

算法设计与分析

第1讲：课程概述

2022年9月2日

教师信息

唐斌，教授、博导

- 办公室：勤学楼4207
- Email: cstb@hhu.edu.cn
- 主页: <https://jszy.hhu.edu.cn/tb2>

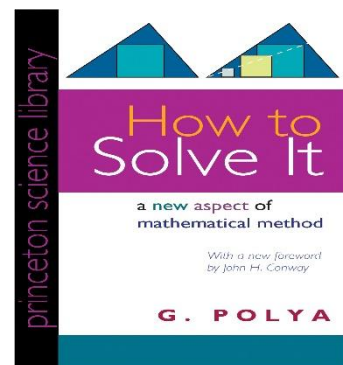
问题1:
解决问题通常包含哪些基本
步骤?

我们如何解题？

- 理解问题：给你提供了什么，以及你想弄清什么
- 设计计划：你将怎么处理问题？
- 执行计划：解决问题
- 回头看：检查结果，然后...



George Pólya



问题2:
计算机能帮我们做什么？

我们如何用计算机解题?

- 计算机如何理解问题?
 - 输入是什么? 输出是什么?
- 如何针对计算机制定计划?
 - 什么样的“计划”可能在计算机上实现目标?
 - 什么样的形式才能让计算机知道该怎么做?
- 执行计划 - “计算机解题”
 - 只有这个才真正是计算机做的!
- 回头看
 - 为什么结果是正确的?
 - 效率能提高吗?



计算
思维

计算思维

美国卡内基梅隆大学教授Jeannette M. Wing(周以真)领导世界上最早的“计算思维研究中心”，并大力推动这一概念。

-----Computational Thinking: What and Why? Link Magazine, 2010

Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent [CunySnyderWing10]

Informally, computational thinking describes the mental activity in formulating a problem to admit a computational solution. The solution can be carried out by a human or machine, or more generally, by combinations of humans and machines.

<https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>

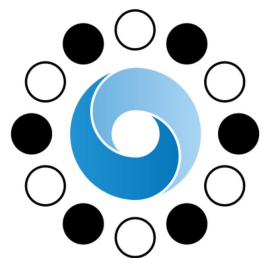
算法课是训练计算思维的重要课程；
算法是计算机解题的关键

算法并不陌生

- 各种排序算法：插入排序、快速排序等等
- Dijkstra最短路径算法
- 最小生成树算法
- ...

问题3：你知道它们的“背后”有什么吗？

Google



AlphaGo

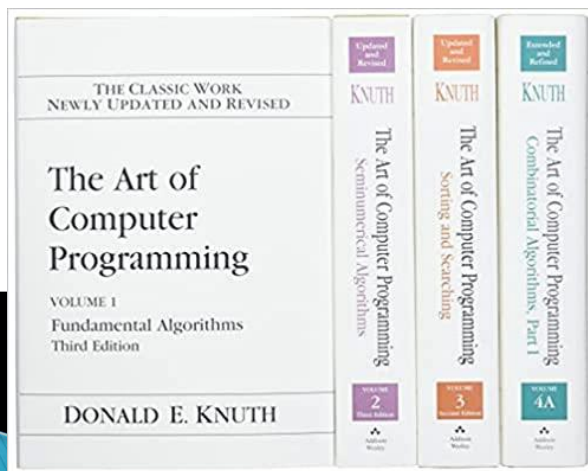
▼ 骑手A、B如果接新订单，他们的新送餐路线和新增送餐距离，如下图所示



算法创新是推动信息技术
发展的一大原动力！

算法是计算机科学的重要主题

- 70年代前，计算机科学基础的主题没有被清楚认清
- 60年代末-70年代初，Don Knuth (高德纳) 出版了《The Art of Computer Programming》前三卷
- 以算法研究为主线
- 确立了算法为计算机科学基础的重要主题
- 1974年获图灵奖（36岁）



算法的定义

- 有限条指令的序列，指令序列确定了解决某种问题^{问题}的运算或操作的步骤，且具有5个重要特征

- 有穷性：执行有限步内必须停止
- 确切性：每一步都必须是精确定义的
- 输入：有0个或多个输入，输入取自某个特定对象的集合
- 输出：有1个或多个输出，输出同输入有着某种特定关系
- 能行性：每一步都是足够基本的，可以被精确地机械执行

问题4：这里的“问题”与日常说的“问题”有什么区别？

- 若仅不满足有穷性，可称为计算方法

欧几里得算法：可能是最古老的算法

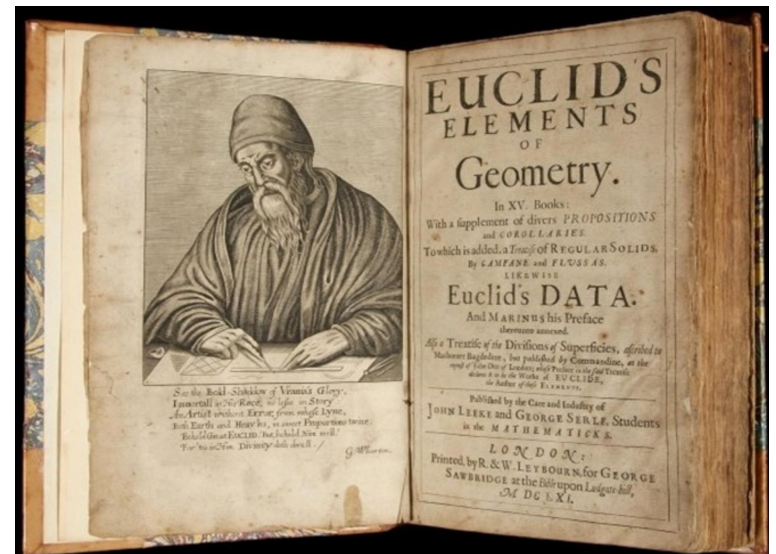
- 用于求两个非负整数的最大公约数

算法Euclid(m, n)

输入：非负整数 m, n ，不全为0

输出： $\gcd(m, n)$

1. **while** $n > 0$ **do**
2. $r \leftarrow m \bmod n$
3. $m \leftarrow n$
4. $n \leftarrow r$
5. **return** m



欧几里得算法

算法Euclid(m, n)

输入：非负整数 m, n ，不全为0

输出：gcd(m, n)

1. **while** $n > 0$ **do**
2. $r \leftarrow m \bmod n$
3. $m \leftarrow n$
4. $n \leftarrow r$
5. **return** m

$$10958 \bmod 1992 = 998$$

$$1992 \bmod 998 = 994$$

$$998 \bmod 994 = 4$$

$$994 \bmod 4 = 2$$

$$4 \bmod 2 = 0$$

$$2 = \text{GCD}$$

Jason Chen's Blog

也叫辗转相除法

欧几里得算法——递归版本

- 输入：非负整数
- 输出：最大公约数

Euclid (**int** m, n)

if $n=0$

then return m

else return **Euclid** ($n, m \bmod n$)

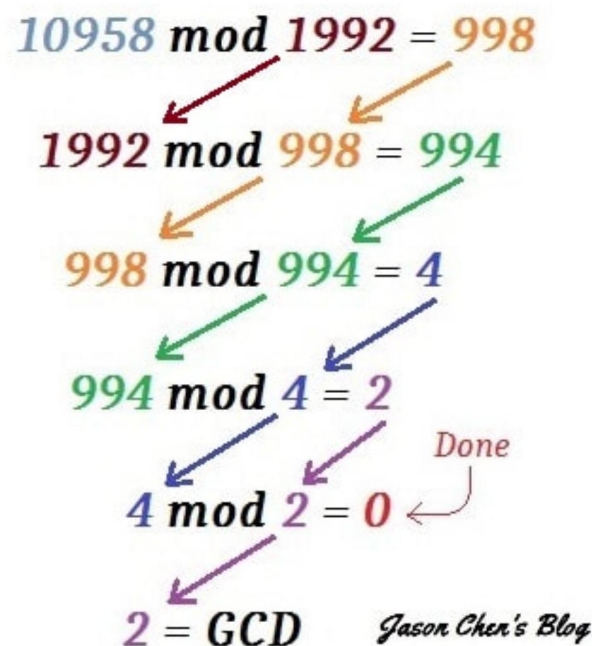
欧几里得算法

算法 $\text{Euclid}(m, n)$

输入：非负整数 m, n ，不全为0

输出： $\text{gcd}(m, n)$

1. **while** $n > 0$ **do**
2. $r \leftarrow m \bmod n$
3. $m \leftarrow n$
4. $n \leftarrow r$
5. **return** m



问题4:

每循环一轮，均会变小！

你能证明欧几里得算法满足有穷性么？

正确性：算法最基本的评价指标

- **正确性：**对每个合法输入，算法均会停止且输出正确的结果
- 不正确算法：在某个输入上不停止，或输出不正确的结果
- 不正确算法不代表没用！例如，
 - 随机算法（有一定概率不停止或返回不正确结果）
 - 近似算法（要求的是最优解，但返回的是接近最优的解）

欧几里得算法的正确性

算法Euclid(m, n)

输入：非负整数 m, n ，不全为0

输出：gcd(m, n)

```
1. while  $n > 0$  do
2.    $r \leftarrow m \bmod n$ 
3.    $m \leftarrow n$ 
4.    $n \leftarrow r$ 
5. return  $m$ 
```

- 有穷性：已证
- 只需输出结果正确

对循环过程的分析：

- 经过一轮循环，
，
- 循环不变量：

欧几里得算法的正确性

证明：设算法总共执行了 n 轮循环，第 n 轮结束的值分别为 a_n, b_n ，分别为输入的和，且满足 $a_n \geq b_n$ ，对于 $n < N$ ，

运用数学归纳法可得

即。注意到循环终止时，算法输出 a_n, b_n ，因此返回结果正确。



如何评价算法运行效率？

麻烦1：算法实际运行时间依赖机器硬件性能等外部因素

■ 解决方案：基本运算次数

麻烦2：算法在不同的问题实例上运行效率不同

■ 解决方案：考虑输入规模，在固定的输入规模下，考虑以下情况下算法所需的基本运算次数

- 最坏情况：基本运算次数最多
- 平均情况：考虑各种可能的输入下算法的平均性能

时间复杂度：算法效率评价指标

- 时间复杂度：针对问题**指定基本运算**，算法所做的基本运算次数
- 最坏情况下的时间复杂度：算法求解**输入规模**为的示例所需要的最长时间
- 平均情况下的时间复杂度：在指定输入的概率分布下，算法求解输入规模为的实例所需要的平均时间

欧几里得算法的时间复杂度分析

■ 基本运算

- 比较操作
- 取余操作 → 真的基本么?
- 赋值操作

■ 输入规模：取决于如何表示

- 1 进制表示：
- 2 进制表示：

算法Euclid(m, n)

输入：非负整数 m, n ，不全为0

输出：gcd(m, n)

```
1. while  $n > 0$  do
2.    $r \leftarrow m \bmod n$ 
3.    $m \leftarrow n$ 
4.    $n \leftarrow r$ 
5. return  $m$ 
```

时间复杂度分析：假设。对于任意整数，如果，那么算法执行的循环次数为。

第个Fibonacci数

算法的其它评价指标

- 空间复杂度：算法运行所需要的存储空间
 - 定义与时间复杂度类似
- I/O，能耗等等

课程主要内容

- 算法分析基础
- 算法设计技术
 - 分治策略、动态规划、贪心算法、回溯与分支限界
- 复杂性理论：刻画问题固有的难度
 - 问题复杂性
 - NP难问题
- NP难问题处理策略

$P=NP?$

课程目标

■ 意识：

- 建立持续追求效率更高、质量更好的算法的创新意识
- 培养计算思维的意识

■ 知识：掌握算法设计与分析的基础知识

■ 能力：

- 具备分析问题，并采用一定策略进行算法设计的能力
- 具备对算法进行基本分析的能力
- 具备自我探索的能力

■ 技能：掌握熟练使用编程语言正确实现算法的能力

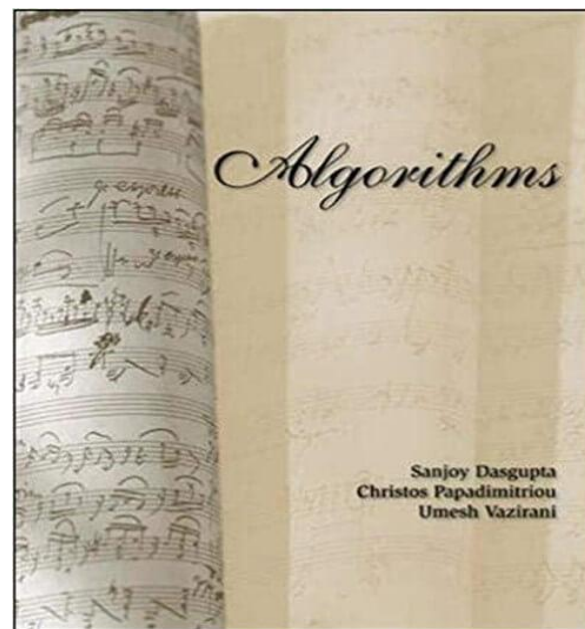
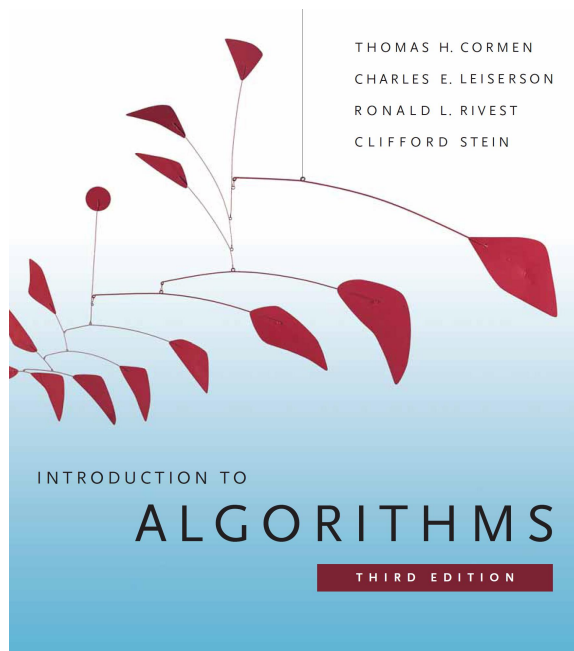
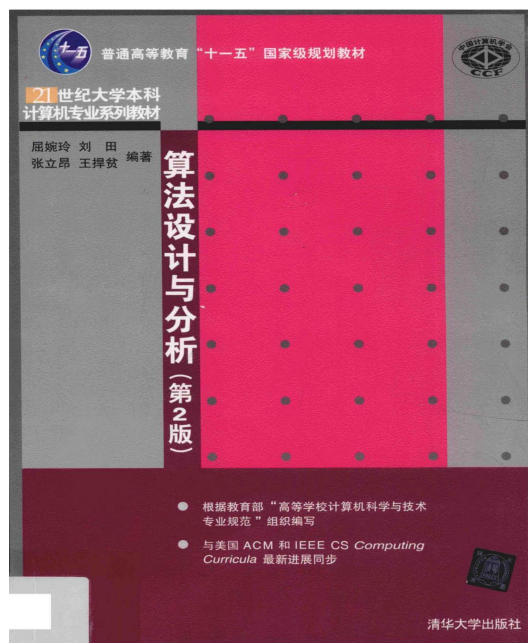
教学基本理念

自我探索
充分引导
理论严密
训练充分

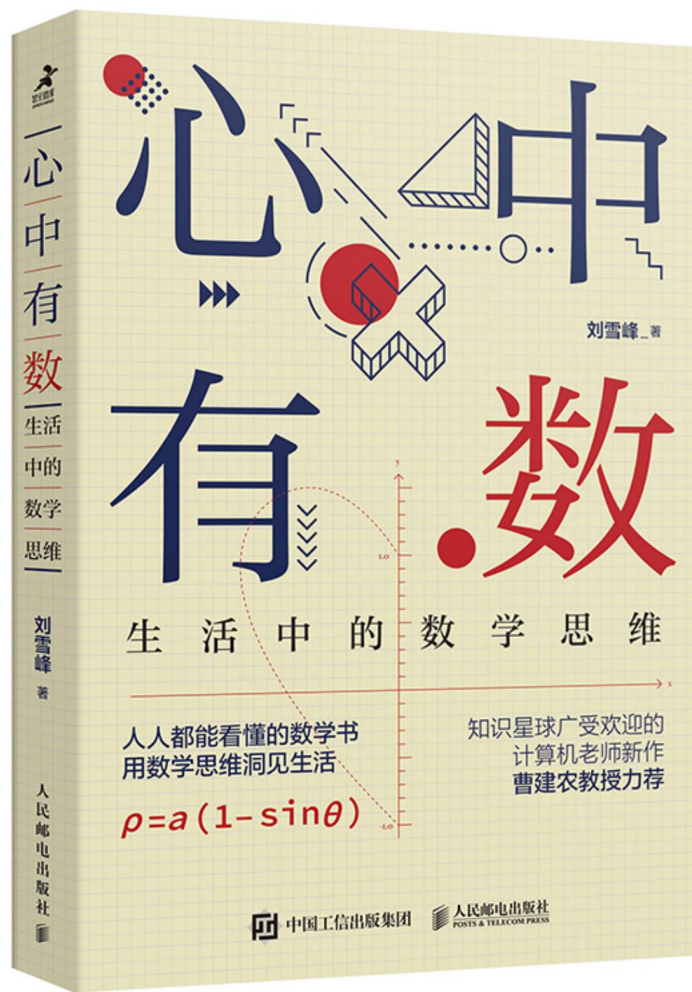
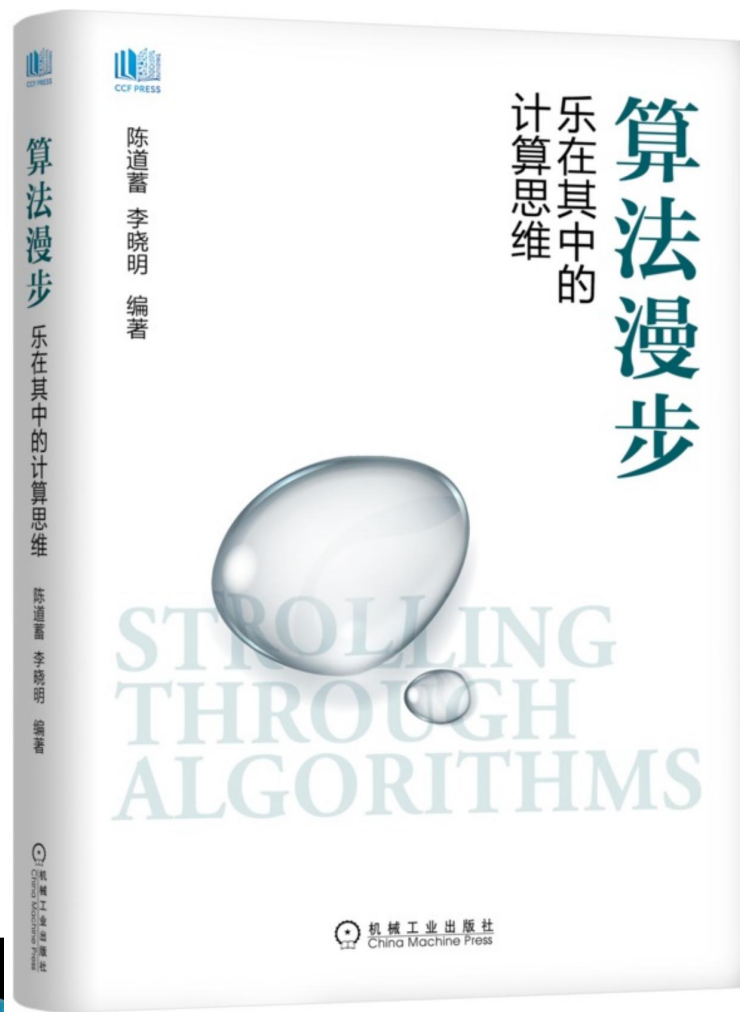
要求和考核方式

- 课前阅读指定的内容
- 作业严禁抄袭！不会做的题目可以写上自己的思考过程
- 根据实际情况决定是否期中考试
 - 如有，平时：20%；期中：30%；期末：50%
 - 如没有，平时：30%；期末：70%

教材与参考用书



推荐课外读物



欧几里得算法的时间复杂度分析

■ 基本运算

- 比较操作
- 取余操作 → 真的基本么？
- 赋值操作

■ 输入规模：取决于如何表示

- 1 进制表示：
- 2 进制表示：

算法Euclid(m, n)

输入：非负整数 m, n ，不全为0

输出：gcd(m, n)

```
1. while  $n > 0$  do
2.    $r \leftarrow m \bmod n$ 
3.    $m \leftarrow n$ 
4.    $n \leftarrow r$ 
5. return  $m$ 
```

时间复杂度分析：假设。对于任意整数，如果，那么算法执行的循环次数为。

第个Fibonacci数

作业

1和2和下次课的作业一起交，3和4自行完成

1. 给出欧几里得算法时间复杂性分析那一页性质的严格证明。
2. 用伪代码描述直接插入排序算法，指出循环不变量，进而证明算法的正确性。
3. (数字华容道游戏中的不变性) 完成

<https://people.csail.mit.edu/meyer/mcs.pdf> Page 164 Problem 5.38。

4. 了解Donald Knuth的一些事迹

下次课前预习第1.3节！