01 Course Information and Overview

CS201 Discrete Mathematics

Instructor: Shan Chen

Course Information

- **Instructor:**

  CHEN Shan 陈杉
  Office: Room 614, South Tower, CoE Building
  Email: chens3@sustech.edu.cn

- **Q&As:**

  Office hours: 3pm-5pm, Tuesdays, at my office

  QQ group chat (with all TAs) for online Q&A: 328476025

- **Platform:**

  Blackboard: bb.sustech.edu.cn → "Discrete Mathematics Fall 2024"

Please ask questions in class and your classmates will appreciate that!
希望大家课堂上练习用英语交流（不会的部分用中文代替）

Course Information

- **Grading Scheme:**

Assignments (~6)	20%
Quizzes (~2, open-book)	10%
Midterm (close-book)	30%
Final (close-book)	40%
Project (optional)	–1% ~ +5%

- **Clarification:**

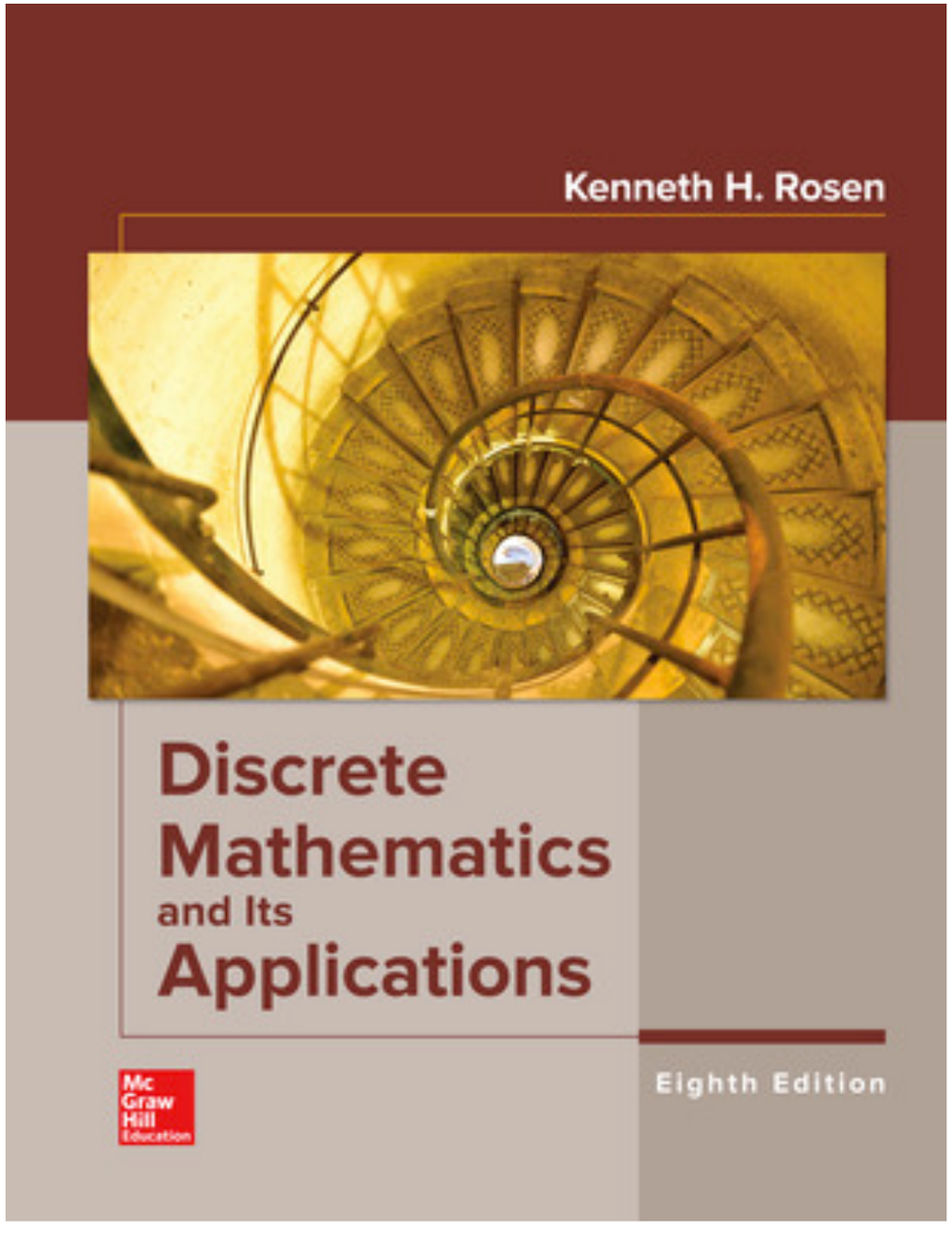
  Most material will be covered in class, but some details might be omitted. You are responsible for learning all content in the assigned sections of the textbook, even for those not explicitly taught in class.

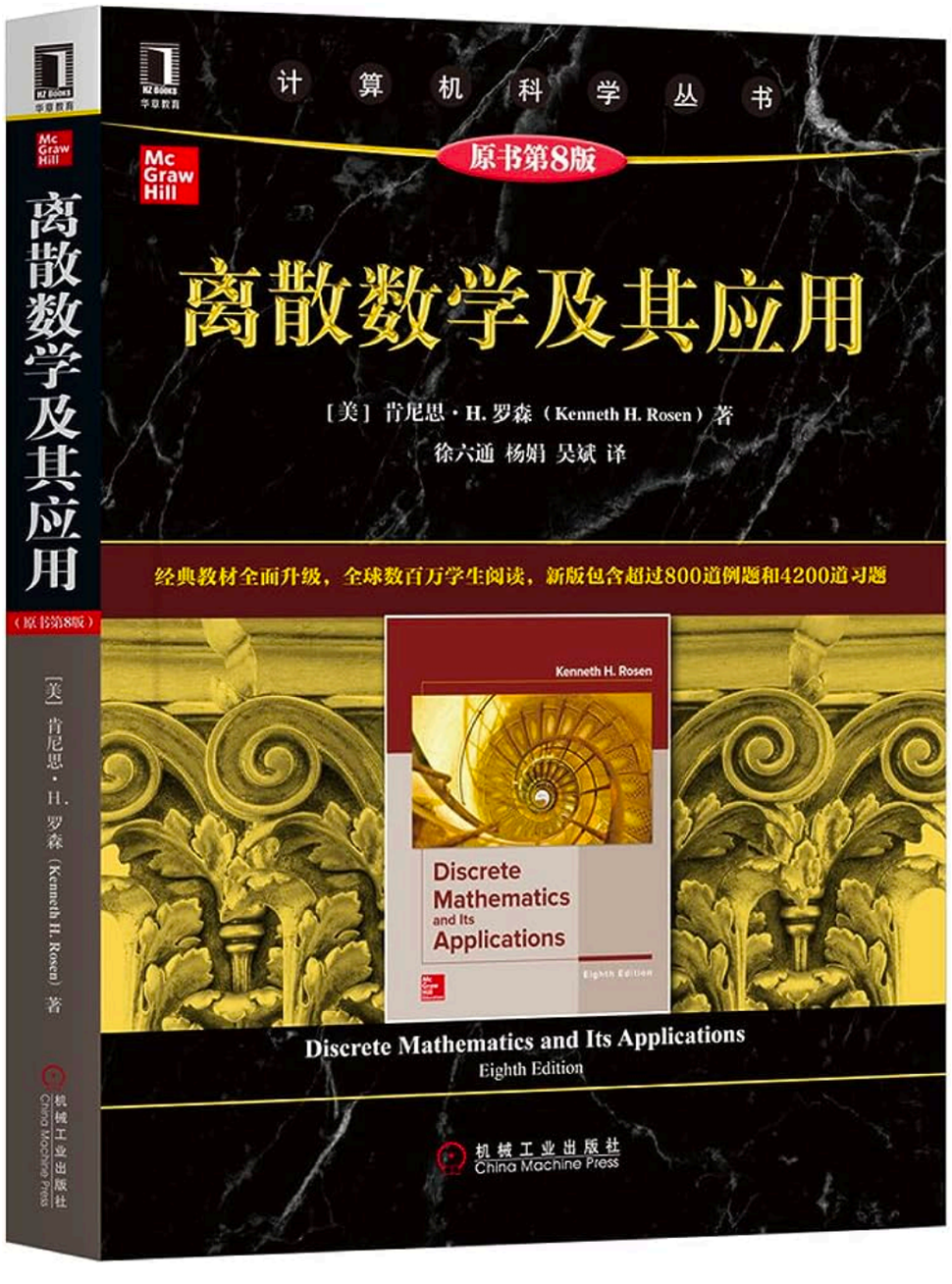
- **Grade consistency:**

  Average course grade roughly aligns with CS201 in other semesters.

Course Information

- **Textbooks:**

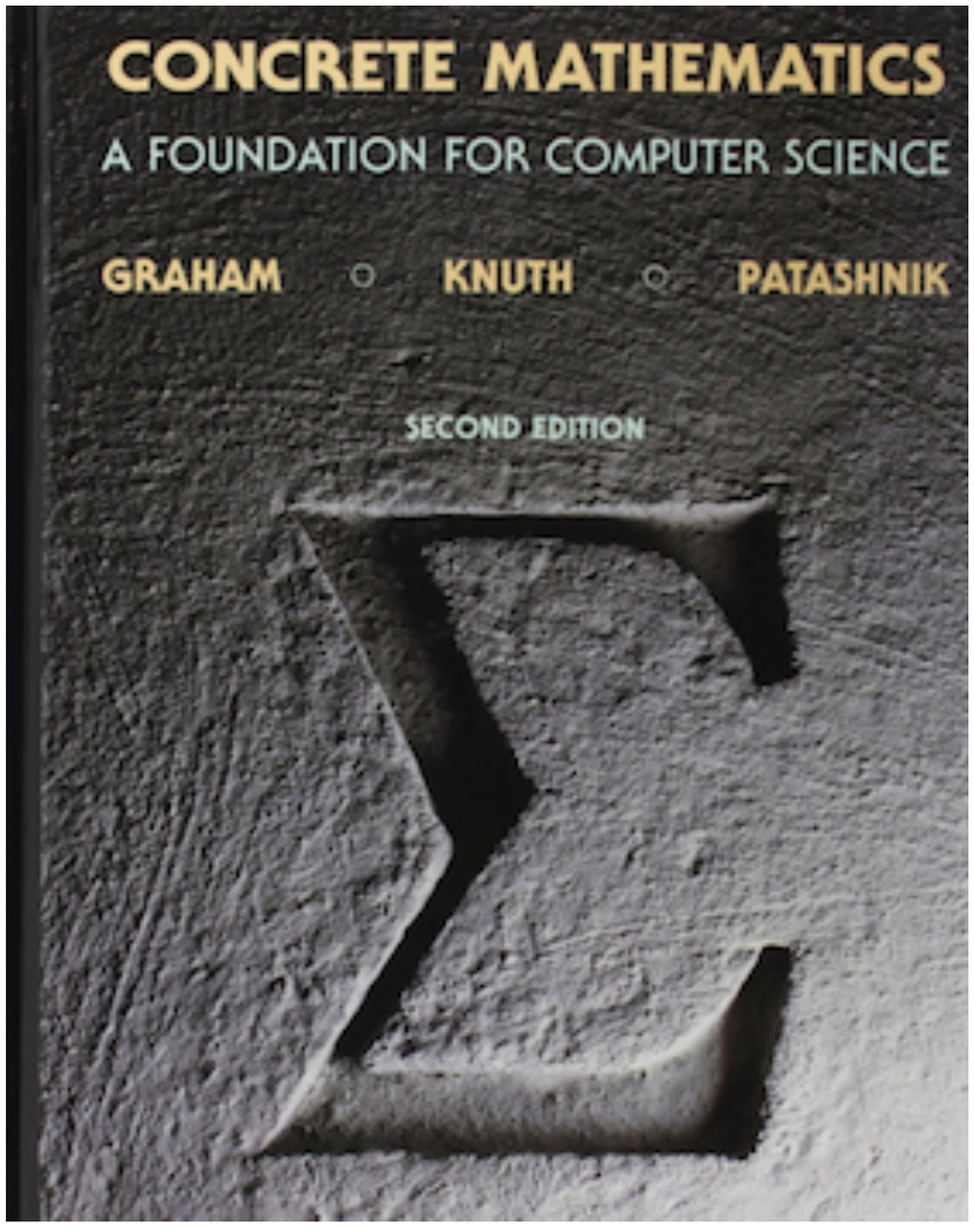


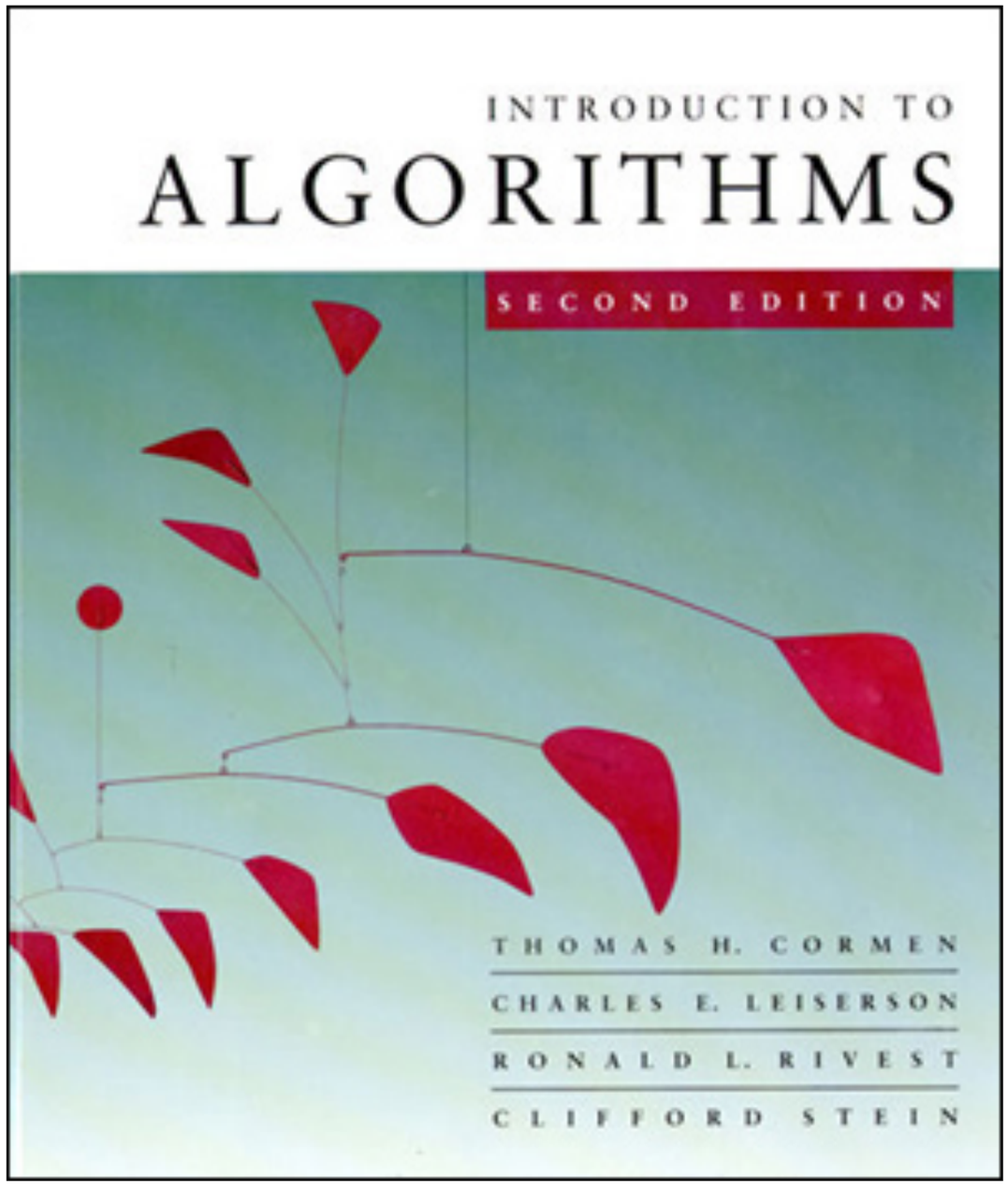


SUSTech

Course Information

- **Reference Books:**





SUSTech

Plagiarism

- **Plagiarism Policy:**
  - If plagiarism is found in a student's assignments, course projects, or exams, the corresponding assignment, course project, or exam will receive a score of 0; If the same student is found to have plagiarized for the second time in the same course, the grade for that course will be 0 points.
  - If a student does not sign the Declaration Form or cheats in the course, including regular assignments, midterms, final exams, etc., in addition to the grade penalty, the student will not be allowed to enroll in the two CS majors through 1+3 mode, and cannot receive any recommendation for postgraduate admission exam exemption and all other academic awards.
  - As it may be difficult to determine who actually wrote it when two assignments are identical or nearly identical, the policy will apply to BOTH students, unless one confesses having copied without the other knowing (uploading your code to public sites like GitHub is considered as one having the knowledge).

Plagiarism

- **What is OK, and What is not OK?**
  - It's OK to work on an assignment with a friend, and think together about the program/solution structure, share ideas and even the global logic. At the time of actually writing the code or assignment, you should write it alone.
  - It's OK to use in an assignment a piece of code or other resources found on the web, as long as you indicate in a comment/reference where it was found and don't claim it as your own work.
  - It's OK to help friends debug their programs (you'll probably learn a lot yourself by doing so).
  - It's OK to show your code/assignment to friends to explain the logic, as long as the friends write their code/assignment on their own later.
  - It's NOT OK to take the code/assignment of a friend, make a few cosmetic changes (comments, some variable names) and pass it as your own work.

Assignment 0

- Please fill out the Undergraduate Students Declaration Form, submit it on Blackboard with your handwritten signature. Otherwise, your course grade will be 0.
  - You can sign on a tablet, but DO NOT simply type in your name!

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY | 计算机科学与工程系 Department of Computer Science and Engineering

Undergraduate Students Declaration Form

This is_____(student ID: _____, who has enrolled in _____ course of the Department of Computer Science and Engineering. I have read and understood the regulations on courses according to "Regulations on Academic Misconduct in courses for Undergraduate Students in the Department of Computer Science and Engineering". I promise that I will follow these regulations during the study of this course.

Course Overview

Acknowledgements

- **My lecture slides are edited from the slides used in**

  CS201 Fall 2021 instructed by Prof. Qi Wang at CSE SUSTech

- **Some slides from the above course are further based on lecture materials used in the following institutions:**

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

STANFORD UNIVERSITY

Massachusetts Institute of Technology

UNIVERSITY OF PITTSBURGH

Let me know if you find mistakes on the slides :)

SUSTech

Discrete Mathematics

- **What is Discrete Mathematics?**

  It studies mathematical structures that are discrete (or countable):

  - e.g., integers, graphs, statements in logic, etc.

  It does not focus on “continuous” mathematics:

  - e.g., real numbers, calculus, etc.

- **Why is it important for us?**

  It provides an essential foundation for studying and describing objects and problems in almost every area of computer science:

  - since computers operate in “discrete” steps and store data in “discrete” memory

Now let’s look at some example topics…

Example Topics

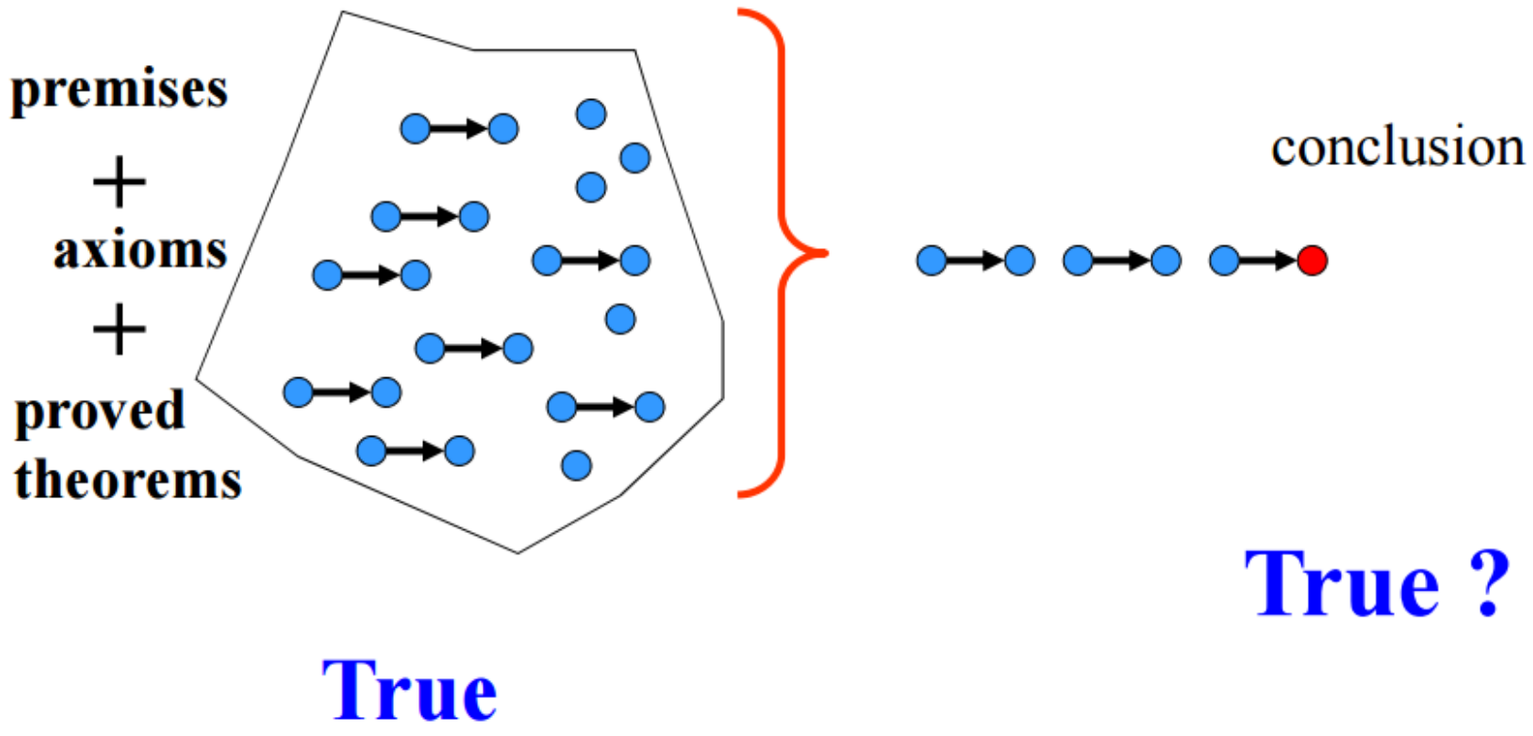
- **Logic and Proofs** (logical formulas are discrete structures)

  translates statements from natural language to symbolic language

  p : I am interested in Discrete Math

  q : I am taking CS201 this semester

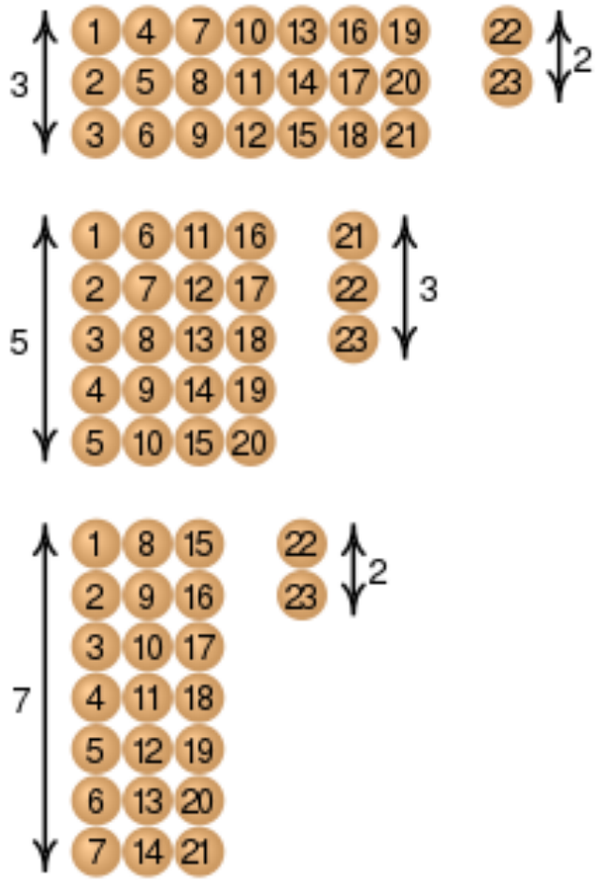
  $p \rightarrow q$: represents "p implies q"



SUSTech

Example Topics

- **Number Theory** (mainly focusing on integers)



How many soldiers in the group? $(x = ?)$

Chinese Remainder Theorem

$$x \equiv 2 \mod 3$$
$$x \equiv 3 \mod 5$$
$$x \equiv 2 \mod 7$$

recorded in ***Sunzi Suanjing*** (200s~400s, China)

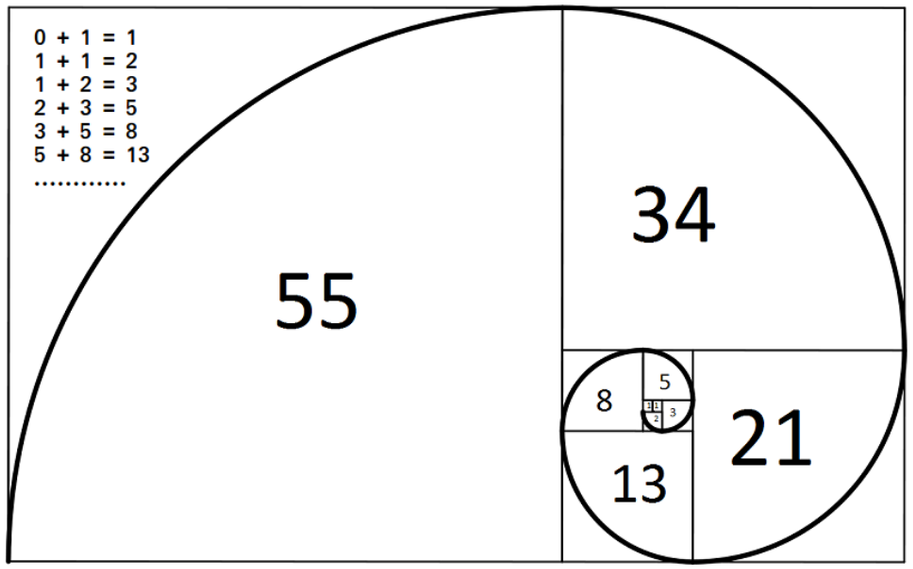
** not **Sun Tzu** that wrote **The Art of War***

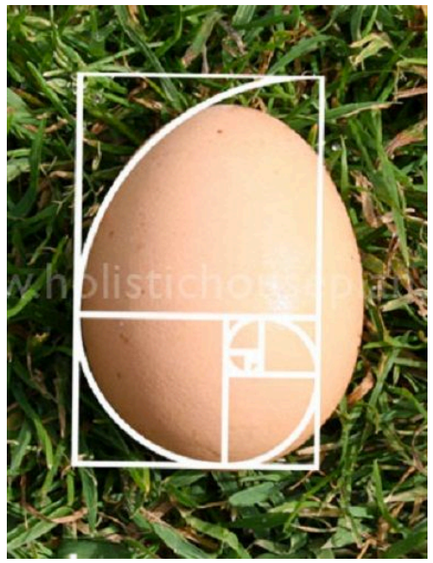
Example Topics

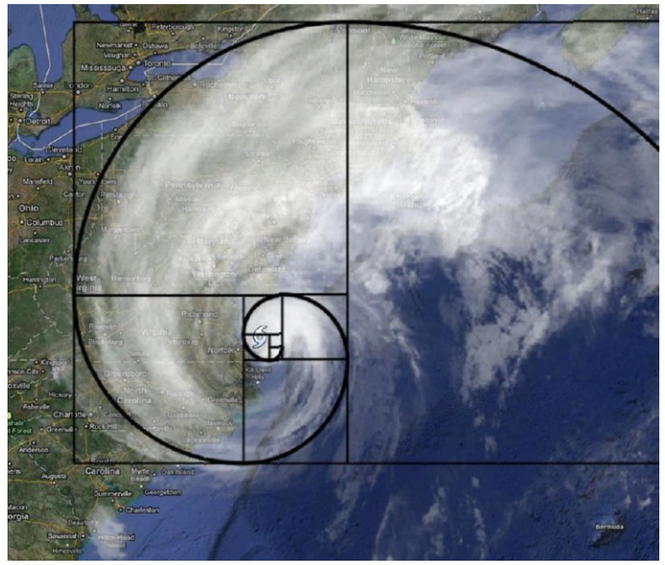
- **Recursion** (many involving functions defined on integers)

Fibonacci sequence:

$$F_0 = 0,\ F_1 = 1,\ \forall n \geq 2 : F_n = F_{n-1} + F_{n-2}$$







Fibonacci spiral ≈ golden spiral (growth factor: golden ratio)

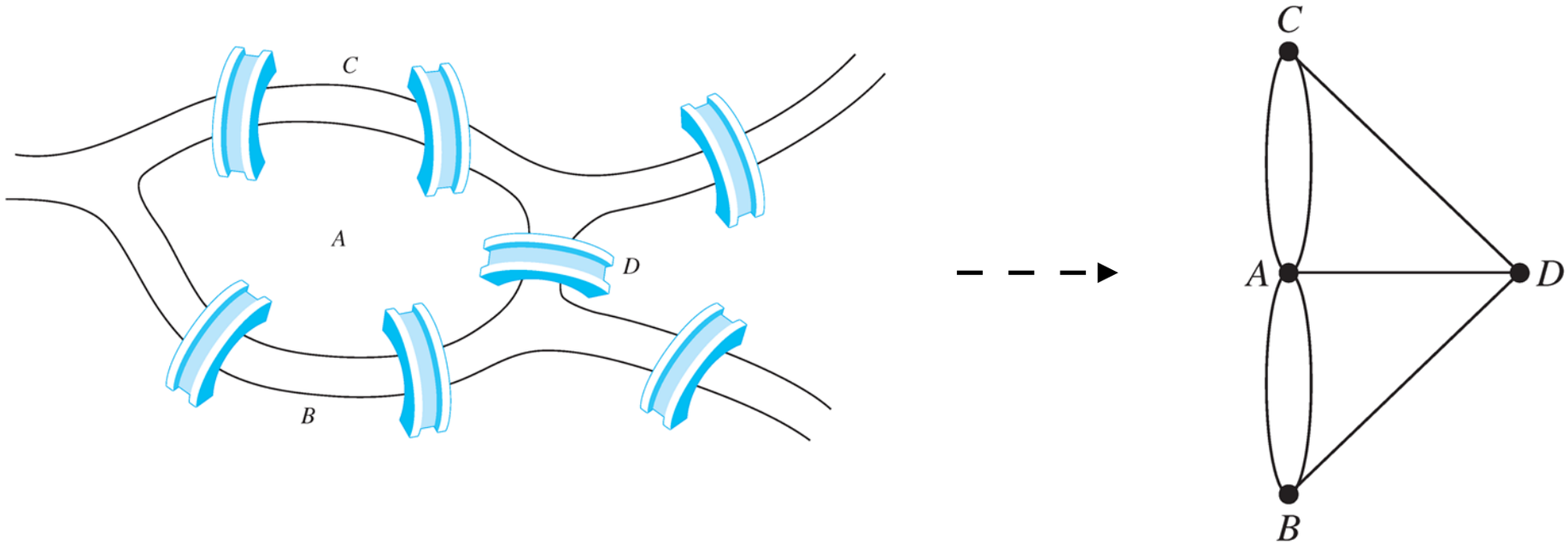
What is the closed-form expression $F_n = ?$

SUSTech

Example Topics

- **Graph Theory** (“discrete” model of real-world problems)

  Seven Bridges of Königsberg Problem: People wondered whether it was possible to start at some location in the town, travel across all the bridges once without crossing any bridge twice.



In 1736, **Leonhard Euler** proved this problem has no solution:

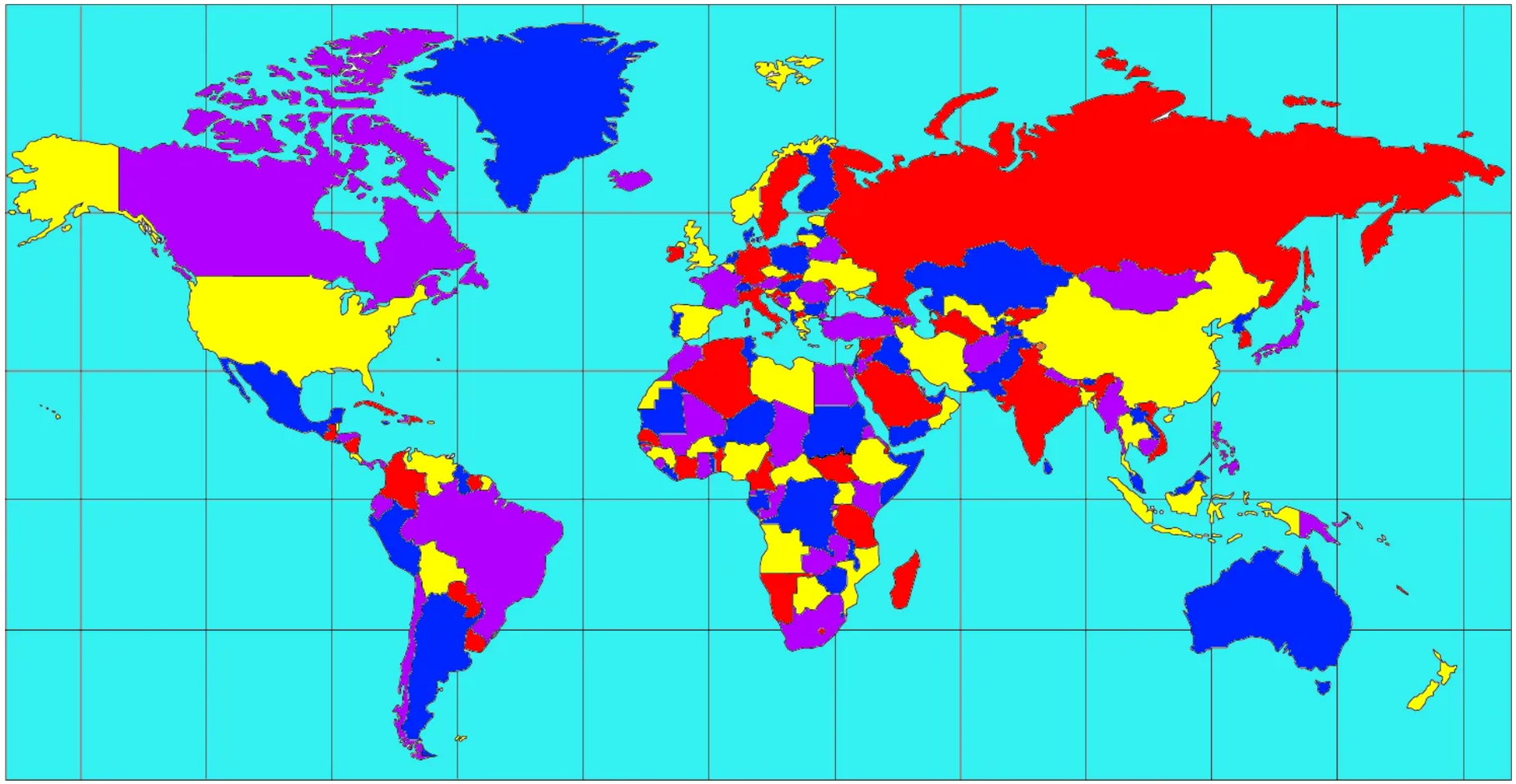
- An Euler walk (traversing each edge once) exists if and only if the graph is connected and it has exactly *0* or *2* nodes of odd degree.

SUSTech

Example Topics

- **Graph Theory** (“discrete” model of real-world problems)

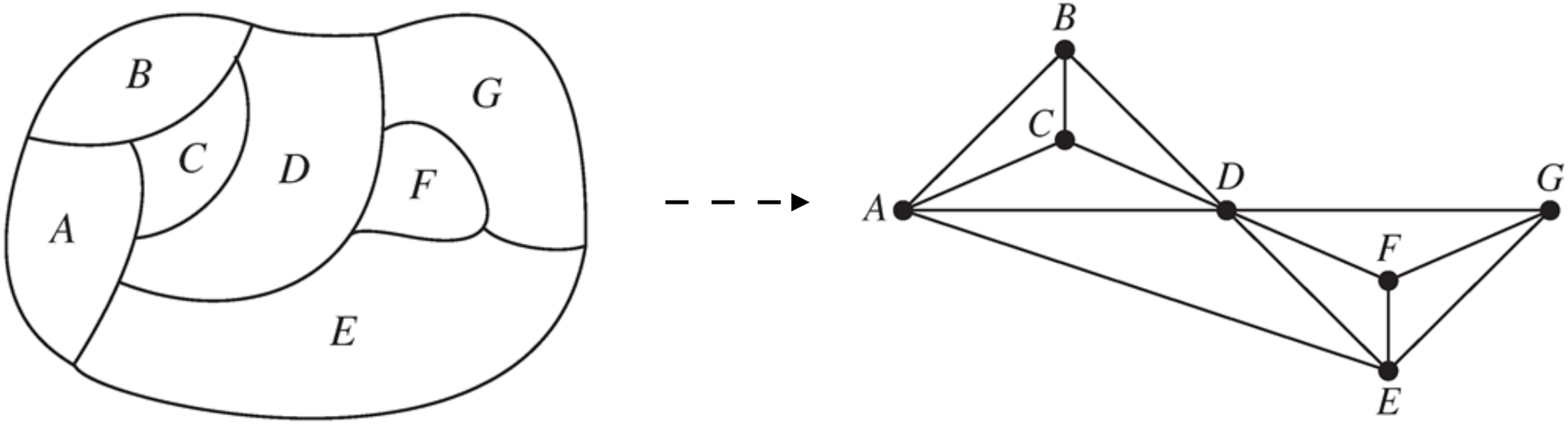
  Four-Color Theorem: Given any separation of a plane into contiguous regions, producing a figure called a map, no more than four colors are required to color the regions of the map so that no two adjacent regions have the same color.



Example Topics

- **Graph Theory** (“discrete” model of real-world problems)

  Four-Color Theorem: Given any separation of a plane into contiguous regions, producing a figure called a map, no more than four colors are required to color the regions of the map so that no two adjacent regions have the same color.



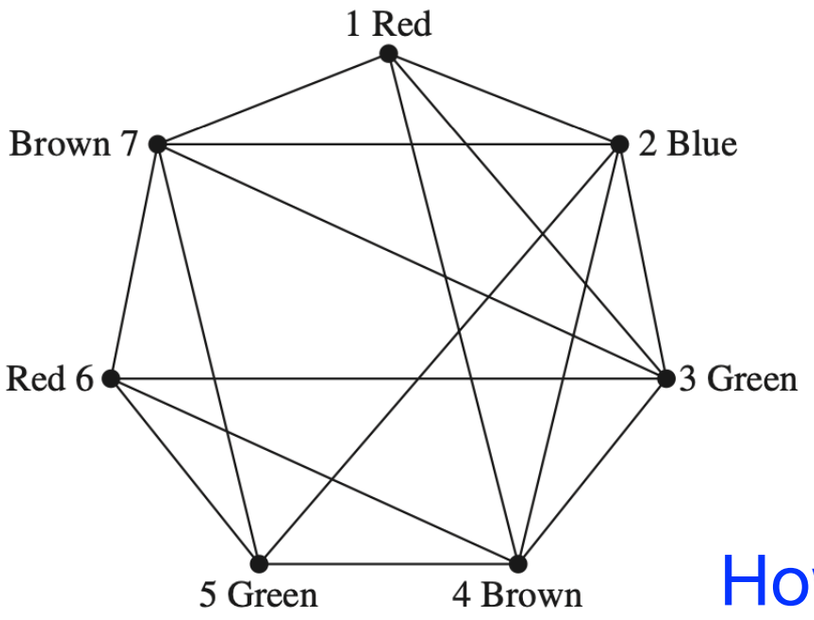
In 1976, **Kenneth Appel** and **Wolfgang Haken** proved it by case-by-case analysis using a computer — the first computer-aided proof.

Example Topics

- **Complexity of Algorithms** (measuring discrete time and space)

  Scheduling Final Exams: How to schedule the final exams at a university such that no student has two exams at the same time?

  - Vertices represent courses, and there is an edge between two vertices if these courses have a common student.
  - This becomes a graph coloring problem: use min number of colors to color all vertices such that adjacent vertices are colored differently.



Time Period	Courses
I	1, 6
II	2
III	3, 5
IV	4, 7

How difficult is graph coloring in general?

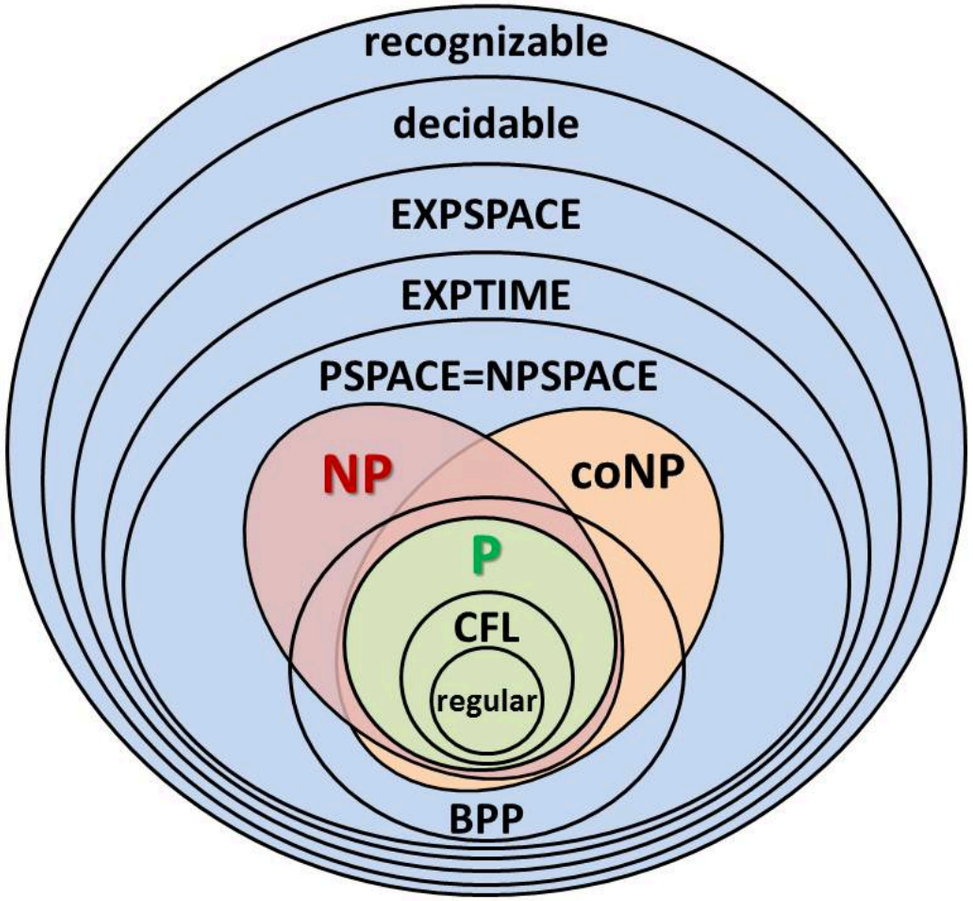
Example Topics

- **Complexity of Algorithms** (measuring discrete time and space)

  Graph Coloring Problem: What is the minimum number of colors such that no two adjacent vertices have the same color?

  - This problem is computationally very hard.

  How to measure computational hardness?



P : e.g., sort *n* numbers

NP : e.g., graph coloring problem

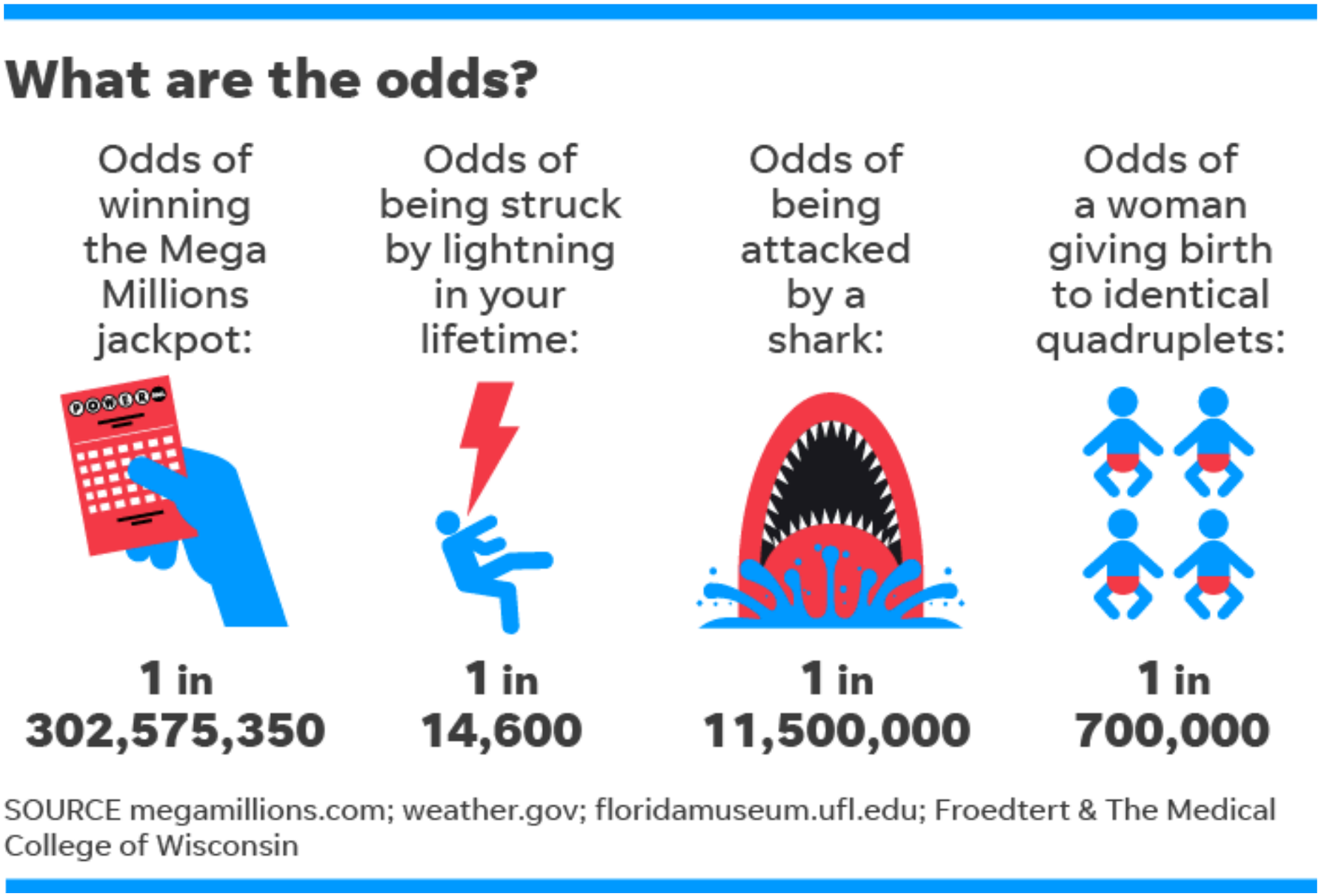
A literal million-dollar question:

P = NP?

SUSTech

Example Topics

- **Combinatorics** (counting)

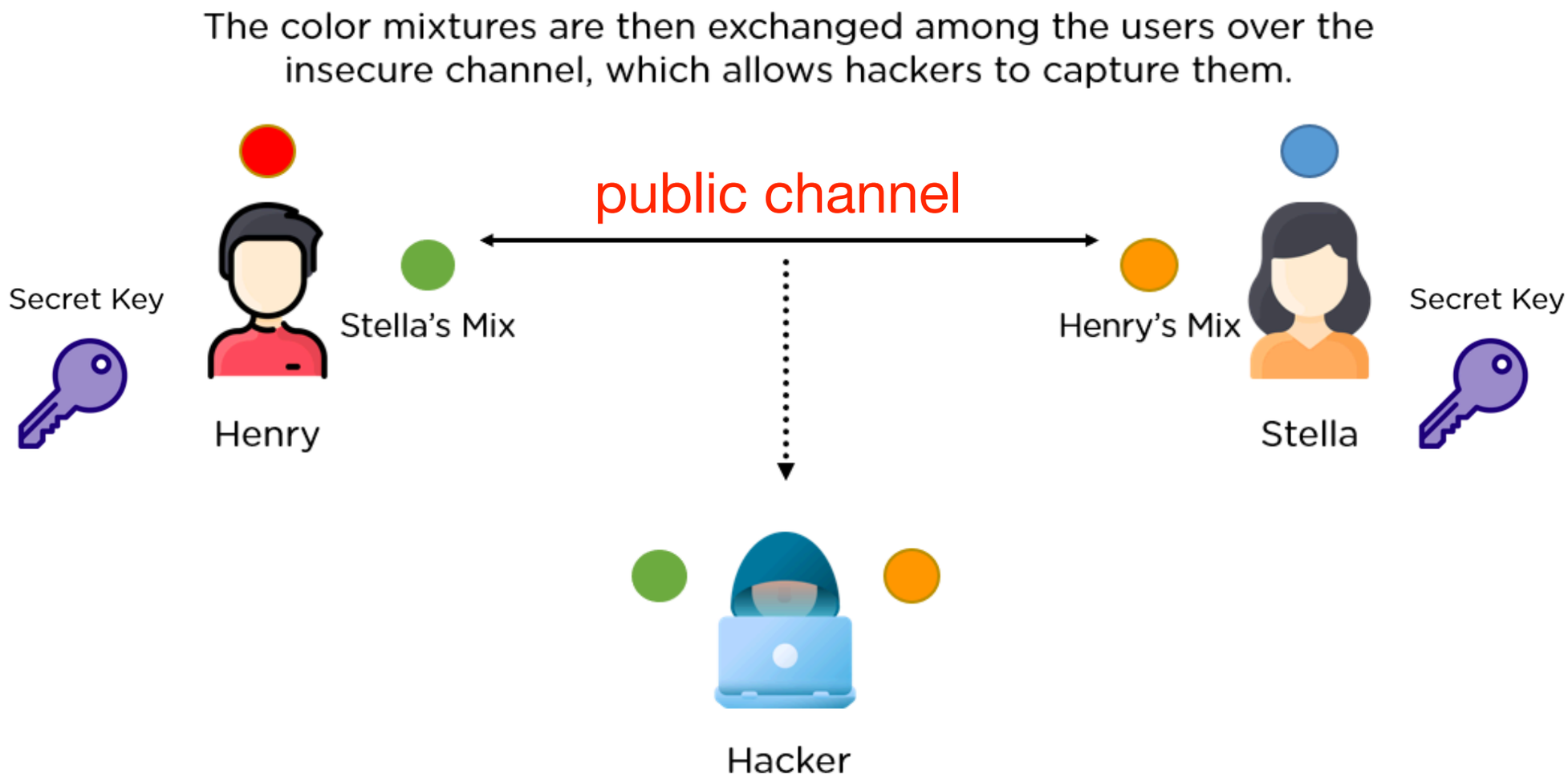


SUSTech

Example Topics

- **Cryptography** (often using number theory) ** my research area :)*

How to establish a shared secret key?



SUSTech

Lecture Schedule (Tentatively)

- Logic and Proofs
- Sets and Functions
- Quiz 1 (around week 4)
- Complexity of Algorithms
- Number Theory
- Cryptography
- Midterm Exam
- Induction and Recursion
- Counting
- Quiz 2 (around week 12)
- Relations
- Graphs
- Trees
- Final Exam

Learning Objectives

- Be able to read, understand, and construct mathematical arguments and proofs.
- Understand the formulations of common problems in several areas of discrete mathematics, including logic and proofs, number theory, cryptography, recursions, counting, relations, graphs and trees, etc.
- Learn a number of discrete mathematical tools.
- Apply discrete mathematical tools to solve certain problems in computer science and engineering.


ARE YOU
READY?

02 Logic and Proofs

To be continued…