

1. 绪论

(a) 计算

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

绳索计算机及其算法

❖ 输入：任给直线1及其上一点A

输出：经过A做1的一条垂线

❖ 算法（2000 B.C.，古埃及人）

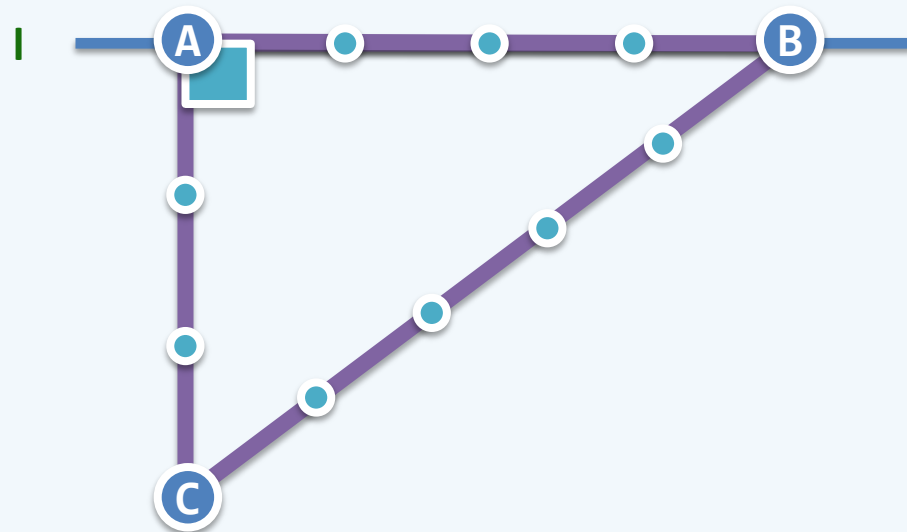
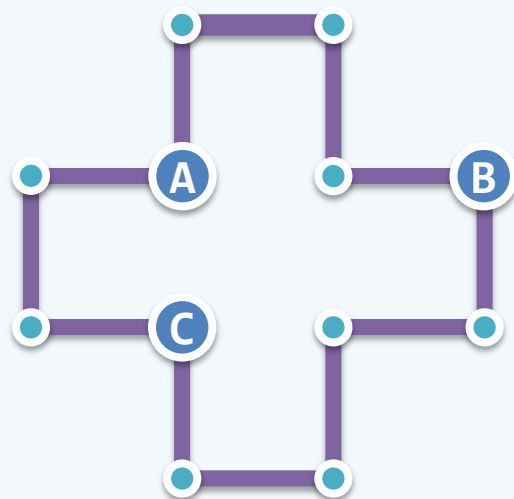
取12段等长的绳索，首尾联接成环

从A点起，将4段绳索沿1抻直并固定于B

沿另一方向找到第3段绳索的终点C

移动点C，将剩余的3+5段绳索抻直

❖ 这里的计算机是什么？



尺规计算机及其算法

❖ 任给平面上线段AB（输入），将其三等分（输出）

❖ 算法：

从A发出一条与AB不重合的射线 ρ

在 ρ 上取 $AC' = C'D' = D'B'$

联接 $B'B$

经 D' 做 $B'B$ 的平行线，交AB于D

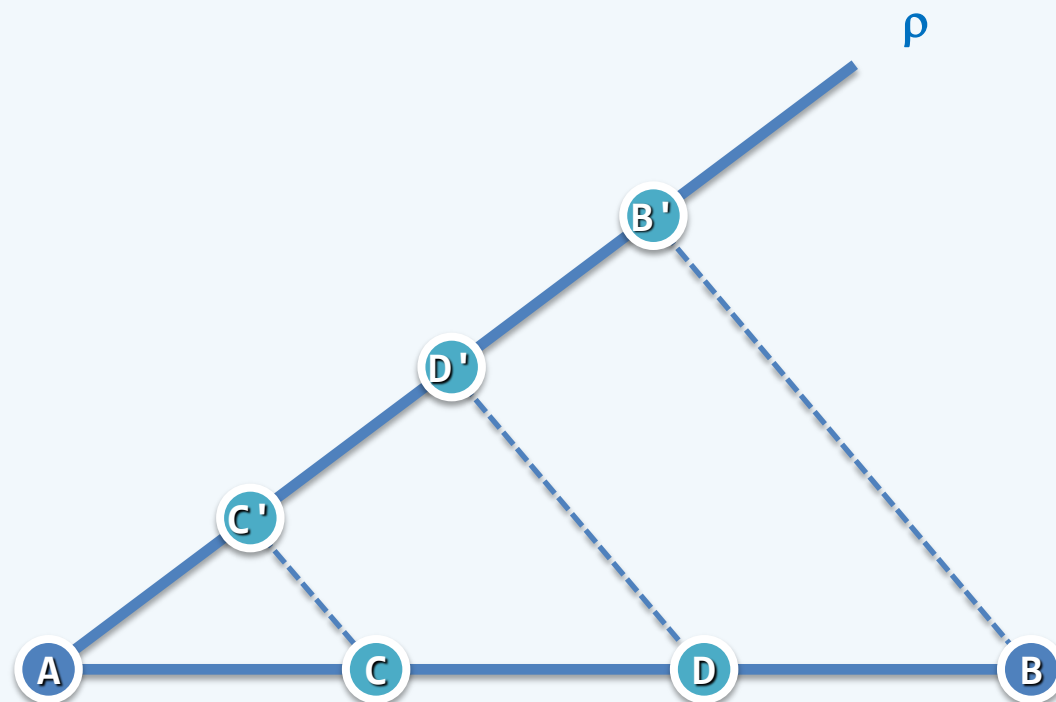
经 C' 做 $B'B$ 的平行线，交AB于C

❖ 这里的计算机是什么？

❖ 它能够解决什么问题？

不能解决什么问题？

❖ 子程序：过直线外一点，做平行线



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

有穷性 对于任何输入，经有穷次基本操作，都可以得到输出

... ..

算法：有穷性

$$\text{❖ 序列 } Hailstone(n) = \begin{cases} \{1\} & n \leq 1 \\ \{n\} \cup Hailstone(n/2) & n \text{ 偶} \\ \{n\} \cup Hailstone(3n+1) & n \text{ 奇} \end{cases}$$

$$\text{❖ } Hailstone(42) = \{ 42, 21, 64, 32, \dots, 1 \}$$

❖ `int hailstone(int n) { //计算序列Hailstone(n)的长度`

`int length = 1; //从1开始，以下按定义逐步递推，并累计步数，直至n = 1`

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

`return length; //返回|Hailstone(n)|`

`}`

$$\text{❖ } Hailstone(7) = \{ 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, \dots, 1 \}$$

$$Hailstone(27) = \{ 27, 82, 41, 124, 62, 31, 94, 47, 142, 71, 214, 107, \dots \}$$

❖ 对于任意的n，总有 $|Hailstone(n)| < \infty$ ？

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 + ...
- ❖ 效率：速度尽可能快；存储空间尽可能少

Algorithms + Data Structures = Programs

//N. Wirth, 1976

(Algorithms + Data Structures) x Efficiency = Computation

为何要学？学什么？学习目标？

- ❖ **数据结构** 在计算机相关专业课程体系中，一直处于核心位置
是计算机科学的重要组成部分
是设计与实现高效算法的基石
- ❖ **讲授范围** 各类数据结构设计和实现的基本原理与方法
算法设计和分析的主要技巧与工具
- ❖ **学习数据结构，就是要学会**
高效地利用计算机，有效地存储、组织、传递和转换数据
掌握各类数据结构功能、表示、实现和基本操作接口
理解各类（基本）算法与不同数据结构之间的内在联系
了解各类数据结构适用的应用环境
灵活地选用各类（基本）算法及对应的数据结构，解决实际问题

❖ 各数据结构的ADT接口及其不同实现

序列（向量、列表、栈、队列），树及搜索树（AVL树、伸展树、红黑树、B-树、kd-树）
优先队列（堆），字典（散列表、跳转表），图的算法与应用

❖ 构造有效算法模块的常用技巧

顺序和二分查找，选取与排序，遍历
模式匹配，散列，几何查找

❖ 算法设计的典型策略与模式

迭代、贪心、递归、分治、减治、试探-剪枝-回溯、动态规划

❖ 复杂度分析的基本方法

渐进分析与相关记号
递推关系、递归跟踪
分摊分析、后向分析