Chapter 1

The role of algorithms in computing

songyou@buaa.edu.cn

1 The Role of Algorithms in Computing

- **What are algorithms?**

- **Why is the study of algorithms worthwhile?**
  (为什么算法研究非常有意义)

- **What is the role of algorithms relative to other technologies used in computers?**
  (相比计算机科学的其他技术，算法的角色和地位)

1.1 What are algorithms

一般说来，从程序员角度说，算法就是伪代码（能用计算机语言实现），或就是某种语言写的程序，目的是科学地解决计算(computing)问题，如： $F(n) = F(n\text{-}1) + F(n\text{-}2)$

A1: recurrence

```
f(n)
{
    if(n <= 2)
        return 1;
    else
        return f(n-1) + f(n-2);
}
```

A2: non-recurrence

```
f1 = 1,  f2 = 1;
for(i = 3; i <= n; i++)
{
        f = f1 + f2;
        f2 = f1;
        f1 = f;
}
```

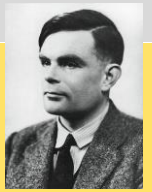
因此，算法又被称为科学计算。

1.1 What are algorithms

- **Computing: information processing**
- **Scientific computing (modeling, computing, verifying)**
  - 数学：公理、规则 => 定理
    - “上帝创造”。描述客观世界，给人们带来无穷乐趣和痛苦

    (例如：Pi的小数点后面是否存在1000个连续的7？若存在，请找出或证明，若不存在，请证明。)

  - 1900, Hilbert (1862~1943), 巴黎世界数学家大会，“是否存在一个通用的过程（算法），可以自动判定任意命题是否正确？”



  - Alan Turing (1912~1954), 1931, undergraduate in Cambridge Univ.
  - Before 1936, no scientific computing. Turing, 论可计算数及其在判定问题中的应用, 1936.
  - John Von Neumann (1903~1957), 1946, first electronic computer.

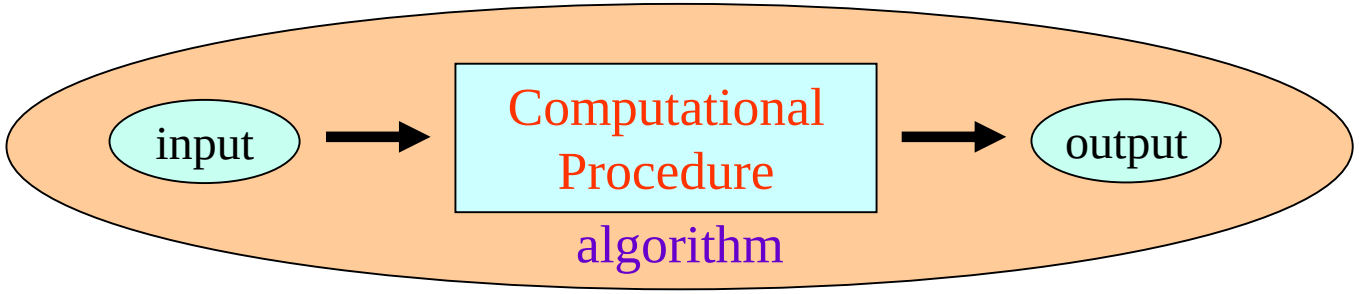
  - 算法：过程、工具
    - “人创造”。方便人们研究数学；解决实际问题。

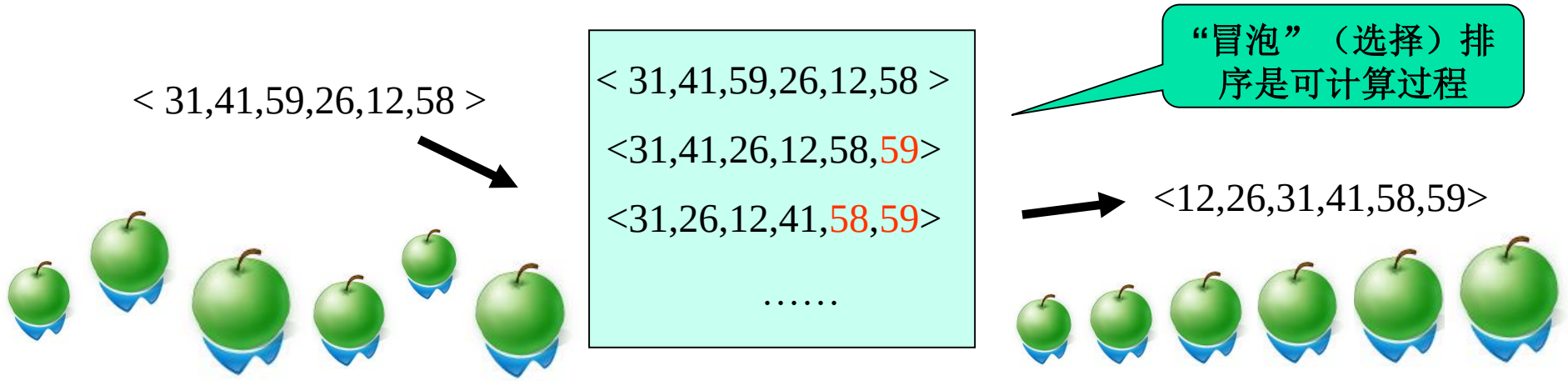
图灵设计的状态自动转移，就是机器指令的例行程序。图灵的指令系统单一不够完善，总结起来主要有两条。第一，没能将指令存储起来重复使用。第二，没能形成实现程序的结构设计。由于这两点缺陷，使图灵机还不能成为处理各种任务的计算机。图灵机欠缺的这两点恰被冯诺依曼提出的程序数据存储的思想解决了。

1.1 What are algorithms

(1) Algorithm is any well-defined computational procedure

(算法：一个定义明确的是可计算过程)

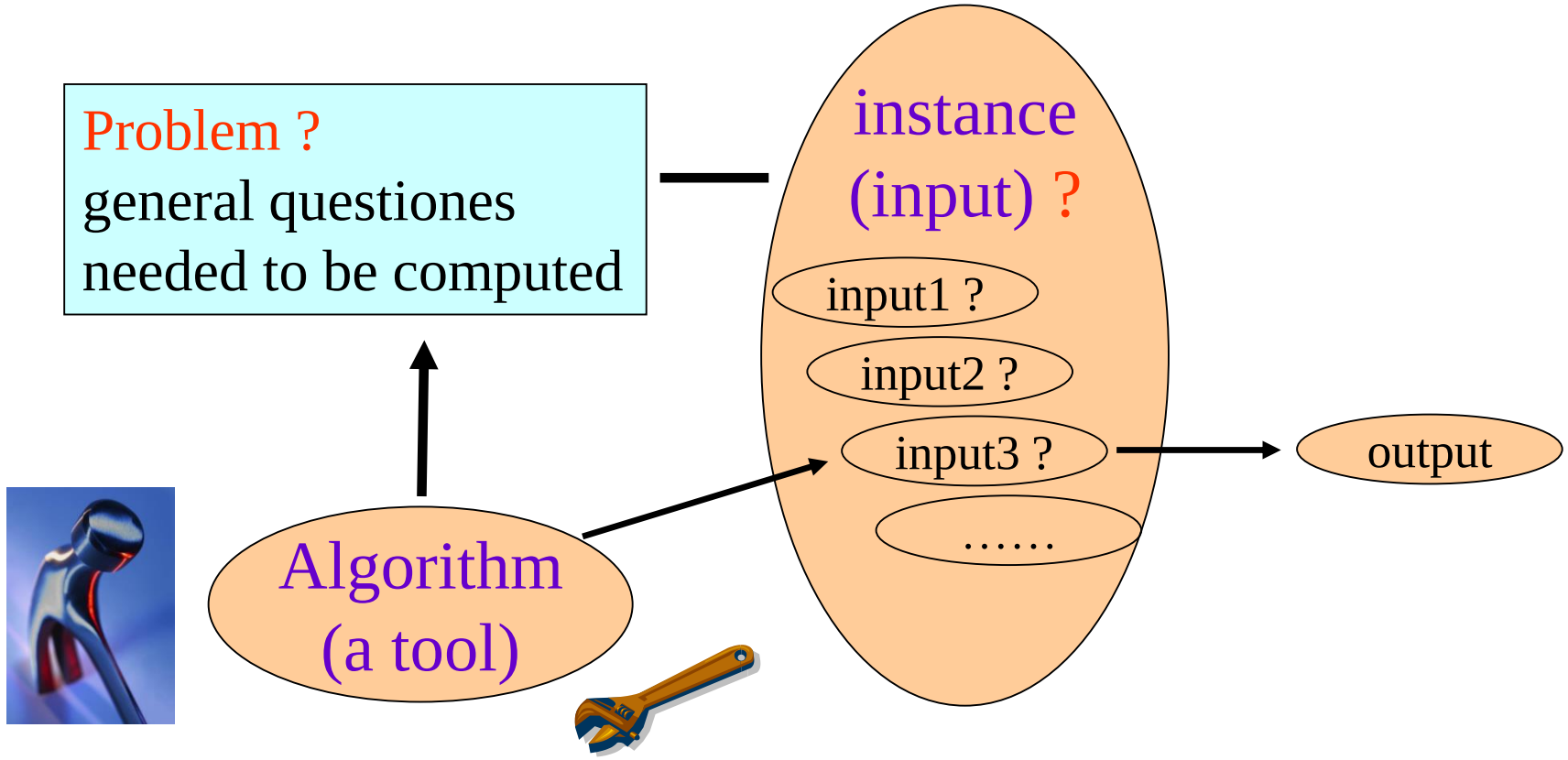




- **Well-defined: know what to do each step; always halts with correct answer .**
- **Efficiency: good or bad?**

(2) Algorithm is a tool for solving a well-specified comutational problem

(算法：一个求解明确的可计算问题的工具)

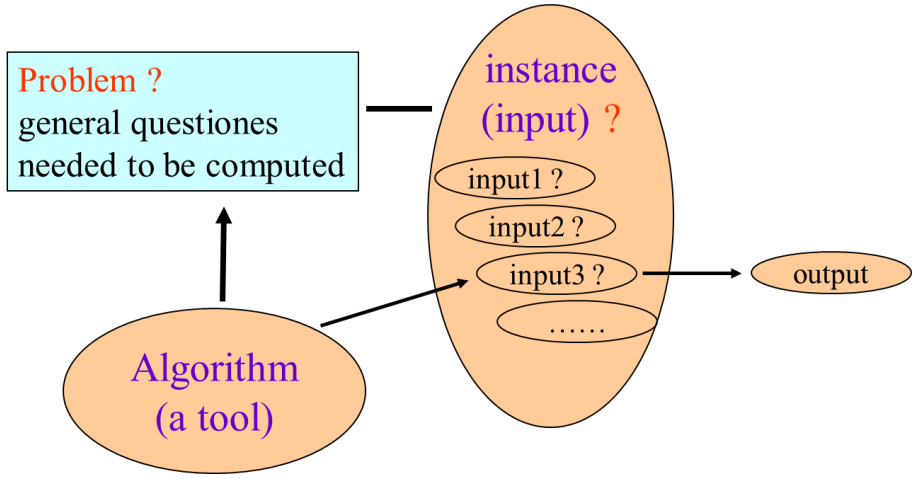


(2) Algorithm is a tool for solving a well-specified comutational problem (example)

Problem:to sort a sequence of numbers into nondecreasing order

Input: A sequence of n numbers $<a_1, a_2, \cdots, a_n>$

Output: A permutation (reordering) $<a_1', a_2', \cdots, a_n'>$ of the input sequence such that $a_1' \le a_2' \le \cdots \le a_n'$



An instance of the sorting problem

Input: <31,41,59,26,41,58>

Output: <26,31,41,41,58,59>

Sorting is a fundamental operation in computer science

1.1 What are algorithms

(2) Algorithm is a tool for solving a well-specified comutational problem

(算法：一个求解明确的可计算问题的工具)

无用之用为大用

The characteristics of the algorithm

- **Output**：at least one.
- **Correct**：An algorithm is said to be **correct**

  if, for **every** input instance, it **halts** with the correct output.（能停机，且结果正确）
- **Feasible**（可行性，可编程实现）
- **Practical** (feasible actually，实际可行)

- Incorrect algorithm
  - might not halt at all on some input instances, or （不能停机）
  - might halt with an answer other than the desired one （停机但结果不正确）
  - can sometimes be useful (if error rate can be controlled)

1.1.1 What kinds of problems are solved by algorithms

- **Genome Project**
- **Internet**

  finding good routes; search engine
- **Electronic commerce**

  keeping information; digital signatures
- **Manufacturing and other commercial settings**

  placing wells; buying campaign advertising

- **Finding the shortest path from one vertex to another**
- **Matrices product:** $A_1A_2A_3A_4$

  $A_1(A_2(A_3A_4))$; $A_1((A_2A_3)A_4)$; $(A_1A_2)(A_3A_4)$

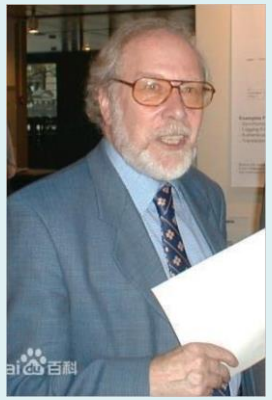
  ……

Algorithms + Data structures = Program

计算机界有这样一种说法：如果说有一个人因为一句话而得到了图灵奖，那么这个人应该就是 Nicklaus Wirth（尼古拉斯·沃斯），

这句话就是他提出的著名公式

“算法+数据结构=程序”



“算法+数据结构=程序”，是Pascal之父Nicklaus Wirth（1984年获得图灵奖）提出的著名公式，该公式对计算机教育、科学、工程、技术产生了深远的影响，这个公式对计算机领域的影响程度足以类似物理学中爱因斯坦的“$E=MC^2$”，这个公式展示出了程序的本质。

通俗地讲，数据结构是程序的基础与骨架，**算法是程序的逻辑与灵魂**。

（卡脖子现状的本质就是缺芯少魂）

1.1.2 Data structures

- **A data structure is a way to store and organize data in order to facilitate access and modifications**（数据结构是存储和组织数据以便于访问和修改的一种方法）
- **No single data structure works well for all purposes**
- **It is important to know the strengths and limitations of data structure**

- **Data structures: static**
- **Algorithms: dynamic**

- **The Art of Computer programming**

  Static + Dynamic

The Art of Computer programming: Static + Dynamic

动静结合总相宜

1.1.3 Technique (有用的技术，有效的方法)

Purpose of studying the course

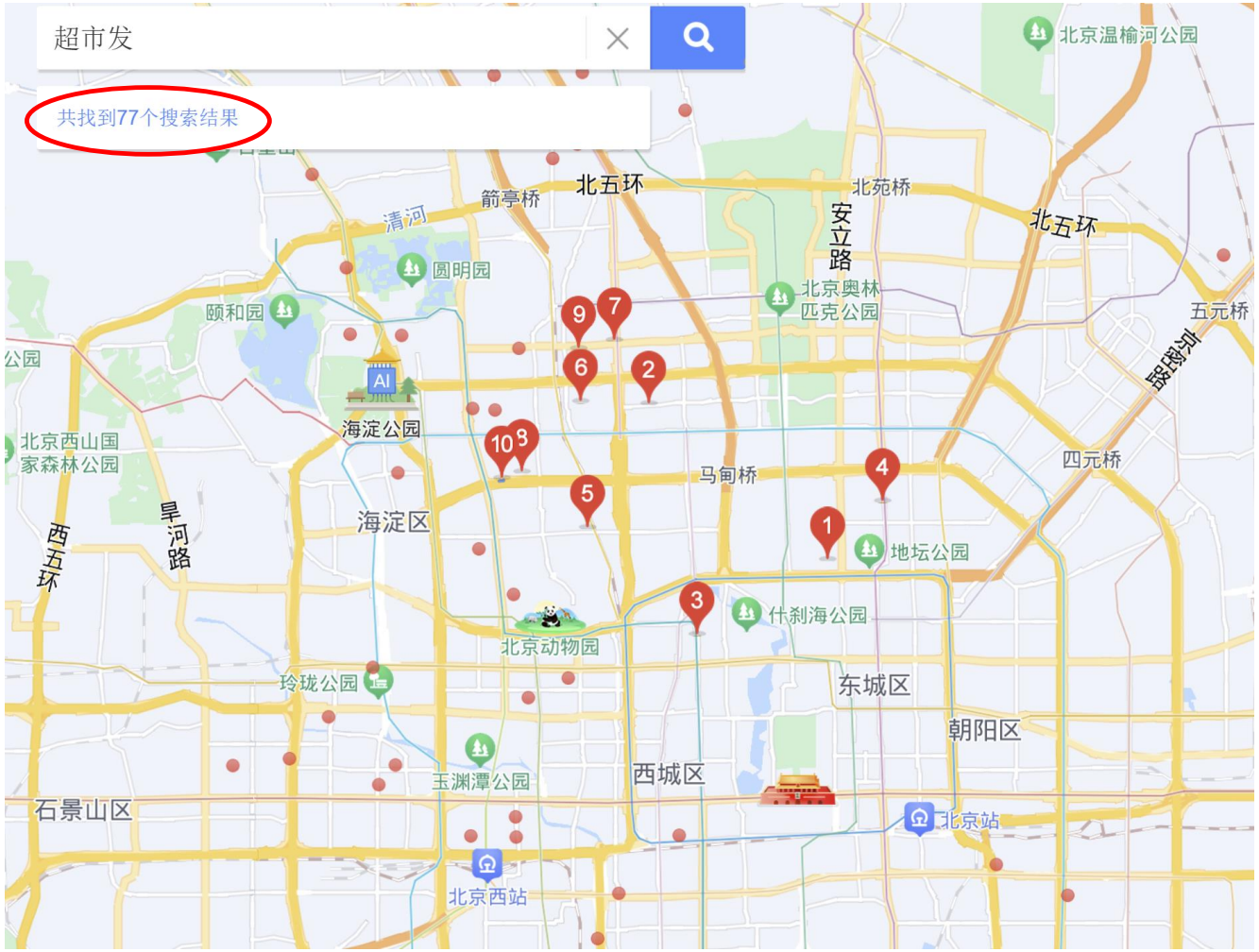
- **Grasping techniques of algorithm design and analysis (chapters and sections)（掌握算法设计与分析的技术与方法）**
- **Developing algorithms on your own, showing that they give the correct answer (exercises and problems)（设计算法求解问题）**
- **Understanding algorithm's efficiency （理解和分析算法的有效性）**

1.1.4 Hard problems（认识“难问题”）

A story of the new manager :

老板让新任配送经理给出一条最快配送路径

20! = 2432902008176640000



一个包裹的实例：北三环中路66号 → 北航

顺丰实际路线

我个人的理想路线：4.4公里

1.1.4 Hard problems（认识“难问题”）

- A story of the new manager
- NP-complete problem
  - It is unknown whether or not efficient algorithms exist for NP-complete problems
    （不知是否存在有效的算法【NP问题有更精细的定义和阐述，以后学习】）
  - In the set of NPC, if an efficient algorithms exists for any one, then efficient algorithms exist for all
  - Several are similar, but not identical, to problems for which we do know of efficient algorithms
- How to show that the problem is NP-complete
  - Traveling-salesman problem
  - Factorization（大的合数分解质因数的问题）：M = p×q （p和q分别什么）
  - ……

1.2 Algorithms as a technology

$$\frac{3\times10^9\times3\times10^9\,\text{instruc}}{10^9\,\text{instruc/s}} = 9\times10^9\,\text{seconds} = \frac{9\times10^9}{60\times60\times24\times365}\,\text{y} \approx 285.39\,\text{years}$$

$$\frac{\infty}{\infty} = ?$$

- **If computer infinitely fast, memory free. Study algorithms worth?**

  YES! Need demonstrate that the solution method terminates with the correct answer.

- **Computing time is a bounded resources, so is space**
  - **Computers may be fast, but not infinitely fast**
  - **Memory may be cheap, but not free**
  - **Resources should be used wisely**

1.2.1 Efficiency

能写出程序来计算就行，别管我用什么算法?

(1) To same problem, algorithms often differ dramatically in their efficiency （同一问题，同样计算环境，不同算法的效率可能悬殊巨大）（A1 < A2）

$$\frac{3\times10^9\times3\times10^9\text{instruc}}{10^9\text{instruc/s}}\approx285.39\text{years}$$ **vs** $$\frac{3\times10^9\times\lg(3\times10^9)\text{instruc}}{10^9\text{instruc/s}}\approx94.45\text{seconds}$$

1.2.1 Efficiency

我有钱，能买足够快的计算机，算法无所谓?

(2) The differences can be much more significant than differences due to hardware and software
（改进算法可能比升级硬件更有益）（A > H & S）

Example: sorting ten million (10^7) numbers

① **Insertion sort:** $T(n) = c_1 n^2$

- **C1: Computer A, 10^9 instruc/s**
- **P1: World's craftiest programmer**
- **L1: Machine language**

$$T(n) = 2n^2$$

$$t = \frac{2 \cdot (10^7)^2 \text{instruc}}{10^9 \text{instruc/s}} = 2 \times 10^5 \text{s} \approx 55.56\text{h}$$

② **Merge sort:** $T(n) = c_2 n \lg n$

- **C2: Computer B: 10^7 instruc/s**
- **P2: Average programmer**
- **L2: High-level language**

$$T(n) = 50 n \lg n$$

$$t = \frac{50 \cdot 10^7 \lg 10^7 \text{instruc}}{10^7 \text{instruc/s}} \approx 19.38\text{m}$$

	好条件：C1+P1+L1	差条件：C2+P2+L2
好算法A1 ($n\lg n$)		19.4m
差算法A2 (n^2)	55.6h	

1.2.1 Efficiency

既然算法是“一切”，硬件不重要?

(3) Efficient algorithm gains much more from the hardware improvement. For example,
（硬件条件升级后，好算法带来的“效益”更高！）

- Suppose that the computer makes one basic step per second (v=1). Algorithm A and B use the same computer.
（设每秒执行1个基本步骤，算法A和算法B的计算环境相同。）
Algorithm A runs as $(n_A)^2$, and B as $100n_B$.
- If an application requires you to complete as many items as possible in one hour (t = 3600 sec).
（应用需求：1小时能处理多大规模的输入问题【1小时内最多能计算多少输入项】？）
- Thus if A can make n_A items , and B makes n_B items, then
$$v_A t = 3600 = (n_A)^2 = 100n_B$$
yields n_A = 60 and n_B = 36.

- But suppose that you replace computer A with B that is hundred times as fast. So you can make 360000 steps in the same time （新计算机速度提高100倍，1小时能计算多少项？）. Then
$v_B t = 100 v_A t = 360000 = (n_A)^2 = 100n_B$
yields n_A = 600 and n_B = 3600.

Algorithm B gains much more from the faster computer than A !!!

1.2.1 Efficiency

(1) A1 vs A2
同条件下，好算法很重要

	好条件C1	差条件C2
好算法A1		X
差算法A2		Y

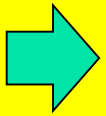
	好条件C1	差条件C2
好算法A1	X	
差算法A2	Y	

(2) A vs H+S
算法好比硬件好重要

	好条件 C1+P1+L1	差条件 C2+P2+L2
好算法A1 (*n*lg*n*)		19.4m
差算法A2 (*n*^2)	55.6h	

(3) A+H+S
硬件好，算法好，效益更明显

	好条件C1	差条件C2
好算法A1		X
差算法A2		Y



	好条件C1	差条件C2
好算法A1	X	
差算法A2	Y	

1.2.2 Algorithms and other technologies（算法与其他技术的关系）

Like computer hardware, algorithms are technology

(1) Algorithms are important on contemporary computers advanced technologies（算法与其他技术同等重要）, such as

- Hardware with high clock rates, pipelining, and superscalar architectures（高性能计算设备、流水线操作、超大规模体系结构）
- Easy-to-use, intuitive graphical user interfaces (GUIs)
- Object-oriented systems
- Local-area and wide-area networking

For example,

1.2.2 Algorithms and other technologies（算法与其他技术的关系）

For example,

a web-based service that determines how to travel from one location to another
（一个网络服务，从一个点到另一个点的应用）

- The implementation would rely on fast hardware, a GUI, wide-area networking, and also possibly on OO
（依赖于GUI，广域网，面向对象技术等）

- It would also require algorithms for certain operations, such as finding routes, rendering maps, interpolating address
（也需要相关算法，如：路径寻找，地图渲染，寻址）

1.2.2 Algorithms and other technologies（算法与其他技术的关系）

(2) Other technologies rely on algorithms

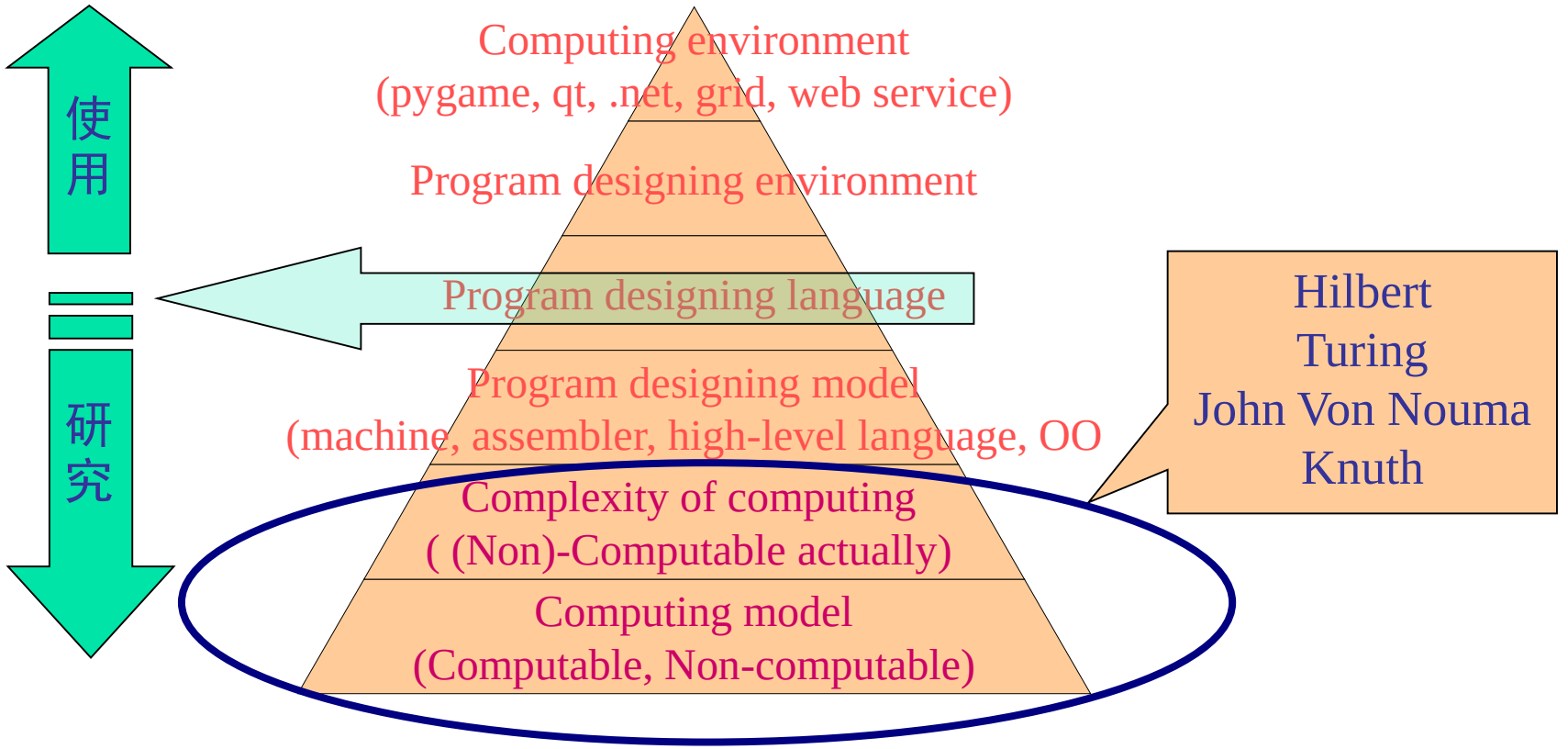
- The hardware design used algorithms（电路规划等硬件涉及使用算法）
- The design of any GUI relies on algorithms
- Routing in networks relies heavily on algorithms

(3) The program languages were processed by a compiler, interpreter, assembler, all of which make extensive use of algorithms (accidence analysis, syntax analysis)

（程序编译、解释、汇编等广泛利用算法，如词法分析、语法分析）

1.2.2 Algorithms and other technologies（算法与其他技术的关系）

Algorithms are at the core of most technologies used in contemporary computers



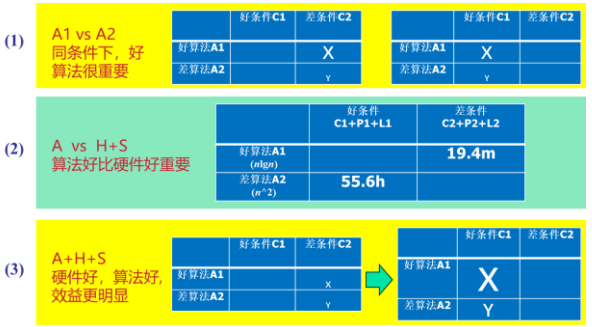
软件专业的学生必须要有坚实的基础

1.2.2 Algorithms and other technologies（算法与其他技术的关系）

- Algorithms are at the core of most technologies used in contemporary computers.
- Having a solid base of algorithmic knowledge and technique is one characteristic that separates the truly skilled programmers from the novices.
（算法能力是区分高水平程序员和新手的重要特征）
- With modern computing technology, you can accomplish some tasks without knowing much about algorithms.
（借助现代计算技术，不需要掌握算法也能完成一些任务）
- With a good background in algorithms, you can do much, much more !!!
（算法素养好，能做得更好）

Exercises and problems

1. All of the exercises and problems.



2. Give an example respectively to explain and analyze the following descriptions:

举一个问题实例，用两种算法求解，分别说明和分析以下描述（不能用前面讲的 insertSort 和 mergeSort）：

(1) To the same problem, in the same computational condition, algorithms often differ dramatically in their efficiency.
（同样的计算条件下，“好”算法比“差”算法重要！）

(2) For the computing efficiency, the differences due to algorithms can be much more significant than differences due to hardware and software.
（好算法比好的软硬件更重要！）

(3) Efficient algorithm gains much more from the hardware improvement.
（硬件重要，但借助好算法，好硬件能发挥更重要作用！）