

学术研究方法浅论

马 帅



北京航空航天大学
BEIHANG UNIVERSITY

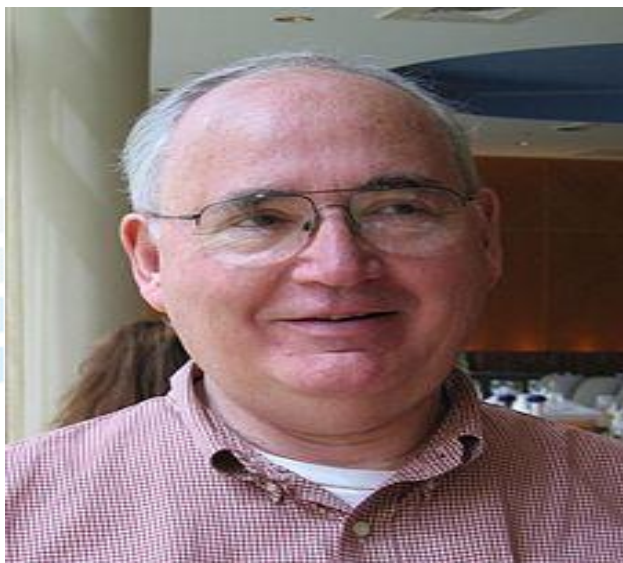
提纲

- ❁ 前辈故事
- ❁ 如何读/写论文
- ❁ 图解科研
- ❁ 励志篇
- ❁ 计算机基础书籍推荐

非确定有限状态自动机理论的开创者



迈克尔·O·拉宾 (Michael Oser Rabin, 1931年9月1日—) 以色列计算机科学家，1976年图灵奖得主。



达纳·斯图尔特·斯科特 (Dana Stewart Scott, 1932年10月11日—) 美国科学家，研究领域涉及计算机科学、数学和哲学，1976年图灵奖得主。

非确定有限状态自动机理论的开创者

- 师兄弟，导师为著名的逻辑学家和计算机专家Alonzo Church
 - ▣ Alonzo Church和Haskell Curry一起提出了与图灵机等价的lambda演算
- 1957年Rabin和Scott一起到IBM研究中心的暑期实习，而IBM允许他们做任何感兴趣的事情😊
- 这样就诞生了非确定有限状态自动机(NFA)
- Scott, Dana; Rabin, Michael (1959). "Finite Automata and Their Decision Problems". IBM Journal of Research and Development 3 (2): 114 - 125.

非确定有限状态自动机理论的开创者

■ 动机：

- 确定有限状态自动机 (DFA) 对于给定的指令集，当输入确定时，输出也是确定的，也就是机器的行为是确定的。
- Rabin和Scott认为这种“确定行为”带来了“局限性”

■ 解决方案：

- 对于给定的输入，允许输出是有限种可能性中的一种选择。实际上，机器的行为从“确定”变为“不确定”，这种机器就是非确定有限状态自动机
- 同时证明了**确定有限状态自动机和非确定有限状态自动机的表达能力是等价的。**

■ NFA意义

- 机器翻译、文献检索和字处理程序中有着重要的作用
- 是计算复杂性中的一个关键的概念，对于定义P和NP类非常重要

NP完全理论的奠基人



史提芬·A·古克 (Stephen A. Cook, 1939年—)，美国计算机科学家，计算复杂性理论的重要研究者。1982年图灵奖得主。

NP完全理论的奠基人

- 哈佛大学研究生院对新兴学科十分重视，在计算复杂性处于萌芽时期邀请了迈克尔·拉宾（Michael Oser Rabin, 76年图灵奖得主）、理查德·斯特恩斯（Richard Edwin Stearns, 93年图灵奖得主）和尤里斯·哈特马尼斯（Juris Hartmanis, 93年图灵奖得主）等人来哈佛讲学或做报告。
- 博士论文《On the Minimum Computation Time for Multiplication》
- 受当时哈佛著名的数理逻辑学家王浩的启发。
 - ▣ 王浩正在研究及其的自动定理证明，并提出了一种图灵机的变形-B-machine。研究一阶逻辑的可满足性的自动证明问题，即是否存在一个模型是的一阶逻辑公式取值为真。
- 在这种环境下，诞生了第一个NP完全问题：
 - ▣ Stephen A. Cook. "The Complexity of Theorem Proving Procedures". STOC, 1971.

NP完全理论的奠基人

■ 动机：

- 虽然，王浩实际上是从计算复杂性的角度研究一阶逻辑的可满足性问题。而本质上这个问题图灵已经解决，即该问题是不可判定的（undecidable）； $P = NP?$ 是一个非常难的问题。
- 但是，自动定理的证明可以作为研究计算复杂性的突破口

■ 解决方案：

- 既然一阶逻辑设计谓词和量词（ \exists 、 \forall ），研究复杂困难，库克从相对简单的命题逻辑着手研究计算复杂性。
- 形式化的定义了NP完全性理论，发现了第一个NP完全问题 - SAT问题（俄罗斯科学家Leonid Levin几乎同时也发现了另一NP完全问题-Tiling问题）。

■ 意义

- 论文开启了NP完备性的研究，令这个领域于之后的十年成为计算机科学中最活跃和重要的研究。

计算复杂性的主要奠基人



尤里斯·哈特马尼斯 (Juris Hartmanis, 1928年—)，拉脱维亚计算机科学家，计算复杂性理论的重要研究者。1993年图灵奖得主。



理查德·斯特恩斯 (Richard Edwin Stearns, 1936年—)，美国计算机科学家，计算复杂性理论的重要研究者。1993年图灵奖得主。

计算复杂性的主要奠基人

- 尤里斯·哈特马尼斯 (Juris Hartmanis) 最初是学物理的，后改学应用数学。理查德·斯特恩斯 (Richard Edwin Stearns) 也是学数学的。
- 斯特恩斯1960年到通用公司去打工，遇到了当时是通用员工的哈特马尼斯。
- 经过几年的合作，诞生了计算复杂性理论
 - Hartmanis, Juris; Stearns, R. E. (1965), "On the computational complexity of algorithms", Transactions of the American Mathematical Society 117: 285 - 306.

计算复杂性的主要奠基人

■ 动机：

- 当时，香农(Claude Elwood Shannon)信息论问世不久，即香农公式：计算在一定的信号和噪声平均功率之下，给定带宽的信道在单位时间内的最大信息通信量。
- 物理出身的哈特马尼斯受此启发，敏锐的想到“抽象的计算过程也应该有精确的定量法则来衡量问题求解的工作量”

■ 解决方案：

- 提出了时间复杂性类TIME ($f(n)$)来衡量基于时间的问题复杂性和并证明了著名的The Time Hierarchy Theorem来描述时间复杂性类的层次关系

■ 意义

- 开辟了计算机科学的一个新的研究领域“计算复杂性”，并奠定了重要的理论基础。

软件可靠性的一个例子



曼纽尔·布卢姆 (Manuel Blum, 1938年4月26日—)，委内瑞拉科学家，密码系统和程序检验先驱，计算复杂性理论的主要奠基人之一，1995年图灵奖得主。

软件可靠性的一个例子

■ 动机:

- Intel公司在推出著名的奔腾Pentium处理器之后，有人发现该处理器的除法运算存在一个问题，在极少数情况下，会导致除法运算的精确度降低。这个缺陷于1994年被发现，变成如今广为人知的Pentium FDIV bug，同时这一事件导致Intel陷入巨大的窘态，创建召回计划来回收有问题的处理器。

■ 和学生Hal Wasserman一起提出了解决方案:

- Manuel Blum, Hal Wasserman: Reflections on the Pentium Bug. IEEE Trans. Computers 45(4): 385-393 (1996)

图论中的最短路径算法

■ 问题:

- 给定带全图 $G(G, V)$ 以及图中两个顶点 u, v , 求解从 u 到 v 的最短路径及其长度



第一个最短路径算法由荷兰科学家艾兹赫尔·迪杰斯特拉 (Edsger Wybe Dijkstra, 1972年图灵奖得主, 1930年5月11日—2002年8月6日) 提出, 时间复杂性为时间复杂度为 $O(|V|*|V|+|E|)$ 。

Dijkstra, E. W. "A note on two problems in connexion with graphs". *Numerische Mathematik* **1**: 269–271, 1959.

图论中的最短路径算法

■ 问题:

- Dijkstra的最短路径算法的效率较慢



迈克尔·弗里德曼 (Michael Fredman), 美国著名计算机学家。



罗伯特·恩卓·塔扬 (Robert Endre Tarjan, 美国计算机科学家, 为1986年图灵奖得主。

- 采用Fibonacci堆将算法时间复杂度提高到 $O(|V| * \log |V| + |E|)$

Fredman, Michael Lawrence; Tarjan, Robert E. "Fibonacci heaps and their uses in improved network optimization algorithms". FOCS, 1984.

故事小结

- 理论型研究
- 应用型研究

■ 新问题&新解决方案

- 非确定自动机NFA
- NP完全问题SAT
- 时间复杂性类Time($f(n)$)

■ 新问题&老解决方案、老问题&新解决方案

- Pentium FDIV bug问题、最短路径问题
- 作为起点

■ 老问题&老解决方案 (Say “No”)

- “Yet another paper on ...”

提纲

- ❁ 前辈故事
- ❁ 如何读/写论文
- ❁ 图解科研
- ❁ 励志篇
- ❁ 计算机基础书籍推荐

How to Evaluate a Paper?

- Novelty of the problem (25%)
- Technical depth (25%)
- Writing (25%)
- Experiments (25%)

How to Get the Idea?

■ Positive

- For any idea, you can always do something

■ Negative

- Extremely challenging to get good ideas.
 - Repeated work is NOT called research!
- Observation – using your brain
- Refine, refine and refine, but with an expectation in your mind!
- Explain by examples

How to Get the Solution?

- Complexity analysis
 - ▣ PTIME, NP, EXPTIME, ...
- Approximation analysis for NPC problems
 - ▣ With performance guarantees
- Heuristic solutions
 - ▣ With certain properties
- No fixed rules to follow for algorithm design
 - ▣ Fully understand the problem
 - ▣ Designed algorithms based on the special characteristics for the problem itself

How to Write the Paper?

- It is art - very difficult!
 - ▣ Practice, practice and practice!
 - ▣ Writing, writing and writing!
 - ▣ Proofreading, proofreading, and proofreading!
- If people could not understand your writing, they could not evaluate your work.
 - ▣ Sir Isaac Newton
- Two good habits
 - ▣ Writing down and remember good sentences when you are reading papers
 - ▣ Ask your “friends”, who could speak truth to you, to check what you have written

How to do Experiments?

- Design experimental plans
- Show people the idea is good, and the solution is good
- Datasets
 - Real life data
 - Synthesized data
- Always remember what you need to show to people!

Stages of Paper Submission

- Submission
- Feedback (optional)
- Shepherd (optional)
- Acceptance/Rejection notification
- Preparing camera ready
- Experimental repeatability (optional)
- Attend conference/present your work
 - Make big noises
 - Show people your good work

Two Rules

- WWH rule
 - ▣ What, why, how
- Think about everything from the view point of reviewers

One Warning

- NO plagiarism!!!



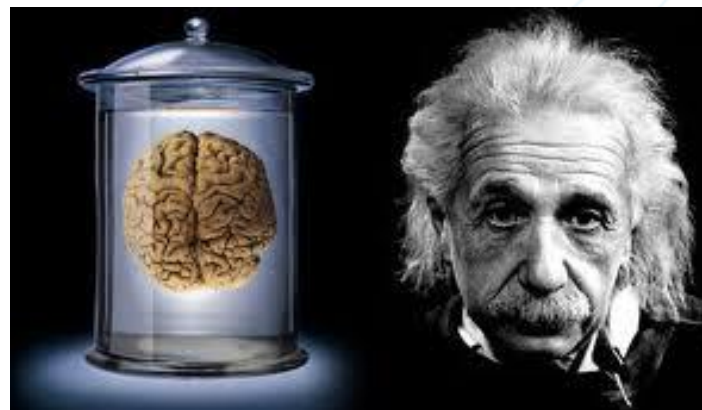
提纲

- ❁ 前辈故事
- ❁ 如何读/写论文
- ❁ 图解科研
- ❁ 励志篇
- ❁ 计算机基础书籍推荐

科研是什么？

Research = Re + search

科研是什么之疯狂的脑袋



科研是什么之根深叶茂



科研是什么之节点控制



科研是什么之合作与独立



科研是什么之根与主干

❁ 坚持主流研究方向

- ◆ 要有“**根**”，要有“**主干**”
- ◆ 可以有“**细枝**”，可以有“**末梢**”

❁ 集中优势兵力，

❁ 瞄准一个方向的**列表** (会议和期刊)

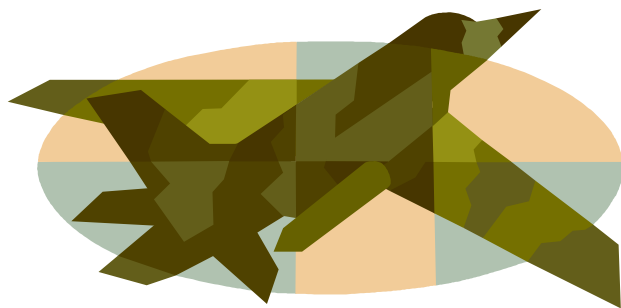


科研是什么之学术道德



违法必究，执法必严！

科研是什么之。 。 。



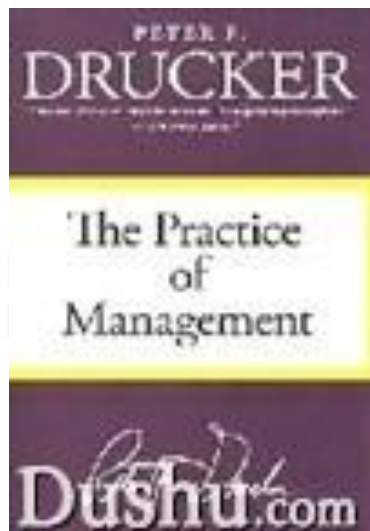
提纲

- ❁ 前辈故事
- ❁ 如何读/写论文
- ❁ 图解科研
- ❁ 励志篇
- ❁ 计算机基础书籍推荐

关于制定目标

● “SMART”法则？

- S (specific) : 明确，不能只是形容概括
- M (measurable) : 可衡量，需要量化
- A (attainable) : 可达到的，不能是遥不可及的
- R (relevant) : 结果导向：与长远目标具有相关性
- T (time-based) : 有时限的

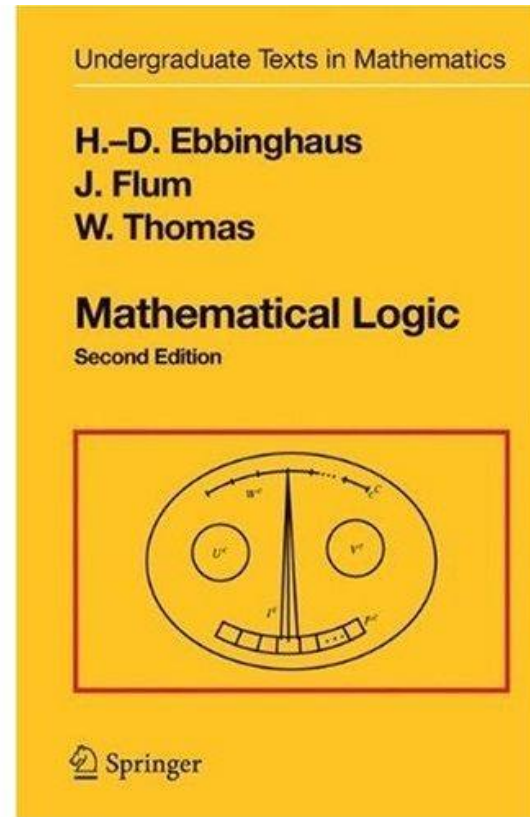
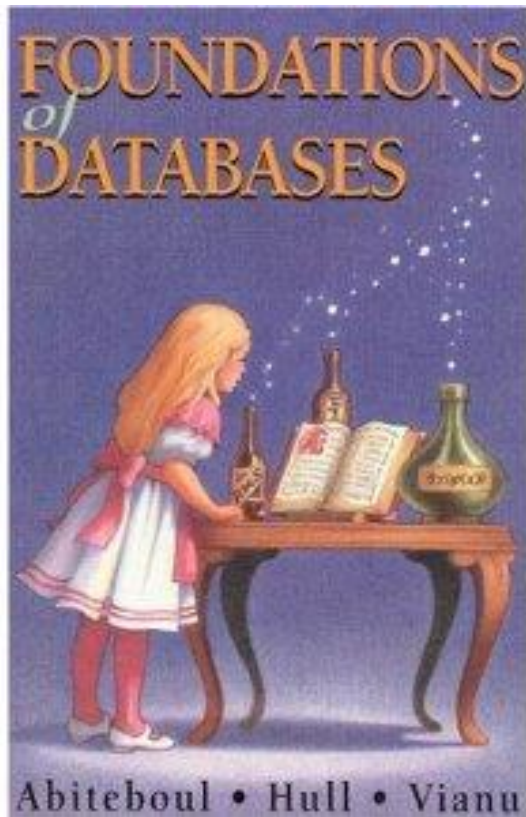


彼得·德鲁克- 维基百科
(Peter F. Drucker)，被
尊称为现代管理学之父

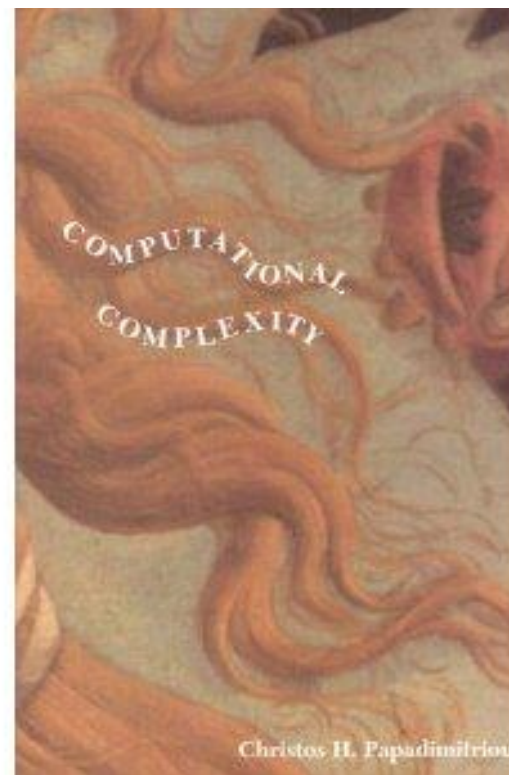
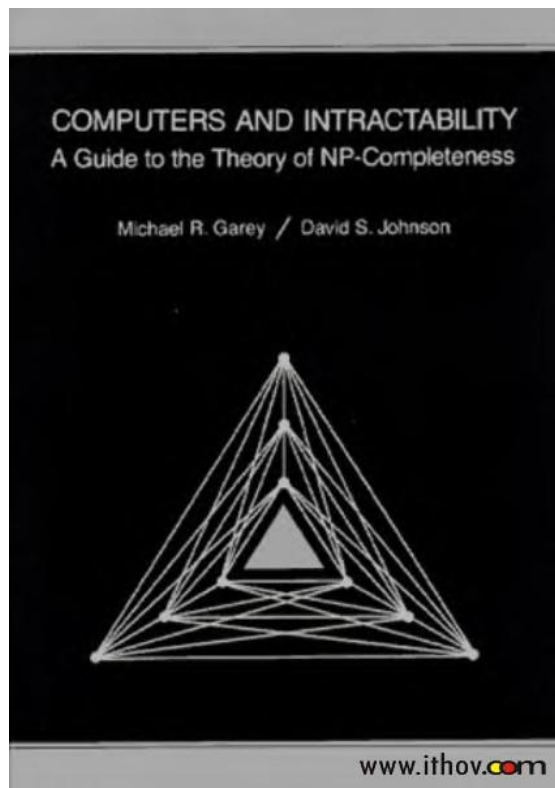
提纲

- ❁ 前辈故事
- ❁ 如何读/写论文
- ❁ 图解科研
- ❁ 励志篇
- ❁ 计算机基础书籍推荐

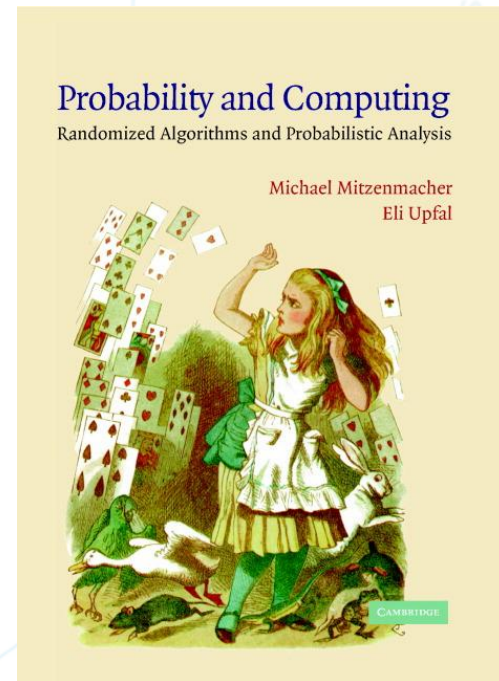
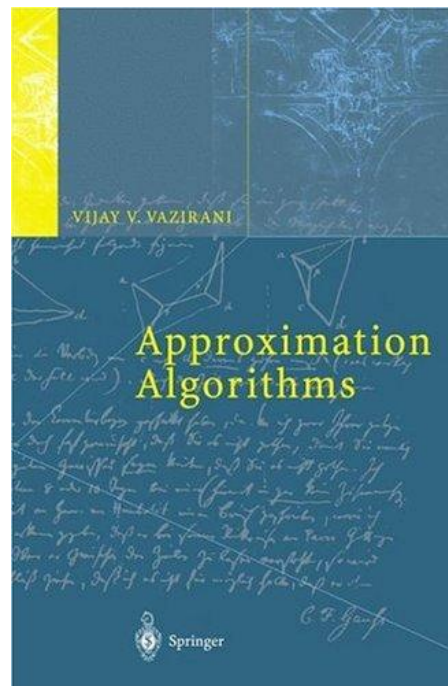
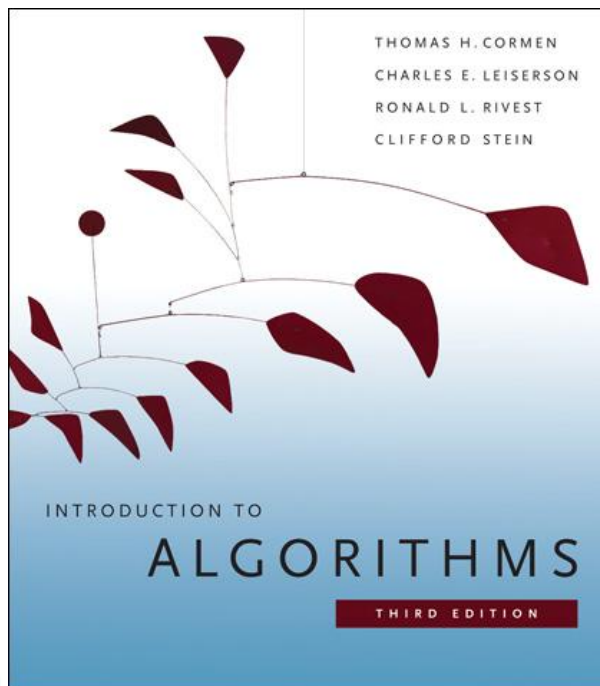
Databases and Logic



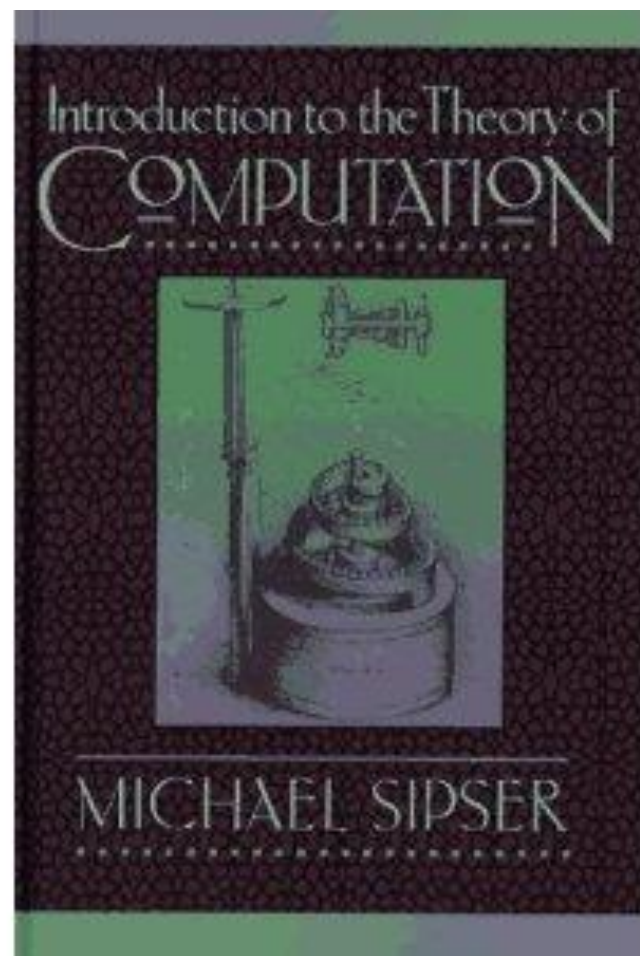
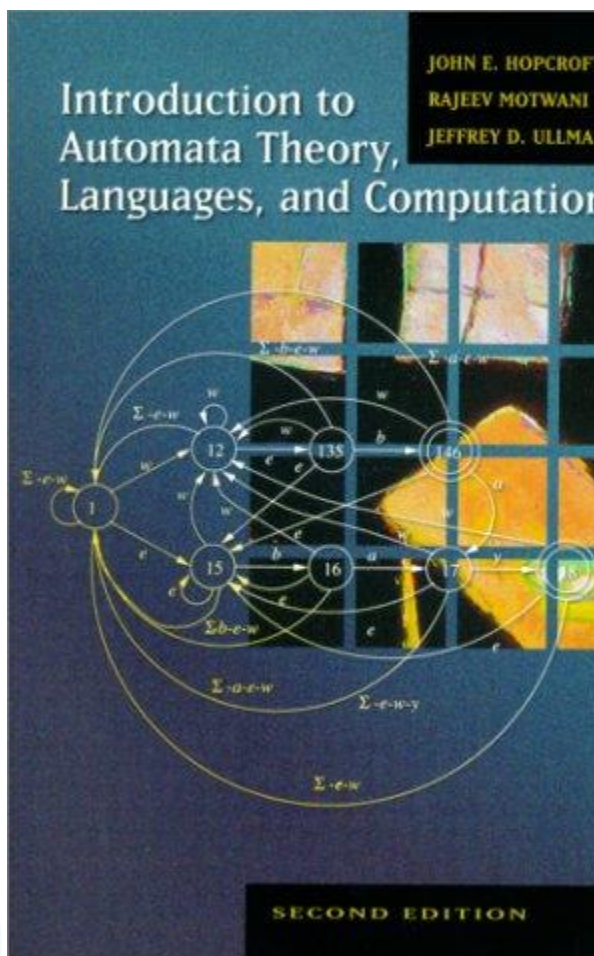
Computational Complexity



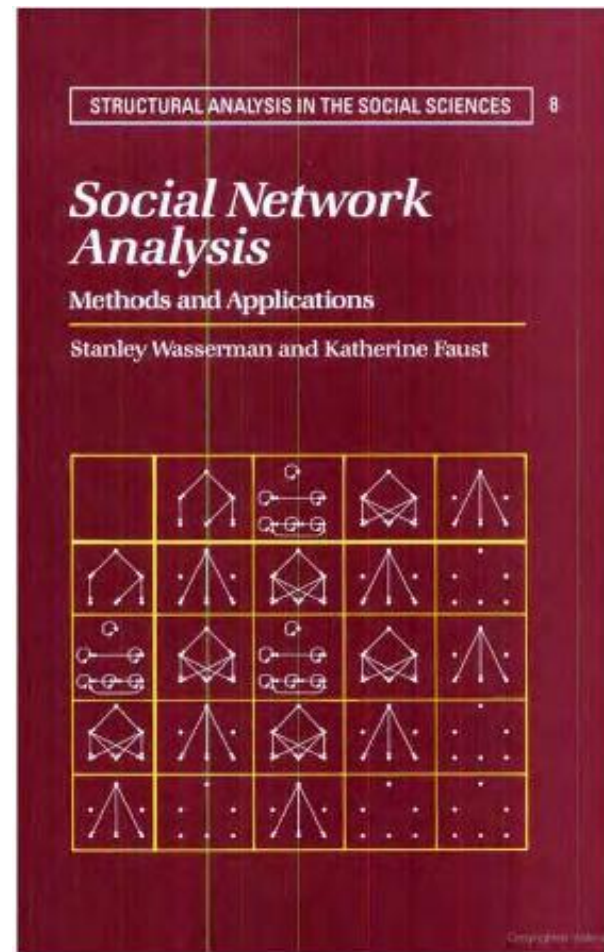
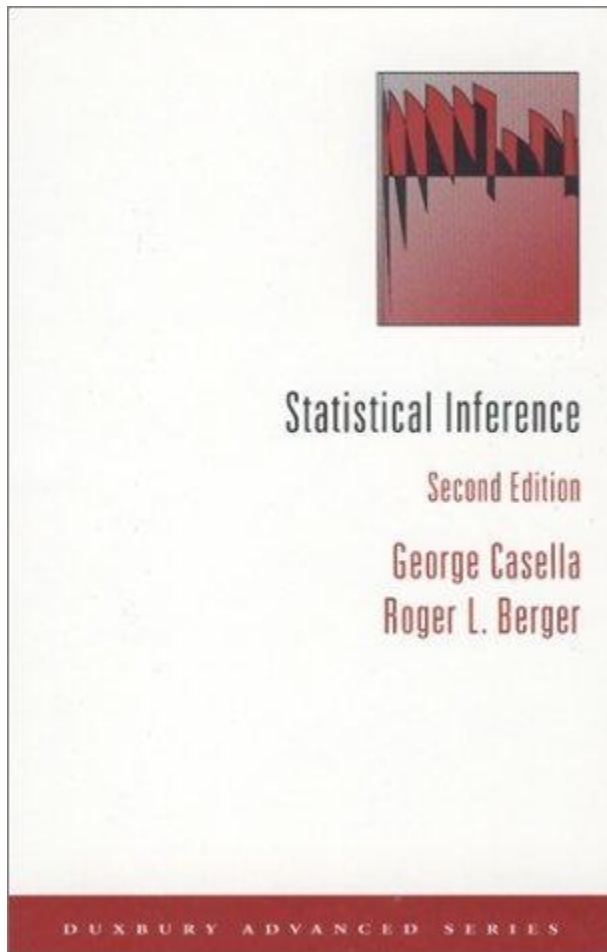
Algorithms



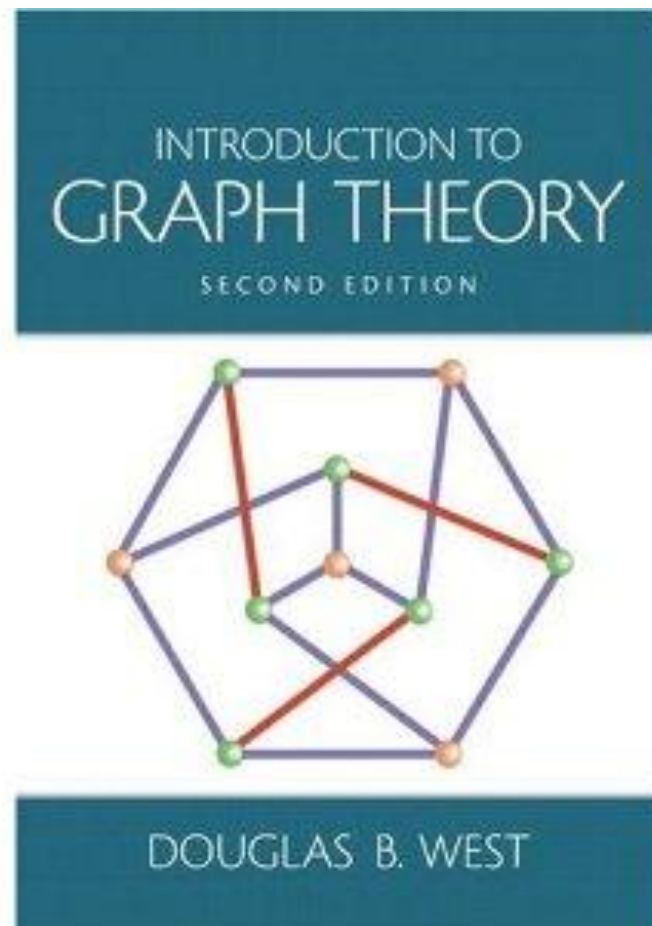
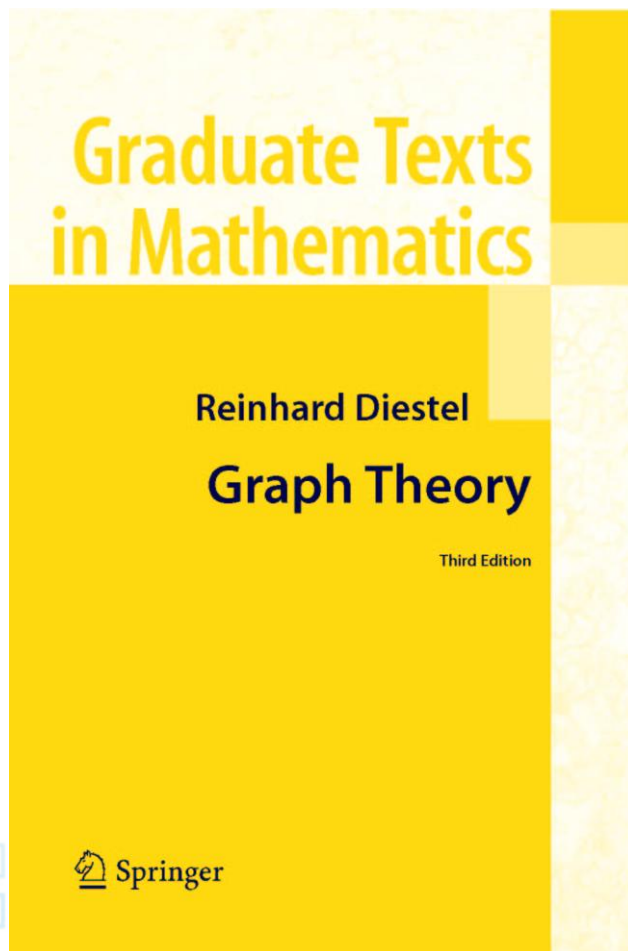
Formal Languages



Statistics and Social Networks



Graph Theory



谢谢