

## 联系方法

姓名：任庆生

办公室：3-523

E-mail: [ren-qs@cs.sjtu.edu.cn](mailto:ren-qs@cs.sjtu.edu.cn)

# 第一章

## 引论

# 无处不在的算法

求解数学问题（如寻找最大公约数）的一个过程，该过程步骤有限，通常还涉及重复的操作

广义地说，算法是按部就班解决一个问题或完成某个目标的过程。

狭义的算法特指用于规定计算机如何完成给定任务的过程

算法被公认为计算机科学的基石

## ARTICLE

doi:10.1038/nature24270

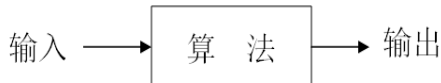
### Mastering the game of Go without human knowledge

David Silver<sup>1\*</sup>, Julian Schrittwieser<sup>1\*</sup>, Karen Simonyan<sup>1\*</sup>, Ioannis Antonoglou<sup>1</sup>, Aja Huang<sup>1</sup>, Arthur Guez<sup>1</sup>, Thomas Hubert<sup>1</sup>, Lucas Baker<sup>1</sup>, Matthew Lai<sup>1</sup>, Adrian Bolton<sup>1</sup>, Yutian Chen<sup>1</sup>, Timothy Lillicrap<sup>1</sup>, Fan Hui<sup>1</sup>, Laurent Sifre<sup>1</sup>, George van den Driessche<sup>1</sup>, Thore Graepel<sup>1</sup> & Demis Hassabis<sup>1</sup>

## 算法基本性质

无论是广义的算法还是狭义的算法，都应满足以下4条性质：

- 有零个或多个输入；（算法可以没有输入）
- 至少有一个输出结果；（算法必须要有输出）
- 构成算法的步骤必须清晰明确；
- 算法能够在有限时间内完成。



## 这是一个算法吗

输入：自然数 $a$ 、 $b$

输出： $a$ 、 $b$ 的最大公约数

步骤：

- 1) 令 $r$ 为 $a / b$ 所得余数，即 $r = a \% b$
- 2) 若 $r = 0$ ，则算法结束， $b$ 即为答案；  
否则置 $a = b$ ， $b = r$ ，转到第1步。

输入：青菜，香菇，油，盐

输出：香菇炒青菜

步骤：

- 1) 青菜洗净掰开。
- 2) 香菇洗净切小片。~~不够清晰明确、~~
- 3) 锅内放油，将香菇煸炒半分钟。
- 4) 倒入青菜大火煸炒。
- 5) 青菜变软的时候即可调入适量盐，翻炒均匀即可。

## 算法与编程

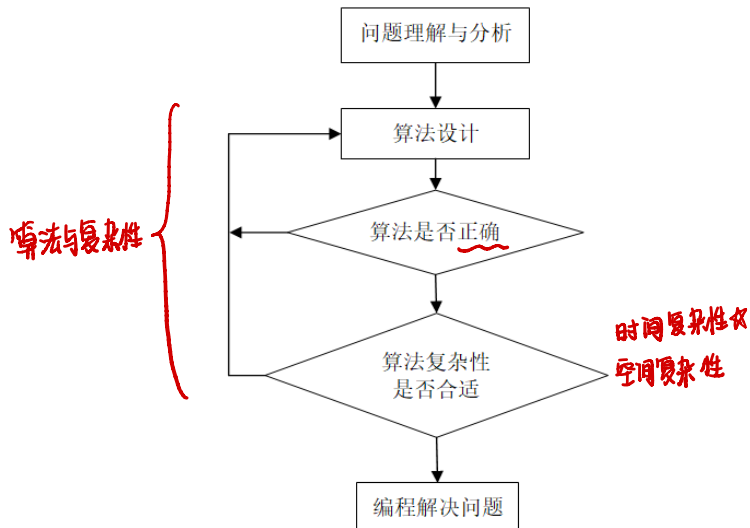
一台计算机可接受的指令应是定义明确、长度有限的基本操作序列。将通常的命令转换为计算机可以理解的指令是一个困难的过程，该过程就是编程。

程序是算法用某种程序设计语言的具体体现，可以在计算机上运行，而且还可以不满足算法的第四条限制，即时间有限性。

例：银行ATM机上运行的程序就是一个在无限循环中执行的程序，不会自动结束。但是其执行的各项任务都是由系统中的一个子程序根据某个特定算法实现的，如查询、取款等，每一个子程序得到结果后便会终止。

算法 → 时间有限。  
程序 → 可以无限循环

# 基于计算机的问题求解过程



Programming  
=algorithm+data structure+language



## 参考书

M H Alsuwaiyel, Algorithm design technique and analysis, World Scientific, 1999

Alfred V Aho, John E Hopcroft, Jeffery D Ullman, The design and analysis of computer algorithms, Addison-Wesley, 1974

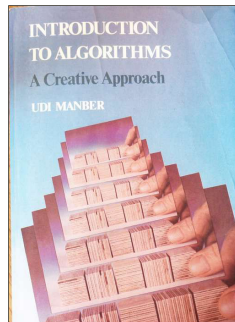
Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, Clifford Stein, Introduction to algorithms (second edition), MIT, 2001

.....

# 教材

Udi Manber, Introduction to Algorithms: A Creative Approach

算法引论：一种创造性方法，电子工业出版社，2005



## 教材的特点——数学归纳法

核心：对证明数学定理的过程和设计算法的过程进行类比。

数学归纳法原理：若对于相对较小的命题实例成立可以导出较大的命题实例成立，  
则对所有实例来说命题都是成立的

算法设计的基本思想：从简单到复杂

算法设计：如何将简单问题的解扩充为较大问题的解。讨论如何加强归纳假设、如何更精明地选择归纳序列，如何应用双重归纳和反向归纳等。

特点：1) 收集了许多似乎无关的算法设计技术并归类于一个框架之下；2) 将已知的数学证明技术应用于算法设计。

缺点：不是一种通用的方法

## 算法描述——伪代码

求两个自然数a和b的最大公约数(设 $a > b$ )

```
#include <iostream>
using namespace std;
int main()
{
    int a,b,r=1;
    cout<<"Input two positive integer numbers:";
    cin>>a>>b;
    while (r>0)
    {
        r=a%b;
        a=b;
        b=r;
    }
    cout<<"The result is:"<<a<<endl;
    return 0;
}
```

输入：自然数a、b

输出：a、b的最大公约数

步骤：

- 1)  $r=1$
- 2) while  $r > 0$
- 3)  $r = a \% b$
- 4)  $a = b$
- 5)  $b = r$
- 6) endwhile
- 7) 输出a

## 算法描述——伪代码

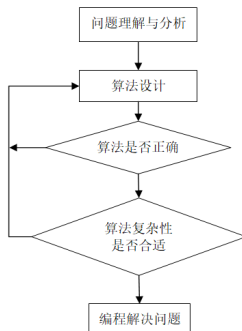
目的：加强算法的形象描述。

未对程序进行太多的优化，也不推荐直接使用它们。在许多情况下有意不包括程序的优化版本，因为这可能会引入额外的复杂性，使注意力离开算法的主要思想。我们有时也不详细解释如何将算法思想转换为程序，这种转换有时是显然的有时却不是。

本书的侧重点在于算法设计的原理。必须了解算法背后的原理，必须知道如何应用它们，当然更重要的是，何时应用它们。

## 如何学好这门课

多看 多练 多想



## 链表节点删除

有一个非空链表，指针p指向节点A，A不是最后一个节点，如何快速地从链表中删去A

单链表

↓  
将A-节点拷贝到A，  
再删除A-节点

## 链表节点删除

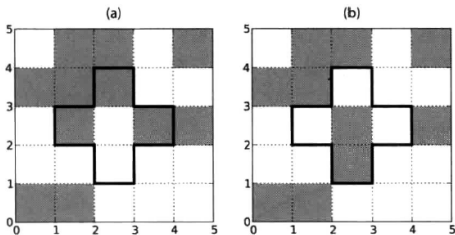
有一个非空链表，指针p指向节点A，A不是最后一个节点，如何快速地从链表中删去A

将A后面的节点的数据项拷贝到A，然后删除A后面的节点

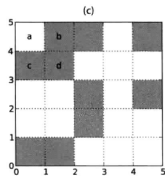


## 格子点灯

5\*5的格子中，每个格子代表一盏灯，点击一次，此格子和上下左右相邻的格子中的灯的状态会发生反转，亮着的会熄灭，熄灭的会点亮。如何用最少的点击次数使所有的灯都点亮？（白色为亮灯，用1表示，黑色为熄灭状态，用0表示）



## 格子点灯



事前分析：点击顺序不重要，每个格子的状态依赖于自身和相邻格子的点击次数  
一个格子没有必要点击两次及以上

结论1：共有225种点击方式

进一步加上条件：

以特定顺序点击，从第一行最左侧格子开始依次判断是否点击，第二行开始则考虑已经点击完成的上一行的状态。以图c为例，格子c点击后a会熄灭并且再也没机会。

结论2：只要确定了第一行的点击方式，其余各行的点击方式就确定了

最后的结论：共有~~25~~<sup>2</sup>种点击方式

平时30% 考试70%

## 内容

第1章 引言          第2章 数学归纳法

第3章 算法分析      第4章 数据结构

☆☆ 第5章 基于归纳的算法设计      第6章 序列和集合的算法 *实际应用*

第7章 图算法          第8章 几何算法

第9章 代数和数值算法      第10章 归约

第11章 NP完全问题      第12章 并行~~算法~~

了解算法      创造算法