

Discrete Mathematics

Welcome & Introduction



What is Discrete Mathematics?

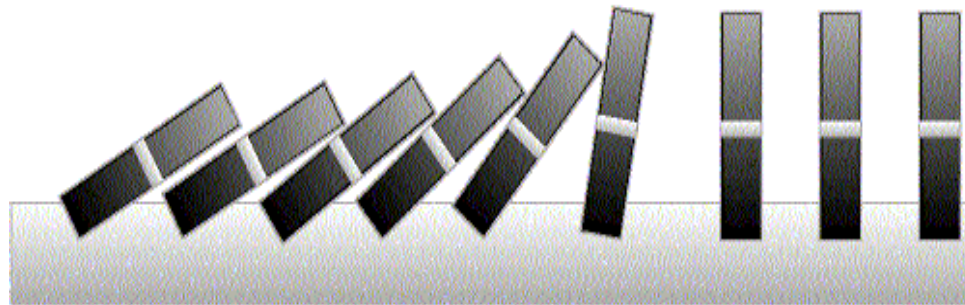
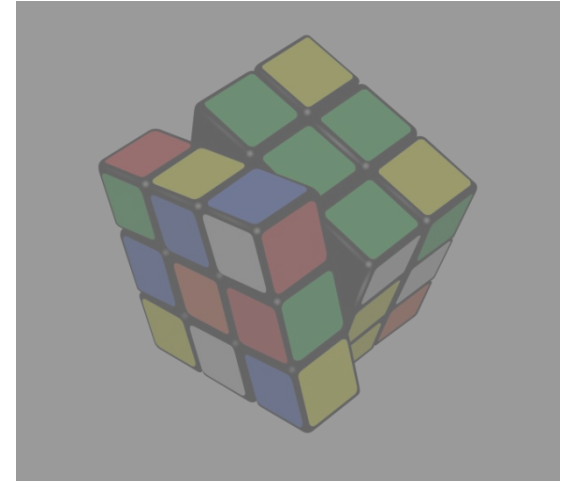
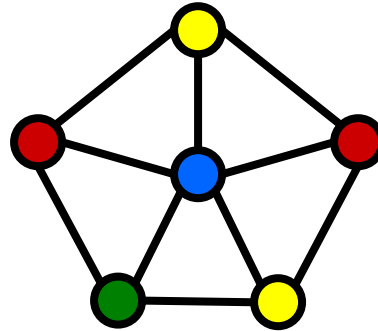
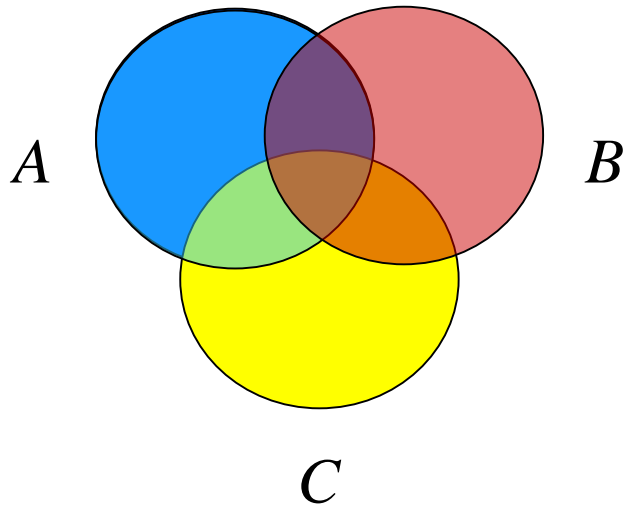
Fundamental Mathematics for Computer Science

discrete mathematics	continuous mathematics
integers	real numbers
graphs	geometric space
induction	calculus
logic	

These two areas are not disjoint, e.g. calculus can be used to solve discrete problems (generating functions).



What is Discrete Mathematics?



Why we learn Discrete Mathematics?

Why discrete mathematics?

In computer science we usually deal with finite, discrete objects. For example,

- We cannot store a real number (infinite precision) in a computer but can only store bits (finite precisions).
- We often model a computer network as a graph, and use the knowledge and techniques in dealing with graphs to solve problems in networks.

The problems and the techniques are often different (e.g. induction, recursion).



Why we learn Discrete Mathematics?

Computer Science: use computer technology to solve problems.

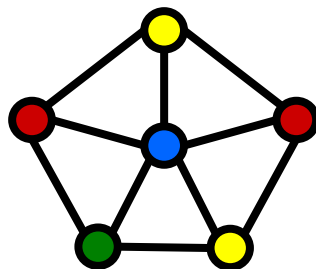
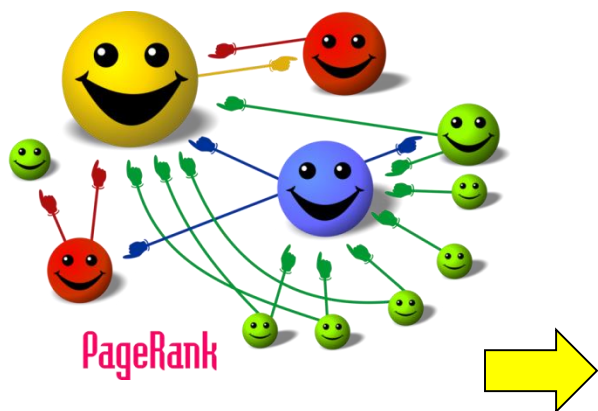
Fundamental Mathematics for Computer Science

Many courses in our curriculum will talk about computer technology.

This course will provide the mathematical foundation to solve problems,

e.g. to design a security system, to design a fast searching algorithm,

to analyze algorithms rigorously (e.g. pagerank and linear algebra), etc.



$$\mathbf{R} = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} \ell(p_1, p_1) & \ell(p_1, p_2) & \cdots & \ell(p_1, p_N) \\ \ell(p_2, p_1) & \ddots & & \vdots \\ \vdots & & \ell(p_i, p_j) & \\ \ell(p_N, p_1) & \cdots & & \ell(p_N, p_N) \end{bmatrix} \mathbf{R}$$

(pictures from wiki)



How to learn Discrete Mathematics?

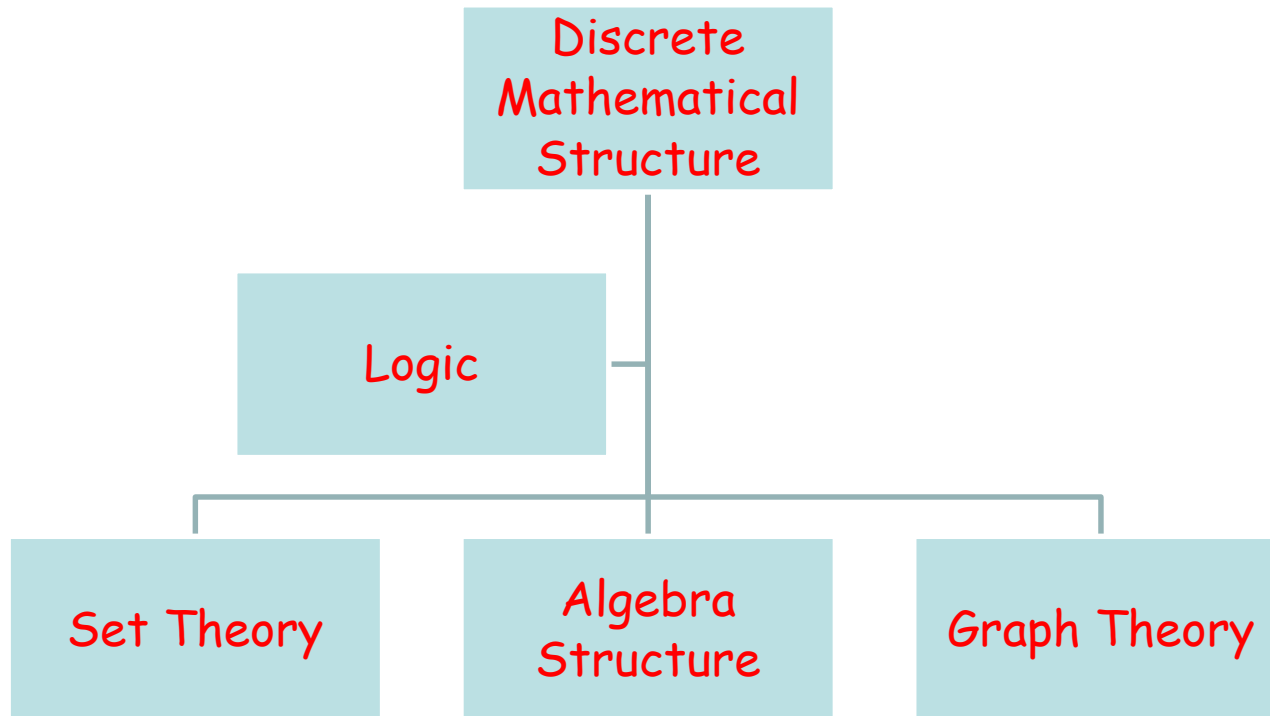
- To learn basic **mathematical concepts**, e.g. sets, relations, algebra structures, graphs
- To be familiar with formal **mathematical reasoning**, e.g. logic, proofs
- To see the **connections** between discrete mathematics and computer science

Knowledge will be used in future courses:

Computer Architecture, Data Structures, Algorithms, Programming Languages, Compilers, Computer Security, Databases, Artificial Intelligence, Networking, Graphics, Game Design, Theory of Computation,



Discrete Mathematics in Our Courses



Logic

- 逻辑是研究推理规律的，它关注的是推理的正确性，逻辑的重点在于研究命题间的关系，而不是一个具体命题的内容
- 考虑如下的论断：
 - 凡是人都是要死的
 - 格拉底是人
 - 所以苏格拉底是要死的
- 从技术上说，逻辑并不帮助确定这些命题是否为真，然而，如果前两个命题为真，逻辑可以保证命题“苏格拉底是要死的”也为真
- 逻辑方法在数学上用来证明定理，在计算机科学中用来证明一个程序做了要求它所做的事
- 用一套符号体系来研究推理的规律，就称为数理逻辑，也称为符号逻辑



Topic 1: Logic (逻辑)

How do computers (and humans) think?

- Propositional logic (命题逻辑)
 - Propositions (命题) and Logical operators (逻辑运算)
 - Propositional equivalences (命题等价)
- Predicate logic (谓词逻辑)
 - Predicates (谓词) and Quantifiers (量词)
 - Nested Quantifiers (嵌套量词)
- Proofs
 - Valid Arguments (有效论证)
 - Rules of Inference (推理规则)

Applications: artificial intelligence, database, circuit, algorithms

Objective: to reason rigorously and learn basic proof techniques (e.g. induction)



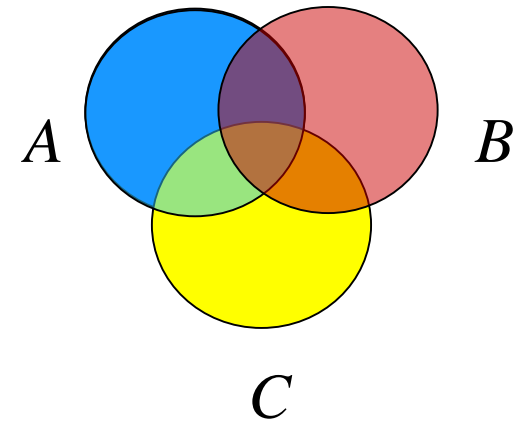
Set Theory

- 集合是现代各科数学的**最基本概念**。集合的思想起源很早，古希腊的原子论学派就把直线看成一些原子的排列。16世纪末，为了建立微积分的可靠基础，人们对数的集合进行了更深入的研究。直到1873年著名德国数学家G. Cantor (1845-1918) 发表了一系列有关集合的论文，奠定了(公理)集合论的基础
- 集合论在数学及其他各学科如自然科学以及社会科学中已经成为必不可少的描述工具。**现代数学与离散数学的“大厦”就是建立在集合论之上的**。随着计算机科学技术的迅速发展，集合论的原理和方法已经成为其主要的理论基础，集合的元素由数学的数集、点集扩展成为包含文字、符号、图形图像、声音等多媒体的信息，构成了包含各种数据类型的集合
- 集合不仅可以用来表示**数及其运算**，还可以用来表示和处理**非数值信息**，如数据的增加、删除、修改、排序以及数据间关系的描述等，这些难以用传统的数值计算进行操作，但用集合运算来处理就十分方便，从而集合论在**编译原理、开关理论、信息检索、形式语言、数据库**等各领域得到了广泛的应用和发展



Topic 2: Set Theory (集合论)

- Sets (集合)
- Functions (函数)
 - Injection (单射), Surjection (满射) and Bijection (双射)
 - Inversion (逆) and Composition (复合)
- Relations (关系)
 - Equivalence Relations (等价关系)
 - Partial Order Relations (偏序关系)



Applications: probability, data structures, algorithms

Objective: to learn basic concepts (sets, functions relations) and fundamental techniques.

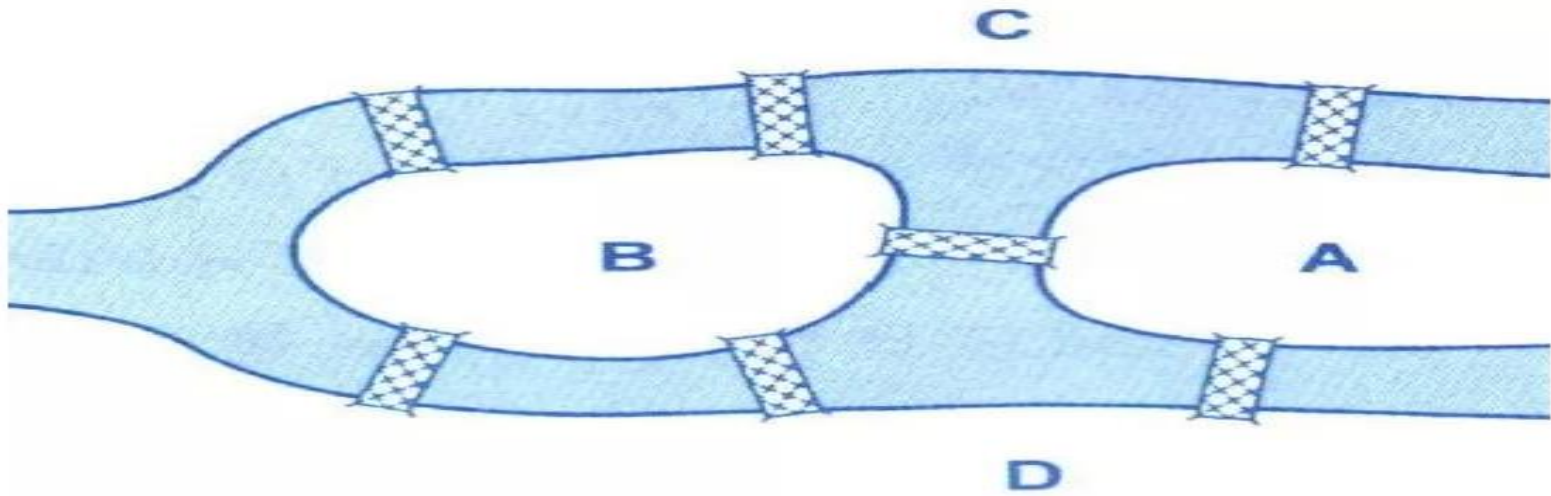


Graph Theory

- 图论是数学的一个古老分支。近年来随着计算机科学的发展而广泛应用，并注入了大量的新鲜内容。该学科起源很早，18世纪出现的“哥尼斯堡七桥问题”引起了人们极大的兴趣，最终由欧拉于1736年发表的一篇论文得以解决。在对图着色问题的研究基础上，曾提出了著名的“四色猜想”。一个多世纪以来，许多数学家为此作出了许多尝试，终于在1976年，美国的阿佩尔和黑肯借助计算机经过1200多个机器小时，作了100亿个判断，终于完成了“四色猜想”的证明。四色猜想的计算机证明轰动了全世界。
“四色定理”对图的着色理论、平面图理论、代数拓扑图论、计算机编码程序设计等分支的发展起到了推动作用。这些似乎无足轻重的游戏引出了许多有实际意义的新问题，开辟了一门新的学科
- 图论被应用到了许多领域，推动这些领域发展的同时，图论本身也得到了迅速发展，克希霍夫把图论应用到电路网络的研究，引入了“树”的概念，是图论向应用方面发展的一个重要标志。化学家凯莱在研究同分异构体的结构中也独立地提出了“树”和“生成树”等概念。近些年来，随着信息时代的发展，尤其作为网络技术的理论基础和研究工具，图论在解决运筹学、电子学、计算机科学、信息论、控制论及网络通信、交通网络、社会科学等领域的问题时，显示出越来越强的效果
- 本篇主要介绍图的基本概念和简单性质、构成元素、分类、连通性及矩阵表示，还将介绍几类特殊的图（欧拉图、汉密尔顿图、平面图、树）等

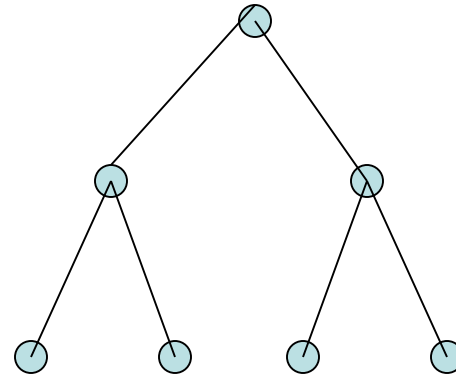
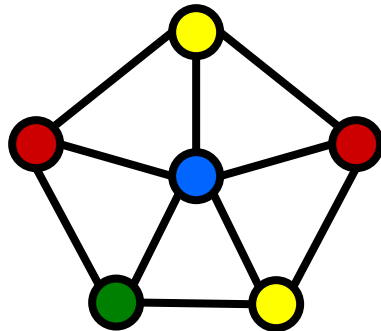


Graph Theory



Topic 3: Graph Theory (图论)

- Direct and Undirected Graphs (有向和无向图)
- Paths (路), Circuits (回路) and Connectivity (连通性)
- Hamiltonian Paths and Circuits (哈密尔顿路和回路)
- Euler Paths and Circuits (欧拉路和回路)
- Trees (树)



Applications: Computer networks, circuit design, data structures

Objective: to model problems and learn basic concepts and knowledge



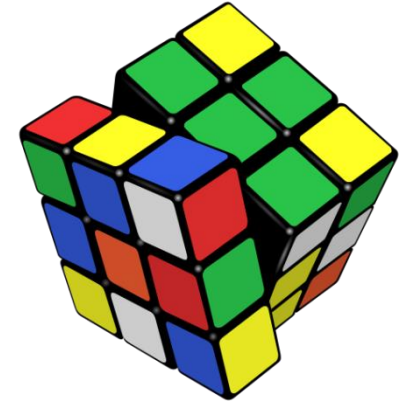
Algebra System

- **代数结构**又称为近世代数或抽象代数，是数学最基础、最重要的分支之一，是在初等代数的基础上产生和发展起来的，代数结构是建立在抽象集合基础上以**代数运算为研究对象的学科**，是用代数的方法从不同的研究对象中概括出一般的数学模型，并研究其**规律、性质和结构**
- 代数学的研究历史十分悠久，人们一直十分关注代数方程的根式解，并做出了许多贡献，1770年法国数学家拉格朗日提出了方程根的排列置换理论，1831年法国数学家伽罗瓦在前人研究成果的基础上，系统地研究了方程根的排列置换性质，提出了群的概念，通过研究与方程相关的群的性质（现称为伽罗瓦群）彻底解决了五次以上方程是否有根式解的问题，成为群论的创建者，伽罗瓦的群的思想是现代数学最重要的概念之一，对数学的许多分支产生了深刻的影响，导致代数学研究对象和方法发生重大变革，开辟了代数学的一个崭新天地，对近代数学的发展产生了极其深远的影响，并导致抽象代数的兴起。
- 代数系统的概念和方法是研究计算机科学和工程的重要数学工具，半群理论在自动机和形式语言中有重要作用，有限域理论是编码理论的数学基础，在通信中发挥重要作用，格和布尔代数理论是电路设计、计算机硬件设计、通信系统设计中的重要工具，抽象代数在网络中关于纠错码的能力方面、描述机器可计算的函数、研究算术计算的复杂性、分析程序设计语言的语义、信息安全及密码学等方面都有广泛的应用



Topic 4: Algebra System (代数系统, optional)

- Operations (运算)
- Algebra Structures (代数结构)
- Homomorphism (同态) and Isomorphism (同构)
- Group (群), Ring (环) and Field (域)
- Lattices (格) and Boolean Algebra (布尔代数)



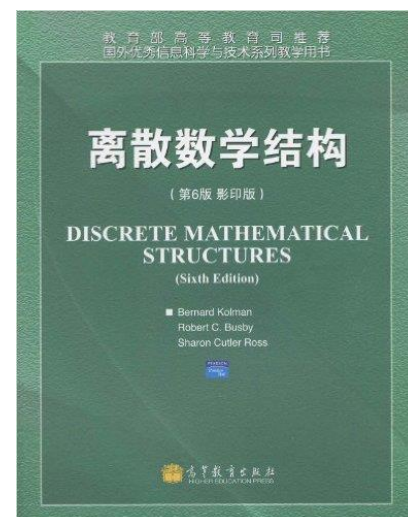
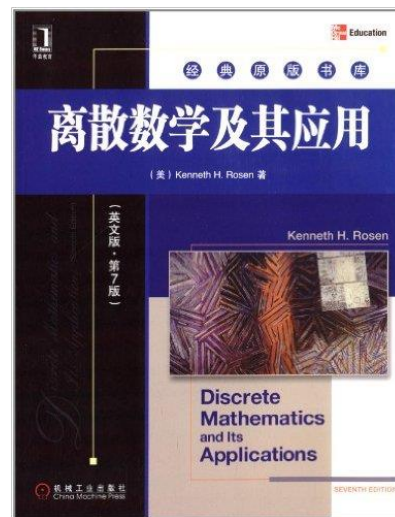
Course Material

■ Textbook:

- Discrete Mathematics and Its Applications (Seventh Edition), Kenneth H. Rosen
- Discrete Mathematical Structures (Sixth Edition), Bernard Kolman, Robert Busdy, Sharon Ross

■ References:

- 《离散数学》陈志奎，周勇编著，邮电出版社
- 《离散数学与算法》曹晓东，原旭编著，机械工业出版社



Tentative Course Schedule

Topics	Courses	lectures
Welcome	Introduction & Overview	1
Logic	Propositional Logic	2-3
	Predicate Logic	4-5
Set Theory	Sets	6
	Functions	7-8
	Relations	9-11
Graph Theory	Graphs	12-13
	Trees	14-15
Review	All	16



Class Policy

- Do not use computers/pads/smart phones in class
- All the slides will be posted on the class web
- Weekly homework must be your own work
- Homework returned in class

Score

- Homework, Class-Exam 30-35%
- Attendances, 5-10%
- Final Exam, 60%



Suggestions

- ✓ Pay more attention to the course itself but not English reading
 - 中文+英文 is ok, but you should finally understand the idea in English
 - English is just the tool, but not the core
- ✓ Both thinking and practices are very important
- ✓ Exams are all about the **slides!! slides!! slides!!**



The End

