





算法设计与分析

骆吉洲
计算机科学与工程系





课程介绍



Information about Instructor


- Instructor: 骆吉洲
- Room: 诚意楼A407
- Office: 综合楼806室
- Telephone: (0451)86403492-806(×)
- Email: luojizhou@hit.edu.cn(√)



Information about Class

- Class News & class Materials

QQ群: 249114247 (招募管理员)
- Homework and Exam.
 - Homework and Reading: 30%
 - Final Exam (Written Test): 70%



课程的整体目标

- 简单依赖程序设计，能否高效解决问题

例1. 计算两个 n 位十进制整数 a, b 的乘积 $a \times b$

➢ 算法1: 将 a 连续相加 b 遍
 $a + a + \dots + a$ (b 个 a 相加)

➢ 算法2: 小学生竖式计算法

哪个算法更快一些?

$n=10$ 时, 计算器将战胜超级计算机

$n=15$ 时, 手工将战胜超级计算机

	1 2 3
× 4 5 6	
7 3 8	
6 1 5	
+ 4 9 2	
5 6 0 8 8	

计算效率取决于算法选择，而非计算机硬件



课程的整体目标

例2



高维数据特征匹配
 根据目标对象的部分特征，从数据库
 中找出符合特征的对象

企业方法
 50亿条数据，9台服务器，3秒

新方法
 60亿条数据，普通单机，0.377秒

- 人员/车辆轨迹分析
- 同行/人/车
- 套牌车
- 落脚点分析

计算效率取决于算法选择，而非计算环境

HIT CS&E

课程的整体目标

- 简单依赖程序设计，能否高效解决问题

例3. 从海量数据中找出符合指定特征的数据

特征 **数据**

1. *周? 康* 500 0000网页流

2. *淘宝*

...

30000个正则表达式列表

➢ 算法1: 循环检查每个数据是否满足每个特征
20台计算机上的Hadoop计算平台, 超过半小时

➢ 算法2: 合理设计算法
一台笔记本电脑, 32秒

计算效率取决于算法选择, 而非计算环境

HIT CS&E

课程的整体目标

- 算法的概念
- 算法的性能分析
- 典型的算法设计方法
- 高效率求解计算问题的思维
- 设计高效算法的能力

➢ 经典算法积累

➢ 分析问题特征

➢ 合理利用算法设计技术

HIT CS&E

图灵机

脑--> $q \in Q$ 有穷状态控制器 <--- CPU

笔--> 读写头 <--- 总线

纸--> $S \ a_1 \ a_2 \ a_3 \ \dots \ a_n \ B \ \dots$ 读写带 <--- 内存

计算步骤 $\delta(q, a) = (q', b, L/R)$ 确定的TM <--- 计算步骤

一个图灵机就是一个程序

计算问题不同, 图灵机就不同

存在模拟所有程序运行的通用图灵机, 其硬件实现即**计算机**

HIT CS&E

例

实现二进制计数器加1的图灵机

$q \in \{q_0, q_1, q_2\}$

$S \ 0 \ 1 \ 0 \ 1 \ 1 \ B \ \dots$

$f(n) \leq 2n$

用图灵机求解问题

确定问题求解过程的所有状态
状态间的转移关系

与编程求解问题非常相似

$\delta(q_0, S) = (q_0, S, R)$
 $\delta(q_0, 0) = (q_0, 0, R)$
 $\delta(q_0, 1) = (q_0, 1, R)$
 $\delta(q_0, B) = (q_1, B, L)$
 $\delta(q_1, 1) = (q_1, 0, L)$
 $\delta(q_1, 0) = (q_2, 1, L)$
 $\delta(q_1, S) = (q_2, S, R)$

HIT CS&E

图灵机

$q \in Q$ 有穷状态控制器

读写头

$S \ a_1 \ a_2 \ a_3 \ \dots \ a_n \ B \ \dots$ 读写带

$\delta(q, a) = (q', b, L/R)$ 确定的TM

$\delta(q, a) = (Q', b, L/R) \quad Q' \subseteq Q$ 非确定TM

输入规模 n : 问题输入数据占用存储单元的数量 n

计算复杂性 $f(n)$: 操作步骤数量表达成 n 的函数

空间复杂性 $g(n)$: 问题求解过程使用的存储单元最大数量

$P = \{\text{存在确定图灵机使得复杂性}(f(n)) \text{ 是 } n \text{ 的多项式}\}$

$NP = \{\text{存在非确定图灵机使得复杂性}(f(n)) \text{ 是 } n \text{ 的多项式}\}$

$P \subseteq NP$

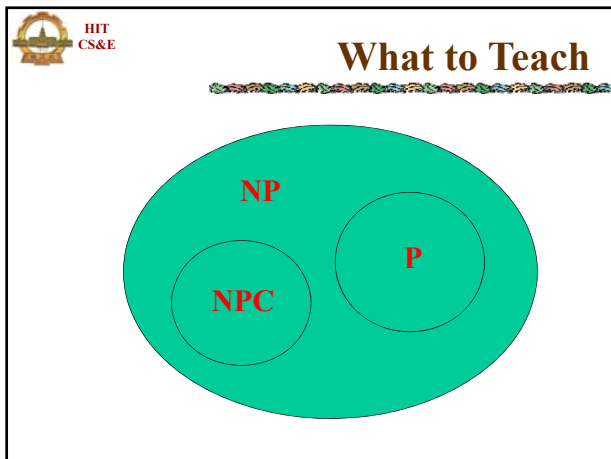
HIT CS&E

P=?NP

NPC-也称NP完全问题

- NP中的一些问题构成的集合
- 只要其中一个问题存在多项式时间算法, 则 $NP=P$

美国麻省Clay数学研究所悬赏100万美元求解 $P=?NP$



HIT CS&E

课程大纲

- 第一章. 绪论 (2学时)
- 第二章. 数学基础 (2学时)
- 第三章. 分治算法 (3学时)
- 第四章. 动态规划 (3学时)
- 第五章. 贪心算法 (3学时)
- 第六章. 平摊分析
- 第七章. 搜索策略 (3学时)
- 第九章. 随机算法 (4学时)
- 第十章. 近似算法 (10学时)
- 第十一章. 在线算法 (2学时)

HIT CS&E

References

1. 算法设计与分析, 骆吉洲, 机械工业出版社, 2014
2. Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*, The MIT Press, 第二版, 2002.
3. Sara Baase. *Computer Algorithms: introduction to design and analysis*. Pearson education press. Third Edition. 1999.
4. 计算机算法设计与分析, 王晓东, 电子工业出版社, 2001

HIT CS&E

Good Books

1. D. E. Knuth等, *Art of the Computer Programming, Vol. 3*, Addison-Wesley, 1973.
2. A.V.Aho, J. D. Ullman等. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.
3. A.V.Aho, J.D.Ullman 等. *Data Structures and Algorithms*. Addison-Wesley, 1983.4.
4. S. Baase, *Computer Algorithms: Introduction to Design and Analysis*. Addison-Wesley, second edition, 1988.
5. E. Horowitz and Sartaj Sahni. *Fundamentals of Computer Algorithms*. Computer Science Press, 1978.

HIT CS&E

Important Journals

1. IEEE Transactions on Electronic Computers
2. IEEE Transactions on Software Engineering
3. IEEE Transactions on Data and Knowledge Engineering
4. Acta Informatica
5. SIAM Journal on Computing
6. Journal of Computer and System Sciences
7. Communication of the ACM
8. Journal of the ACM
9. BIT

HIT CS&E

10. Information and Control
11. ACM Computing Surveys
12. Mathematics of Computation
13. Information Processing Letters
14. Teoretical Computer Science



Important Conferences

- 1. Annual ACM Symposium on Theory of Computing**
- 2. Annual IEEE Symposium on Foundations of Computer Science**
- 3. ACM Annual Computer Science Conference**
- 4. Annual Symposium on Computational Geometry**
- 5. ACM Symposium on Parallel Algorithms and Architectures.**