CSI 2101 / Winter 2015: Discrete Structures.

- CSI2101 Discrete Structures (3,1.5,0) 3 cr.
  - Discrete structures as they apply to computer science, algorithm analysis and design. Predicate logic. Review of proof techniques; application of induction to computing problems. Graph theory applications in information technology. Program correctness, preconditions, postconditions and invariants. Analysis of recursive programs using recurrence relations. Properties of integers and basic cryptographical applications. Prerequisite: MAT1348
- PROFESSOR: Dr. Nejib Zaguia
  - SITE 5031, 562-5800 ext.:6782
  - zaguia@eecs.uottawa.ca
  - Office hours: Monday11:30-13:00
- LECTURES:
  - Monday LEC 10:00-11:30 MCD 146
  - Wednesday LEC 08:30-10:00 MCD 146
  - TUTORIAL Tuesday 17:30 - 19:00 SCS E218 (TA NOT DECIDED YET)
- MANUEL:
  - Kenneth H. Rosen, *Discrete Mathematics and Its Applications*, Seventh Edition

CSI 2101 / Winter 2015: Discrete Structures.

- Evaluation
  - Assignments, 25% (late assignments not accepted). There will be 4 assignments.
  - Mid-Term Test, 25% (Monday, February 23 at 10:00-11:30)

(will take place during class time, closed book)

  - Final Exam, 50%
  - To pass the course, you must obtain a weighted average of at least 50% on the final and Mid-term exams.

Course Plan:

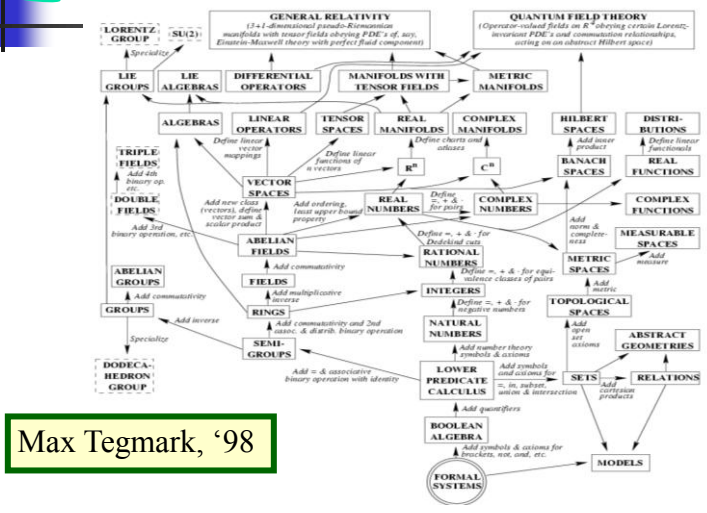
- **Review of Propositional Logic (Chapters 1.1 & 1.2 & 1.3)**
- **Predicates and Quantifiers (Chapters 1.4, 1.5)**
- **Rules of Inference, Proofs – Methods and Strategies (Chapters 1.6, 1.7 & 1.8)**
- **Basic Number Theory and applications (Chapters 4.1, 4.3, 4.4, 4.5, 4.6)**
- **Mathematical Induction (Chapters 5.1, 5.2, 5.3)**
- **Program Correctness/Verification and Recursive Algorithms (Chapters 5.4 and 5.5)**
- **Solving Recurrence Relations (Chapters 8.1, 8.2 & 8.3)**
- **Graphs (Chapter 10)**

What is Mathematics, really?

- It's *not* just about numbers!
- Mathematics is *much* more than that:

> Mathematics is, most generally, the study of any and all *absolutely certain* truths about any and all *perfectly well-defined* concepts.

- But, these concepts can be *about* numbers, symbols, objects, images, sounds, *anything*!



Max Tegmark, '98

What are "discrete structures" anyway?

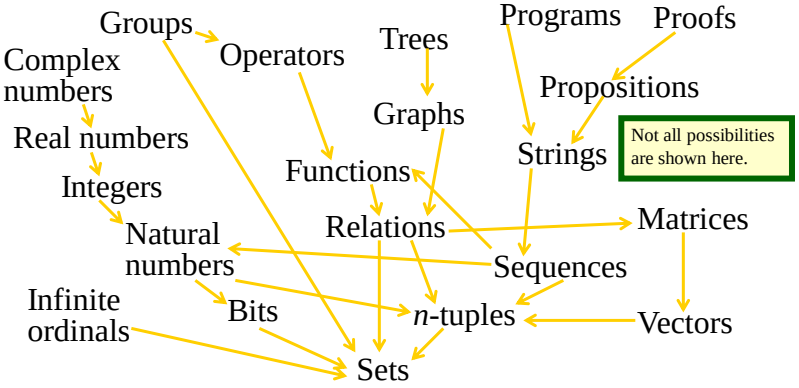
- "***Discrete***" (≠ "discreet"!) - Composed of distinct, separable parts. (Opposite of *continuous*.)
  "***Structures***" - Objects built up from simpler objects according to some definite pattern.
- "***Discrete Mathematics***" - The study of discrete, mathematical objects and structures.

Discrete Structures you will Study in your program

- Propositions
- Predicates
- Proofs
- Sets
- Functions
- Orders of Growth
- Algorithms
- Integers
- Summations
- Sequences
- Strings
- Permutations
- Combinations
- Relations
- Graphs
- Trees
- Logic Circuits
- Automata

Relationships Between Structures

- "$\rightarrow$" :$\stackrel{\text{def}}{=}$ "Can be defined in terms of"



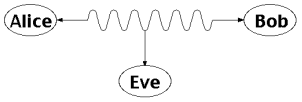
Why Study Discrete Math?

- The basis of all of digital information processing is: *Discrete manipulations of discrete structures represented in memory.*
- It's the basic language and conceptual foundation for all of computer science.
- Discrete math concepts are also widely used throughout math, science, engineering, economics, biology, *etc.*, …
- A generally useful tool for rational thought!

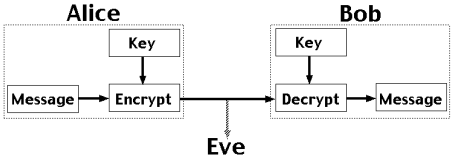
Uses for Discrete Math in Computer Science

- Advanced algorithms & data structures
- Programming language compilers & interpreters.
- Computer networks
- Operating systems
- Computer architecture
- Database management systems
- Cryptography
- Error correction codes
- Graphics & animation algorithms, game engines, *etc.*…
- *I.e.*, the whole field!

Number Theory: RSA and Public-key Cryptography



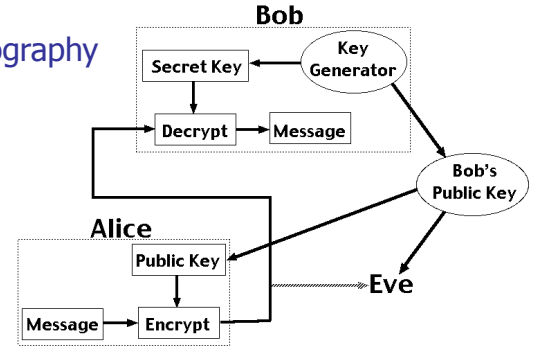
Alice and Bob have never met but they would like to exchange a message. Eve would like to eavesdrop.



They could come up with a good encryption algorithm and exchange the encryption key – but how to do it without Eve getting it? (If Eve gets it, all security is lost.)

CS folks found the solution: *public key encryption*.
Quite remarkable, that is feasible.

Number Theory: Public-key Cryptography



RSA – Public Key Cryptosystem (why RSA?)

Uses modular arithmetic and large primes → Its security comes from the computational difficulty of factoring large numbers.

RSA Approach

Encode:

$C = M^e \pmod n$

M is the plaintext; C is ciphertext

n = pq with p and q large primes (e.g. 200 digits long!)

e is relative prime to (p-1)(q-1)

Decode:

$C^d = M \pmod{pq}$

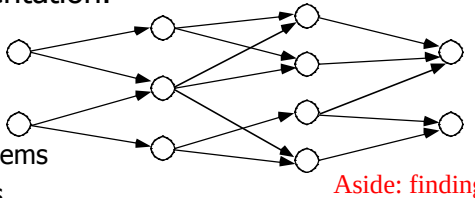
d is inverse of e modulo (p-1)(q-1)

What does this all mean??
How does this actually work?
Not to worry. We'll find out.

The process of encrypting and decrypting a message correctly results in the original message (and it's fast!)

Graphs and Networks

Many problems can be represented by a graphical network representation.

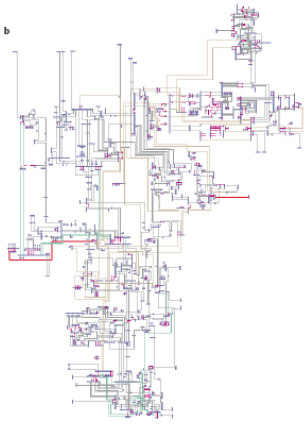


- Examples:
  - Distribution problems
  - Routing problems
  - Maximum flow problems
  - Designing computer/phone / road networks
  - Equipment replacement
  - And of course the Internet

Aside: finding the right problem representation is one of the key issues in this course.

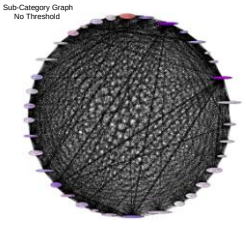
New Science of Networks

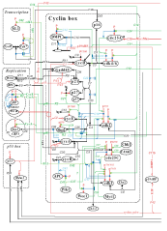
Networks are pervasive



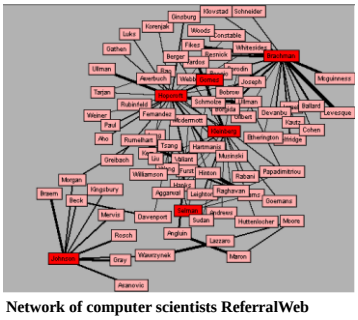
NYS Electric Power Grid (Thorp,Strogatz,Watts)

Utility Patent network 1972-1999 (3 Million patents) Gomes,Hopcroft,Lesser,Selman

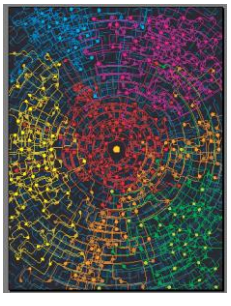




Neural network of the nematode worm C- elegans (Strogatz, Watts)

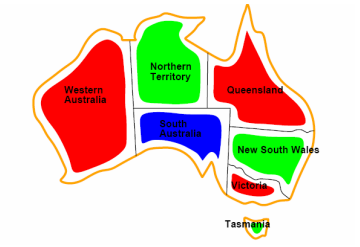


Network of computer scientists ReferralWeb System (Kautz and Selman)

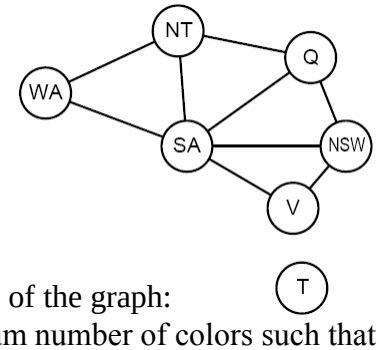


Cybercommunities (Automatically discovered) Kleinberg et al

Example: Coloring a Map: How to color this map so that no two adjacent regions have the same color?



Graph representation

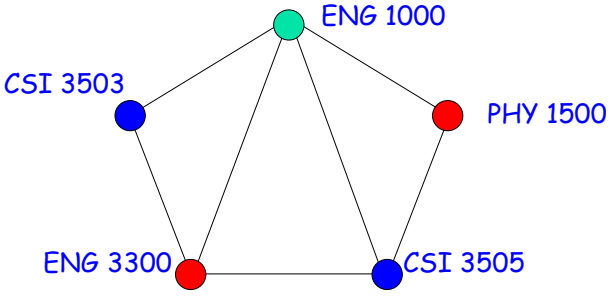


Coloring the nodes of the graph:
What's the minimum number of colors such that any two nodes connected by an edge have different colors?

- The Four Color Theorem – ***Any planar graph can be properly colored by at most four colors.***

Four color map.

- Proof: Appel and Haken 1976; careful case analysis performed by computer; proof reduced the infinitude of possible maps to 1,936 reducible configurations (later reduced to 1,476) which had to be checked one by one by computer. The computer program ran for hundreds of hours.
- The first significant *computer-assisted* mathematical proof.
- *Write-up was hundreds of pages including code!*



Scheduling of Final Exams: How can the final exams at UofO be scheduled so that no student has two exams at the same time? (Note not obvious this has anything to do with graphs or graph coloring.)

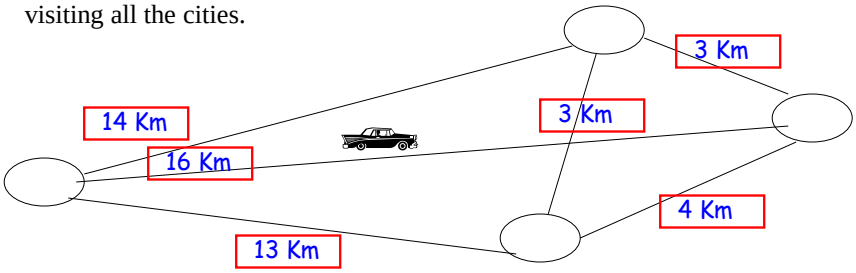
Graph:
A vertex correspond to a course.
An edge between two vertices denotes that there is at least one common student in the courses they represent.
Each time slot for a final exam is represented by a different color.

A coloring of the graph corresponds to a valid schedule of the exams.

Frequency Assignments: T.V. channels 2 through 13 are assigned to stations in North America so that no two stations within 150 miles can operate on the same channel. How can the assignment of channels be modeled as a graph coloring?

•A vertex corresponds to one station;
•There is a edge between two vertices if they are located within 150 miles of each other
•Coloring of graph --- corresponds to a valid assignment of channels; each color represents a different channel.

Traveling Salesman: Find a closed tour of minimum length visiting all the cities.

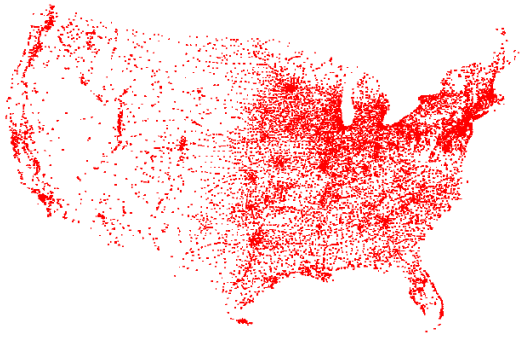


TSP → lots of applications:

Transportation related: scheduling deliveries

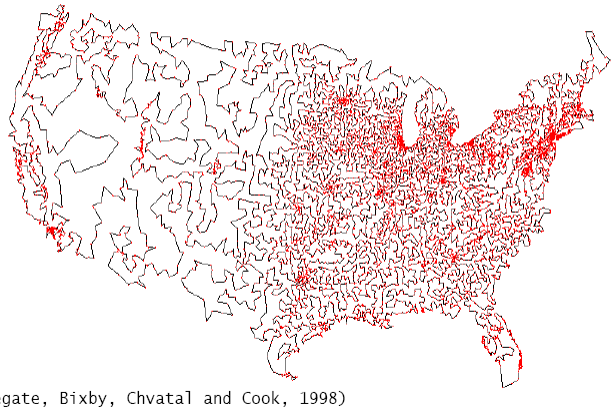
Many others: e.g., Scheduling of a machine to drill holes in a circuit board ; Genome sequencing; etc

13509 cities in the US



13509!= 1.4759774188460148199751342753208e+49936

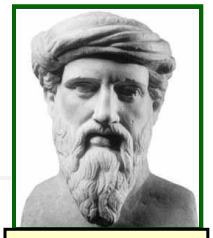
13509 cities in the USA



(Applegate, Bixby, Chvatal and Cook, 1998)

The optimal tour!

- Upon completion of this course, the student should be able to: Think!
  - Check validity of simple logical arguments (proofs).
  - Check the correctness of simple algorithms.
  - Creatively construct simple instances of valid logical arguments and correct algorithms.
  - Describe the definitions and properties of a variety of specific types of discrete structures.
  - Correctly read, represent and analyze various types of discrete structures using standard notations.



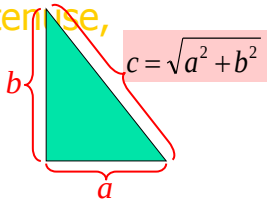
Pythagoras of Samos
(ca. 569-475 B.C.)

A Proof Example

- **Theorem:**

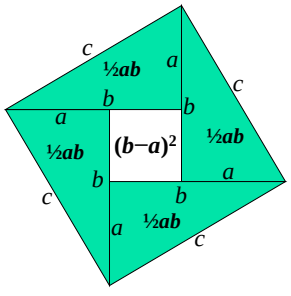
(Pythagorean Theorem of Euclidean geometry)

For any real numbers a, b, and c, if a and b are the base-length and height of a right triangle, and c is the length of its hypotenuse, then $a^2 + b^2 = c^2$.



- **Proof:** See next slide.

- **Proof.** Consider the below diagram:
- Exterior square area = c^2, the sum of the following regions:
  - The area of the 4 triangles = $4(\frac{1}{2}ab) = 2ab$
  - The area of the small interior square = $(b-a)^2 = b^2 - 2ab + a^2$.
- Thus, $c^2 = 2ab + (b^2 - 2ab + a^2) = a^2 + b^2$. ■



Note: It is easy to show that the exterior and interior quadrilaterals in this construction are indeed squares, and that the side length of the internal square is indeed $b-a$ (where b is defined as the length of the longer of the two perpendicular sides of the triangle). These steps would also need to be included in a more complete proof.

Areas in this diagram are in boldface; lengths are in a normal font weight.