

研究的对象：规律、技巧  
研究的目标：高效、低耗

01-A1

绪论

计算工具

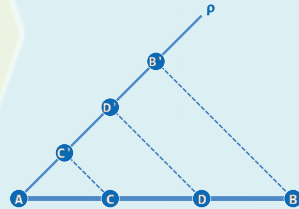
The cubic root of 2 is not constructible by ruler and compass,  
but the cubic root of  $2+\sqrt{5}$ , which looks more complicated, is.  
Things like this make it fun to be a mathematician.

邓俊辉  
deng@tsinghua.edu.cn

## 尺规计算机

任给平面上线段AB (输入)，将其三等分 (输出)

- 从A发出一条与AB不重合的射线p
- 在p上取  $|AC'| = |C'D'| = |D'B'|$
- 联接B'B
- 经D'做B'B的平行线，交AB于D
- 经C'做B'B的平行线，交AB于C



这里的计算机是什么？尺规

它能够解决什么问题？不能解决什么问题？能解决部分复杂几何作图问题

子程序：过直线外一点，做平行线

Data Structures &amp; Algorithms, Tsinghua University

01-A2

绪论

计算：算法

Computer science should be called computing science, for  
the same reason why surgery is not called knife science.  
- E. Dijkstra

邓俊辉  
deng@tsinghua.edu.cn

## 算法

计算 = 信息处理 = 借助某种工具，遵照一定规则，以明确而机械的形式进行

计算模型 = 计算机 = 信息处理工具

所谓算法，即特定计算模型下，旨在解决特定问题的指令序列

输入 待处理的信息 (问题)

输出 经处理的信息 (答案)

正确性 的确可以解决指定的问题

确定性 任一算法都可以描述为一个由基本操作组成的序列

可行性 每一基本操作都可实现，且在常数时间内完成 eg. 大象装冰箱的基本操作

有穷性 对于任何输入，经有穷次基本操作，都可以得到输出

...

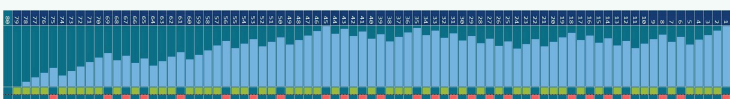
Data Structures &amp; Algorithms, Tsinghua University

## 有穷性：Hailstone序列

$$Hailstone(n) = \begin{cases} \{1\} & (n \leq 1) \\ \{n\} \cup Hailstone(n/2) & (n \text{ is even}) \\ \{n\} \cup Hailstone(3n+1) & (n \text{ is odd}) \end{cases}$$

[42], 21, 64, 32, ..., 1

[7], 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, ..., 1



[27], 82, 41, 124, 62, 31, 94, 47, 142, 71, 214, 107, ...

Data Structures &amp; Algorithms, Tsinghua University

## 有穷性：程序~算法

```
int hailstone( int n ) {
    int length = 1;
    while ( 1 < n ) { n % 2 ? n = 3*n + 1 : n /= 2; length++; }
    return length;
}
```

//对于任意的n, 总有  $|Hailstone(n)| < \infty$  ? 尚无定论 - Hailstone 尚不能称为算法!



Erds: Mathematics is NOT yet ready for such problems

Data Structures &amp; Algorithms, Tsinghua University

## 有穷性：理想硬币~几何分布

假设：rand()为理想的随机整数发生器

于是：rand()为奇、偶的概率均为50%

考查算法程序：

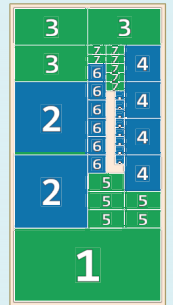
```
void dice( ) { while ( rand() & 1 ); }
```

其中的循环，将迭代多少次？

$Pr(s = k) = (1/2)^{k-1} \cdot (1 - 1/2) = 2^{-k}$

s的数学期望： $E(s) = \sum_{k=1}^{\infty} k \cdot 2^{-k} = 1/(1 - 1/2) = 2$

尽管在理论上可能任意多次 不满足有穷性



Data Structures &amp; Algorithms, Tsinghua University

## 好算法 = ?

正确：符合语法，能够编译、链接

能够正确处理简单的输入

能够正确处理大规模的输入

能够正确处理一般性的输入

能够正确处理退化的输入

能够正确处理任意合法的输入

健壮：能辨别不合法的输入并做适当处理

而不致非正常退出

可读：结构化 + 准确命名 + 注释 + ...

关键：效率：速度尽可能快

存储空间尽可能少

Algorithms + Data Structures = Programs

(Algorithms + Data Structures) x Efficiency = Computation

Data Structures &amp; Algorithms, Tsinghua University

绪论

计算模型：统一尺度

To measure is to know. If you can not measure it,  
you can not improve it.

- Lord Kelvin

洞察一个对象的形式性质，把它同它的质料部分完全分开，然后沉思  
它，然后判断时间，即这一特殊形式的事物自然要持续的最长时间。

邓俊辉  
deng@tsinghua.edu.cn

## 算法分析

两个主要方面...

正确：算法功能与问题要求一致？

数学证明？可不那么简单...

成本：运行时间 + 所需存储空间

如何度量？如何比较？

将计算成本描述为函数，比如...

$T_A(P)$  = 算法A求解问题实例P的计算成本

意义不大，毕竟...

可能出现的问题实例太多

如何归纳简化、概括？

观察：问题实例的规模，往往是

决定计算成本的最主要因素

通常：规模接近，计算成本也接近

规模扩大，计算成本亦上升

Data Structures &amp; Algorithms, Tsinghua University