



# 算法设计与分析基础 《Introduction to the Design and Analysis of Algorithms》 算法知识回顾

南京大学软件学院

李传艺

[lcy@nju.edu.cn](mailto:lcy@nju.edu.cn)

费彝民楼917



# 算法是什么？



## ■ 算法

- Wikipedia: In mathematics and computer science, an algorithm is an unambiguous specification of how to solve a class of problems.
  - Calculation, data processing, automated reasoning tasks
  - Expressed within a finite amount of space and time
  - Initial state, initial input, successive states, output and ending state
- 百度百科：算法是指解题方案的准确而完整的描述，是一系列解决问题的清晰指令。
  - 有穷性，确切性，输入项，输出项，可行性

## ■ 表达方式

- 流程图
- 为代码
- PAD图（日本日立公司1973年提出的一种接近于编程语言的算法表示图）



# 算法之于生活



- 例1. 早上醒来一看时间八点二十，刚准备抱怨早起，突然想起有算法课，又要抱怨自己，但又想到第一次课无所谓，反正还没有选，此时核心问题来了：**要不要选算法课？**
  - 选不选课？
  - 解决问题的一般步骤？
- 例2. 躁郁症康复后的小李好不容易找到一份兼职，最后却被一纸测试挡在了面试大门之外。据知情人士透露，是用于评测的电脑将小李的成绩标记为不合格。
  - HR筛选简历前会适用计算机辅助筛选
  - 计算机性格测试
  - 工作绩效考核、保险精算、信用评估
  - 小李通过努力成功在麦当劳获得一份收拾餐桌的兼职
    - 这些算法有待验证



# 生活之于算法（一）



## ■ 遗传算法

- 是模拟达尔文生物进化论的**自然选择**和遗传学机理的**生物进化过程**的计算模型，是一种通过模拟自然进化过程**搜索最优解**的方法。
- 初始种群
- 编码、基因
- 遗传算子：交叉、变异；得到下一代
- 评价、选择
- 终止条件

## ■ 应用例子：云计算环境中考虑能源消耗的工作流调度算法

## ■ 相关研究方向

- **使用遗传算法的机器学习**（散发浓郁的令人心之神往的神秘气息）
- 遗传算法和神经网络、模糊推理、~~混沌~~混沌理论
- 并行遗传算法



# 知识补充



## ■ 模糊推理

- 模糊逻辑(Fuzzy Logic)指模仿人脑的不确定性概念判断、推理思维方式, 对于模型未知或不能确定的描述系统, 以及强非线性、大滞后的控制对象, 应用模糊集合和模糊规则进行推理, 表达过渡性界限或定性知识经验, 模拟人脑方式, 实行模糊综合判断, 推理解决常规方法难于对付的规则型模糊信息问题。

## ■ 混沌理论

- 混沌理论是一种兼具质性思考与量化分析的方法, 用以探讨动态系统中无法用单一的数据关系, 而必须用整体, 连续的数据关系才能加以解释及预测之行为。
- “一切事物的原始状态, 都是一堆看似毫不关联的碎片, 但是这种混沌状态结束后, 这些无机的碎片会有机地汇集成一个整体”
- 古希腊哲学家对于宇宙之源起即持混沌论, 主张宇宙是由混沌之初逐渐形成现今有条不紊的世界。
- 自然规律如地心引力、杠杆原理可以使用数学公式描述, 甚至是星体的运行轨迹



# 补充



## ■ 蝴蝶效应

- 但是很多时候无法预测使用公式准确表达行径的物体的运动情况，因为一些不为人知的因素会导致难以想象的变化
- 如蝴蝶效应
- 西方民谣：
  - 钉子缺，蹄铁卸；蹄铁卸，战马蹶；战马蹶，骑士绝；骑士绝，战事折；战事折，国家灭

## ■ 混沌理论的应用

- 多是对现实的指导意义
- 教育
- 企业管理
  - 企业是开放的，很大程度收到环境的影响
  - 环境是瞬息万变的
  - 用于决策的简单线性因果关系模型已经不再适用



# 补充



- 混沌控制
  - 将此想法化为实用技术，用微小的变化开始，造成希望所想的巨大改变
- 因果理论
  - Causal inference is the process of drawing a conclusion about a causal connection based on the conditions of the occurrence of an effect.
- 因果性和相关性
  - 大数据、数据挖掘、机器学习、混沌理论、模糊推理
  - 精确的因果性？



## 生活之于算法（二）



### ■ 蚁群算法

- 蚁群算法是一种用来寻找优化路径的概率型算法
- 研究蚂蚁觅食的过程中，蚁群整体可以体现一些智能的行为
- 例如，蚁群可以在不同的环境下，能寻找最短到达食物源的路径
  - “信息素”；传递；浓度；一种反馈机制
  - 经过一段时间后，整个蚁群就会沿着最短路径到达食物源了

### ■ 蚁群智能得益于蚂蚁个体的多样性和正反馈

- 多样性类似创新性，不单一重复
- 正反馈使得在正确的基础上创新

### ■ 应用

- 旅行商问题
- 分配问题
- 车间调度问题
- 车辆路由、图着色问题等





# 学习算法的作用



- 一个人接受科技教育的最大收获，是那些能够受用一生的通用智能工具。——George Forsythe
- 算法和生活关系密切
  - 算法之于生活
  - 生活之于算法
- 算法的特性
  - 有穷性，精确性，输入项，输出项，可行性



## 例子. 求最大公约数



### ■ 最大公约数

- 两个不全为0的非负整数 $m$ 和 $n$ 的最大公约数记为 $\gcd(m,n)$ .

### ■ 欧几里得算法伪代码

**Algorithm** Euclid( $m,n$ )

//使用欧几里得算法计算 $\gcd(m,n)$

//输入：两个不全为0的非负整数

//输出： $m,n$ 的最大公约数

**while**  $n \neq 0$  **do**

$r \leftarrow m \bmod n$

$m \leftarrow n$

$n \leftarrow r$

**return**  $m$

### ■ 用于计算 $\gcd(m,n)$ 的连续整数检测法

### ■ 质因数求 $\gcd(m,n)$



# 算法定义

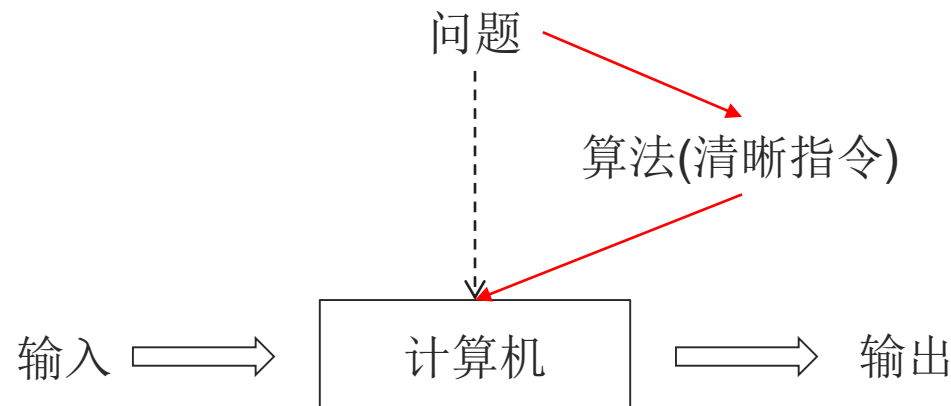


## ■ 理解

- 算法是解决问题的一种特殊方法，不是问题本身的答案，而是经过准确定义的、以获得问题解的过程

## ■ 定义（计算机世界）

- 算法是问题的程序化解决方案，是一系列解决问题的清晰指令，对于符合规范的输入，能够在有限的时间内获得所需要的输出。





# 算法的重要性



- 诗人做学问，功夫在生活、读书：素材累积、表达方式累积
- 程序员做学问，功夫在算法、实践
- 算法指导编程、提高程序质量
- 软件 = 文档 + 程序；程序 = 算法 + 数据结构
- 算法工程师？
- 必须首先是一个优秀的软件工程师，软件工程师又必须懂算法。
  - 本课程：成为合格软件工程师所需要了解和掌握的算法知识



# 如何学好算法知识



/\* \*

\* 学好算法知识的方法

\*/

```
public static Method learnAlgorithmWell(Person p){  
    if ( p.relyOnOthers()){  
        return null;  
    } else {  
        return new Method(p);  
    }  
}
```



# 介绍算法的方案



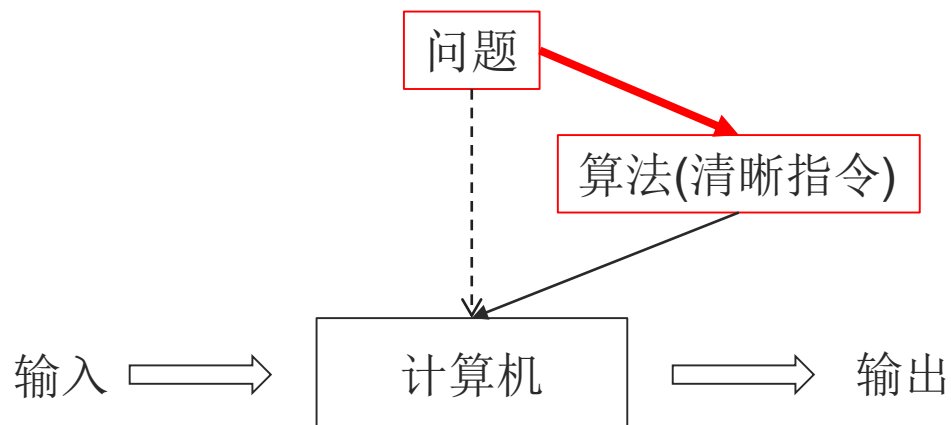
- 从问题类型出发
  - 排序；查找（搜索）；字符串处理；图问题；组合问题；几何问题；数值问题等
  - 优点？——方案对比
  - 不足？——忽略设计
- 从解题思路出发
  - 优点？
    - 注重设计技巧，更加符合应用需求
    - 掌握问题的共性
    - 具体算法的通用设计算法即策略。



# 解决算法问题的一般步骤



- 解决算法问题的“算法”
- 输入：问题
- 输出：算法





## 具体步骤



- 1. 理解问题
- 2. 决定计算方式
- 3. 决定精确还是近似解法
- 4. 使用的数据结构
- 5. 算法的设计策略
- 6. 设计并描述算法
- 7. 算法正确性证明
- 8. 算法分析
- 9. 算法的代码实现





# 数据结构回顾（1）



## ■ 线性数据结构

- 一维数组：连续存储、大小相同、时间相同
- 链表：数据+指针；访问时间不同、快速操作
- 更高级：栈、队列
  - 使用数组、链表实现栈和队列
  - 优先队列

## ■ 图

- $G = \langle V, E \rangle$ ：有向、无向；有环、无环；完全、稠密、稀疏；加权图
- 思考：无向图中两个顶点之间不允许有多条边？
- 表示方法
  - 邻接矩阵，权重矩阵
  - 邻接链表
- 路径和环
  - 路径长度、简单路径、连通性、连通分量、回路、无环图



## 数据结构回顾（2）



### ■ 树

- 自由树、森林、连通分量
- 有根树
- 应用
  - 目录结构、数据存储、数据编码、字典的实现等
- 祖先、真祖先、父节点、子节点、兄弟节点、子孙、真子孙、子树、深度、高度
- 有序树：二叉树、二叉查找树、多路查找树

### ■ 集合与字典

- 互不相同的项的无序组合
  - 检查成员是否存在、并集、交集
- 多重集、包
- 字典：一种基于集合的抽象数据类型



# 算法问题回顾（1）



## ■ 排序问题

- 按照升序对给定列表中的数据项进行排序
- 为什么要排序？已有很多，为什么还要学习和研究排序算法？
- 稳定的、在位的

## ■ 查找问题

- 在给定的集合中查找给定值，该值称为“查找键”
- 需要平衡“查找”、“增加”、“删除”和“修改”操作的效率
- 顺序查找、二分查找、堆查找等

## ■ 字符串处理

- 字符串匹配
- 字符串相似度计算：编辑距离



## 算法问题回顾（2）



### ■ 图问题

- 最短路径、图遍历、拓扑排序
- 旅行商问题
- 图着色问题

### ■ 组合问题

- 寻找一个组合对象，比如一个排列、组合或者一个子集，使得这些对象能够满足特定的条件并具有我们想要的特性
- 优化问题
  - 遗传算法、蚁群算法、粒子群算法等

### ■ 几何问题

- 计算机图形学：图形绘制、阴影计算、图遮挡等
- 最近对问题、凸包问题

### ■ 数值问题

- 解方程、方程组；计算定积分；求函数值等



# 算法效率分析回顾



## ■ 算法效率分析框架

- 时间效率：多快
- 空间效率：额外空间
- 输入的规模：一般越大时间、空间效率越低
- 分析步骤
  - 输入规模度量：表示要处理的单元个数
  - 时间度量的单位：基本操作的次数
  - 增长情况：当输入规模增长时，执行时间的变化情况
  - 最优、最差、平均效率
  - 效率相关的度量符号
    - $O$ ,  $\Omega$ ,  $\Theta$
  - 比较两个算法的效率
    - 平均效率比较、增长次数比



# 算法类型回顾



- 通过实现具体算法的做法来确定算法的类型（解决问题的方案）
- 递归
  - $F(n) := F(n-1) * OPs$  ——递推关系
  - 停止（初始）条件
  - 递归调用树
- 非递归
  - 用循环解决问题；一般要加上数据结构
- 经验
  - 用递归的思维理解问题、分析问题
  - 用递归给出基本的解决方案：递归的开销大？
  - 尽力用循环+设计的数据结构改造原方案



# P、NP、NP完全等问题类型回顾（1）



- P问题
  - 能够在多项式时间内求解的判定问题
- 判定问题
  - 答案是“是”“否”的问题
  - 排除了解空间是非多项式表达的那些问题
  - 包括了那些求解最优解的问题（解空间大但是要的只是最优解）
- 所有的判定问题都是多项式时间内能解决的吗？
  - 否
- “停机问题”
  - 给定一个程序P和它的输入I，判断P在处理I是会终止还是会永远会计算下去
- 判定问题
  - 多项式问题
  - 难解问题——不确定是否存在多项式类算法解决的问题
  - 无解问题



## P、NP、NP完全等问题类型回顾（2）



- 判定问题→难解问题（不能确定是否存在多项式级别的解）
  - 哈密顿回路：所有点一次
  - 旅行商问题：N个点一次最短距离（最短哈密顿回路）
  - 背包问题：将多个物品放入一个背包，最多放多少个
  - 划分问题：N个正整数划分成两个子集，和相等
  - 装箱问题：将一批物体放入固定大小的箱子，最少要多少箱子
  - 图着色问题：最少多少颜色使相邻颜色不同
  - 整数线性规划问题等：线性函数在约束条件下的最大值或最小值
- 共同点？
  - 计算规模按照输入规模呈指数增长
  - 虽然不能求所有解，但是可以快速的判断一个解是否是解空间中的
    - 多项式时间内判断
    - 旅行商问题如何判断解是否是最短的？





## P、NP、NP完全等问题类型回顾（3）



- 不确定算法
  - 猜测阶段：即非确定阶段，生成一个任意的可能的解，作为候选
  - 验证阶段：确定的阶段，判定候选解是否是真实解
- 不确定多项式类型算法
  - 那些验证阶段属于多项式类型算法的不确定算法
- NP类问题
  - 能够使用不确定多项式类算法解决的问题
- NP完全问题
  - 首先是一个NP问题
  - 其他NP问题能够在多项式时间内化简为该问题
- 如果找到任何一个NP完全问题的多项式解，则 $P=NP$



# 总结



- 算法和生活
  - 算法重要性
  - 解决算法问题的一般步骤
  - 数据结构
  - 算法问题
  - 算法效率
  - 算法类型——解决方案角度
  - 问题类型——解决方案的效率角度
- 
- 下节课：基于排序问题的解决方案介绍算法设计的几种策略



谢谢！