ? How many faces, edges, vertices does it have the truncated solid? How about same question for the truncated octahedron?

Good Luck!