算法设计与分析

- LECTURE 1

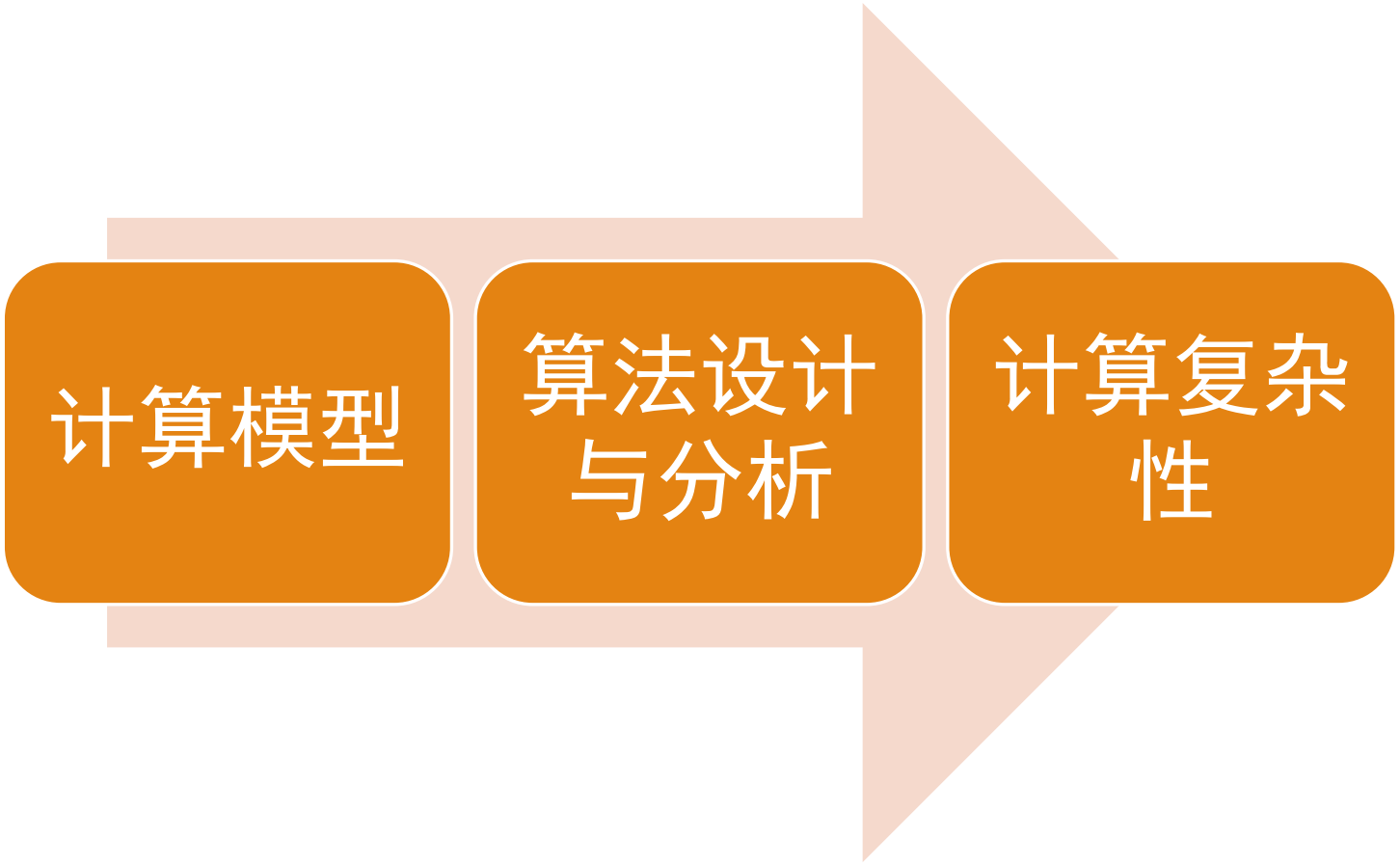
Lecture 1

准备知识

Outline

- 课程介绍
  - 课程大纲
  - 教材
- 抽象的算法设计与分析
  - 计算模型
  - 抽象算法设计
  - 抽象算法分析

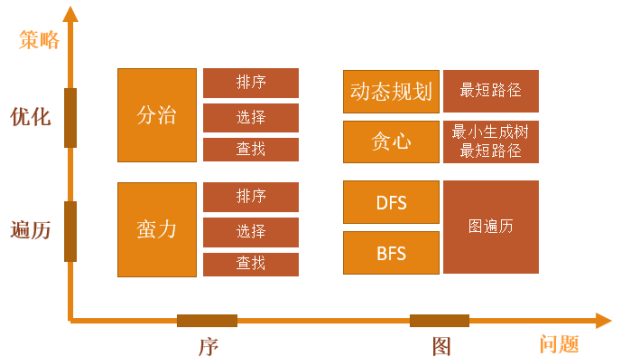
课程大纲



课程大纲

课程大纲

计算复杂性



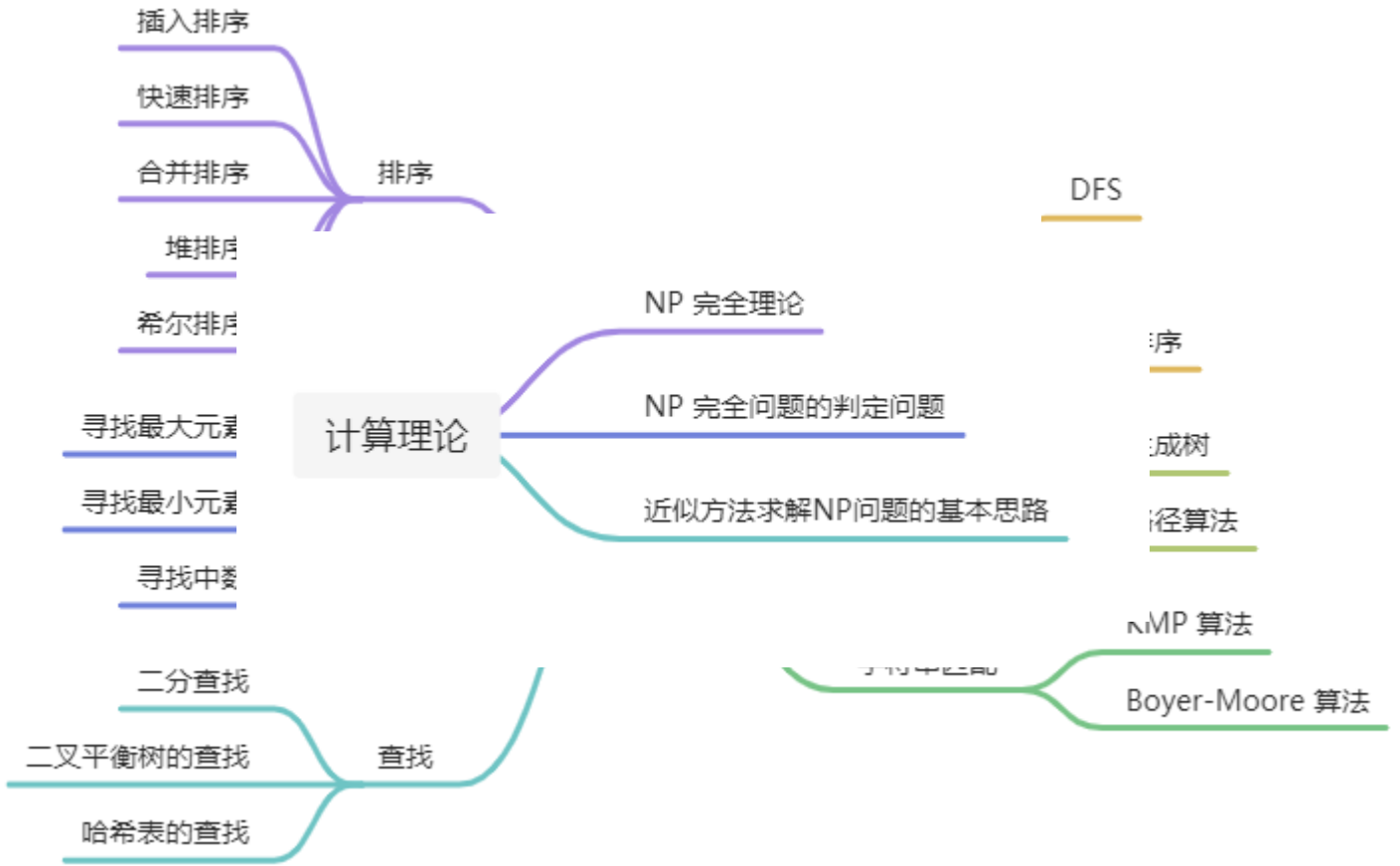
从遍历到优化

分析策略

数据结构

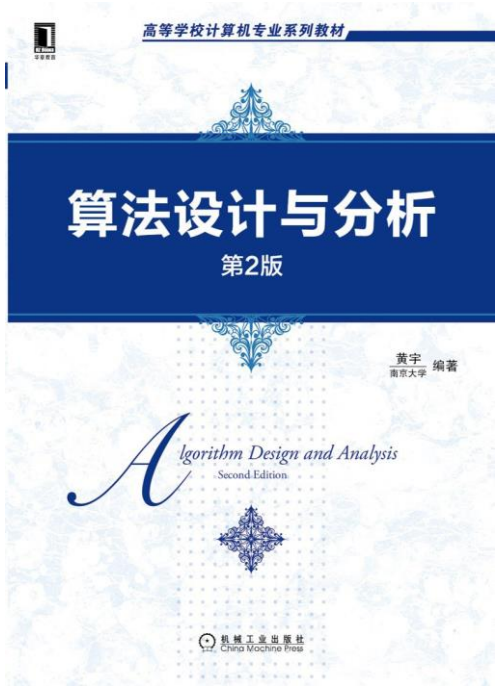
计算模型

课程大纲


插入排序
快速排序
合并排序
堆排序
希尔排序
排序
寻找最大元素
寻找最小元素
寻找中数
二分查找
二叉平衡树的查找
哈希表的查找
查找
计算理论
NP 完全理论
NP 完全问题的判定问题
近似方法求解NP问题的基本思路
DFS
序
成树
径算法
MP 算法
Boyer-Moore 算法

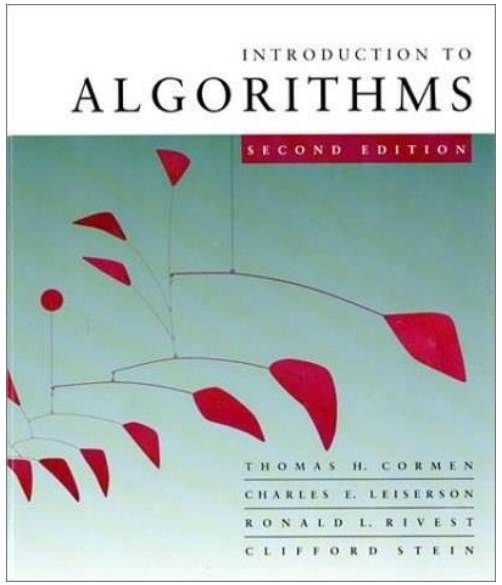
教材

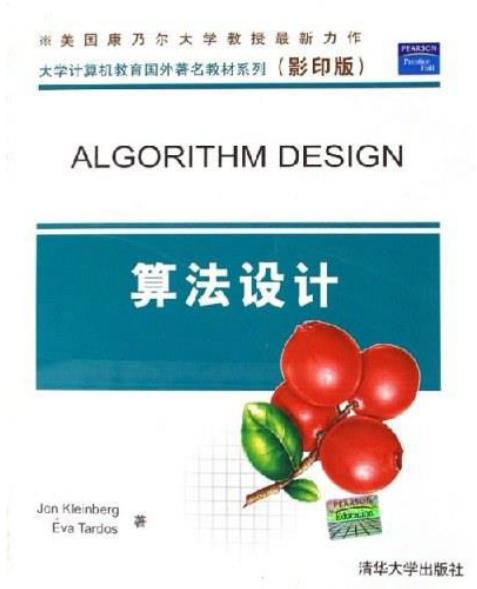
- 黄宇. 算法设计与分析（第2版）. 机械工业出版社.

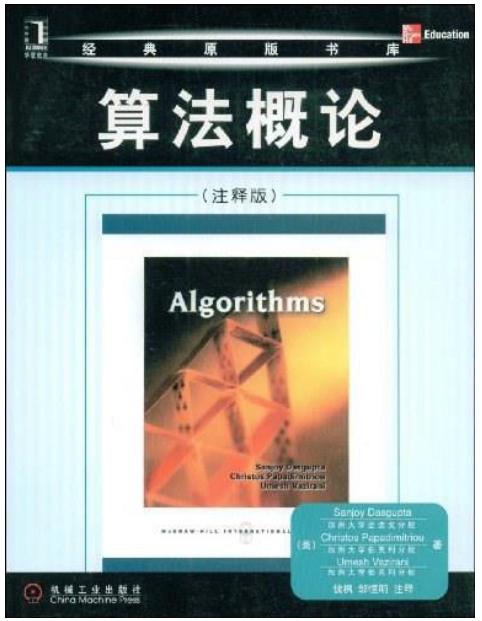


参考书

- Introduction to Algorithms
- Algorithm Design
- Algorithms







Lecture 1

抽象的算法设计与分析

Outline

- 算法-计算的灵魂
  - 计算模型
- 算法示例
  - 最大公约数
  - Sequential search
- 抽象算法设计与分析
  - 算法正确性
  - 最坏/平均情况代价分析

什么是计算？

问题1：

- 为什么计算机似乎无所不能？
  - 科学计算、文档处理、计算机游戏、电影、电子书……

什么是计算？

问题2：

- 什么事情对计算机来说是困难的？
  - 谱曲、写作

计算的本质

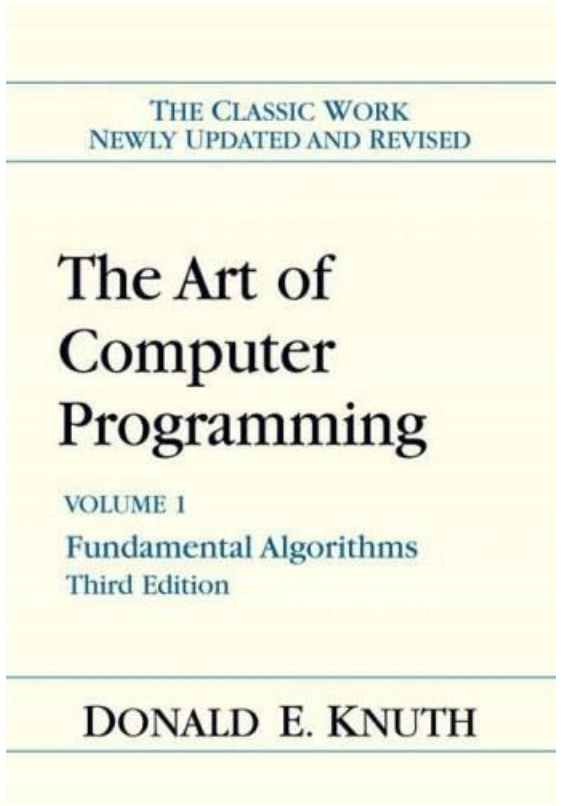
计算：

- 0-1编码
- 操作
- 0-1解码

计算的本质

计算模型

- 算法是计算的灵魂
  - 解决一个特定的问题
  - 基本操作的组合
    - in a precise and elegant way

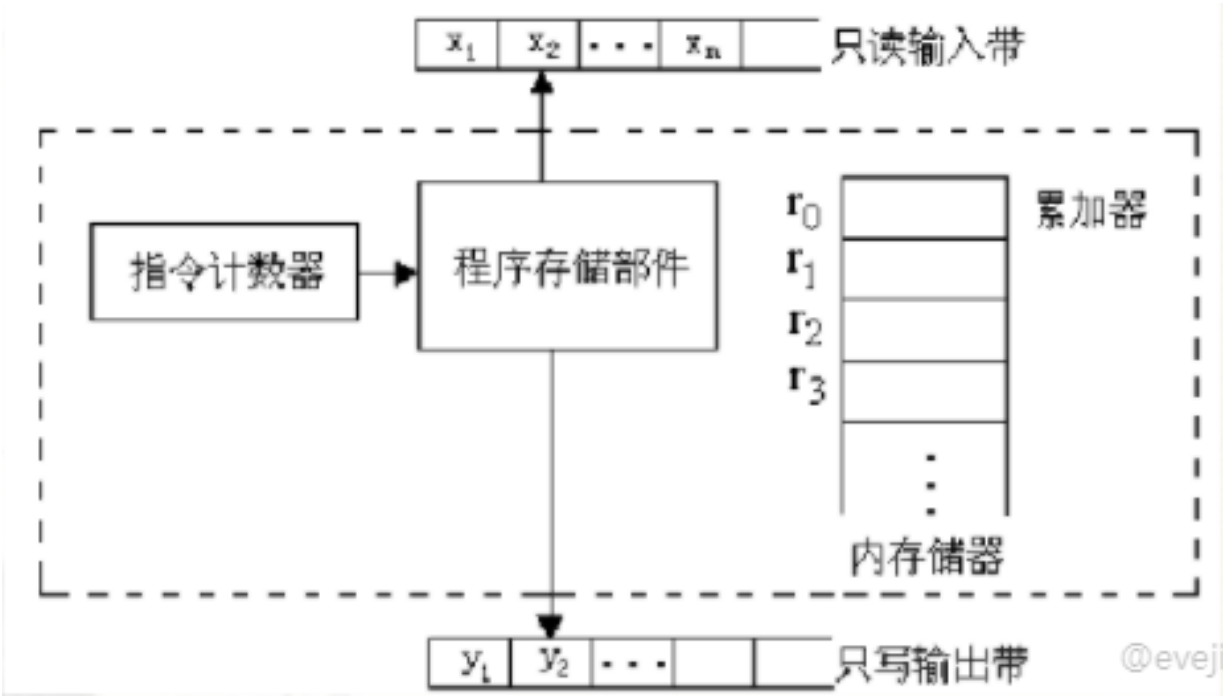


计算模型

- 问题
  - 为什么我们学的算法可以在任意机器上执行？
  - 为什么我们学的算法可以用任意语言实现？
- 与机器、实现语言无关。
- 抽象的机器—计算模型
  - 图灵机
  - RAM 模型

RAM 模型

- RAM 模型是一般计算工具的简化与抽象
- 使我们可以独立于具体的平台，对算法的效率做出可信的比较与评判



RAM 模型

- 简单操作
  - 比较、+/-、存储访问、……
- 复杂操作
  - 循环
  - 子过程
- 存储访问
  - 存储访问是简单操作
  - 不限制内存
- 易用性与精确性的权衡

Outline

- 算法示例
  - 最大公约数
  - Sequential search
- 抽象算法设计与分析
  - 算法正确性
  - 最坏/平均情况代价分析

算法设计与分析

- 算法设计
  - 简单操作的组合，去解决算法问题
- 算法分析
  - 使用的计算量和存储
    - 最坏/平均情况下
  - 更多问题
    - 最优性、近似.....

算法示例

- 算法问题 1
  - 求两个非负整数 a 和 b 的最大公约数
- 算法问题 2
  - 一个元素是否在一个数组里

Euclid Algorithm

- 算法问题规约
  - 输入：明确规定了算法接受的所有合法输入。
  - 输出：明确规定了对于每一组合法的输入，相应的输出值应该是什么。

- 求最大公约数算法问题的规约
  - 输入：任意两个非负整数 a、b
  - 输出：a、b的最大公约数

定理：两个整数的最大公约数等于其中较小的那个数和两数相除余数的最大公约数。

```
1  def EUCLID(a,b):
2      if b=0:
3          return a
4      return EUCLID(b, a mod b)
```

Sequential Search

- 一个元素是否在一个数组里
  - 输入：K, E[1…n]
  - 输出：location of K [1…n]；-1 if not found

```
1  def seqSearch(K, E):
2      for i, v in enumerate(E):
3          if v == K:
4              return i+1
5      return -1
6
```

算法设计

- 标准
  - 定义正确性
- 主要挑战
  - 证明正确性
- 策略
  - 数学归纳法
  - ……

挑战：无穷可能的输入，如何证明算法正确性？

Euclid Algorithm

- 对 b 做归纳
  - Base case
    - b=0: 对任意 a，Euclid(a, 0) = a;
    - b=1: 对任意 a，Euclid(a, 1) = 1;
    - b=2…
  - 假设
    - 对任意 b < k, Euclid(a, b) 是正确的
  - 归纳

$$\text{EUCLID}(a,k) \underbrace{=}_{\text{根据算法实现}} \text{EUCLID}(k, a \bmod k)$$

$$\underbrace{=}_{\text{根据归纳假设}} gad(k, a \bmod k) \underbrace{=}_{\text{根据最大公约数性质}} gcd(a,k)$$

算法分析

- 标准
  - 性能指标
- 最坏情况
  - 最坏情况时间复杂度分析
- 平均情况
  - 平均情况时间复杂度分析

性能指标

- 一般性
  - 必要性指标
- 精确性
  - 机器无关
  - 语言无关
  - 实现无关

性能指标

- 指标
  - 在抽象模型 RAM 上做抽象算法分析
  - 算法本质是让 RAM 模型做若干操作
  - 指标：关键操作的计数
    - 统计简单操作的个数和存储单元的个数

算法问题	关键操作
排序、选择、查找	元素的比较
图遍历	节点信息的简单处理
串匹配	字符的比较
矩阵运算	两个矩阵元素之间的简单操作

算法分析

- 算法复杂度
  - 通常跟输入规模 n 相关
  - 不完全取决于输入规模 n

最坏情况时间复杂度

- 最大代价 W(n)
  - 对于所有可能输入

- $W(n) = \max_{I \in D_n} f(I)$

平均情况时间复杂度

- 平均代价 A(n)
  - 对于所有可能输入的代价加权平均

- $\mathrm{A}(n) = \sum_{I \in D_n} \Pr(I) \cdot f(I)$

Sequential Search

- K is in E[]
  - 假设：
    - K is in E[]
    - 没有相同元素
    - 每个可能的输入等概率出现
  - $A(n) = \sum_{\{i=1\}}^{n} \frac{1}{n} \cdot i = \frac{n+1}{2}$

Sequential Search

- K may (or may not) be in E[]
  - 假设 K is in E 的概率为 q

- $A(n) = q\frac{n+1}{2} + (1-q)n$

算法分析

- Advanced topics
  - 下界分析（选择）
  - 最优性 （Greedy, DP）
  - 计算复杂性
  - 近似/随机/概率算法

Thanks