

Chapter 1

The role of algorithms in computing

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，或某种语言写的程序，目的是解决计算(computing)问题，如：

$$F(n) = F(n-1) + F(n-2)$$

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;
}
```

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）
 - ◆ 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。

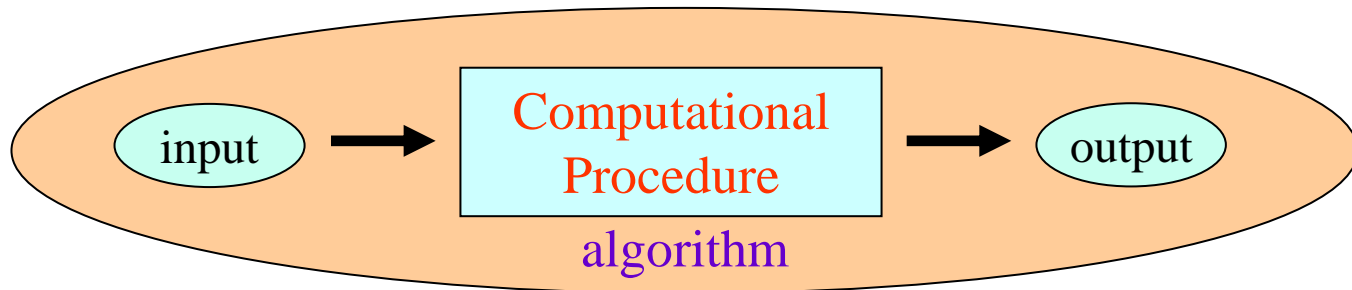


图灵设计的状态自动转移，就是机器指令的例行程序。图灵的指令系统单一不够完善，总结起来主要有两条。第一，没能将指令存储起来重复使用。第二，没能形成实现程序结构设计。由于这两点缺陷，使图灵机还不能成为能够处理各种任务的计算机。图灵机欠缺的这两点恰被冯诺依曼提出的程序数据存储的思想解决了。

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)



$\langle 31, 41, 59, 26, 12, 58 \rangle$



$\langle 31, 41, 59, 26, 12, 58 \rangle$

$\langle 31, 41, 26, 12, 58, 59 \rangle$

$\langle 31, 26, 12, 41, 58, 59 \rangle$

.....

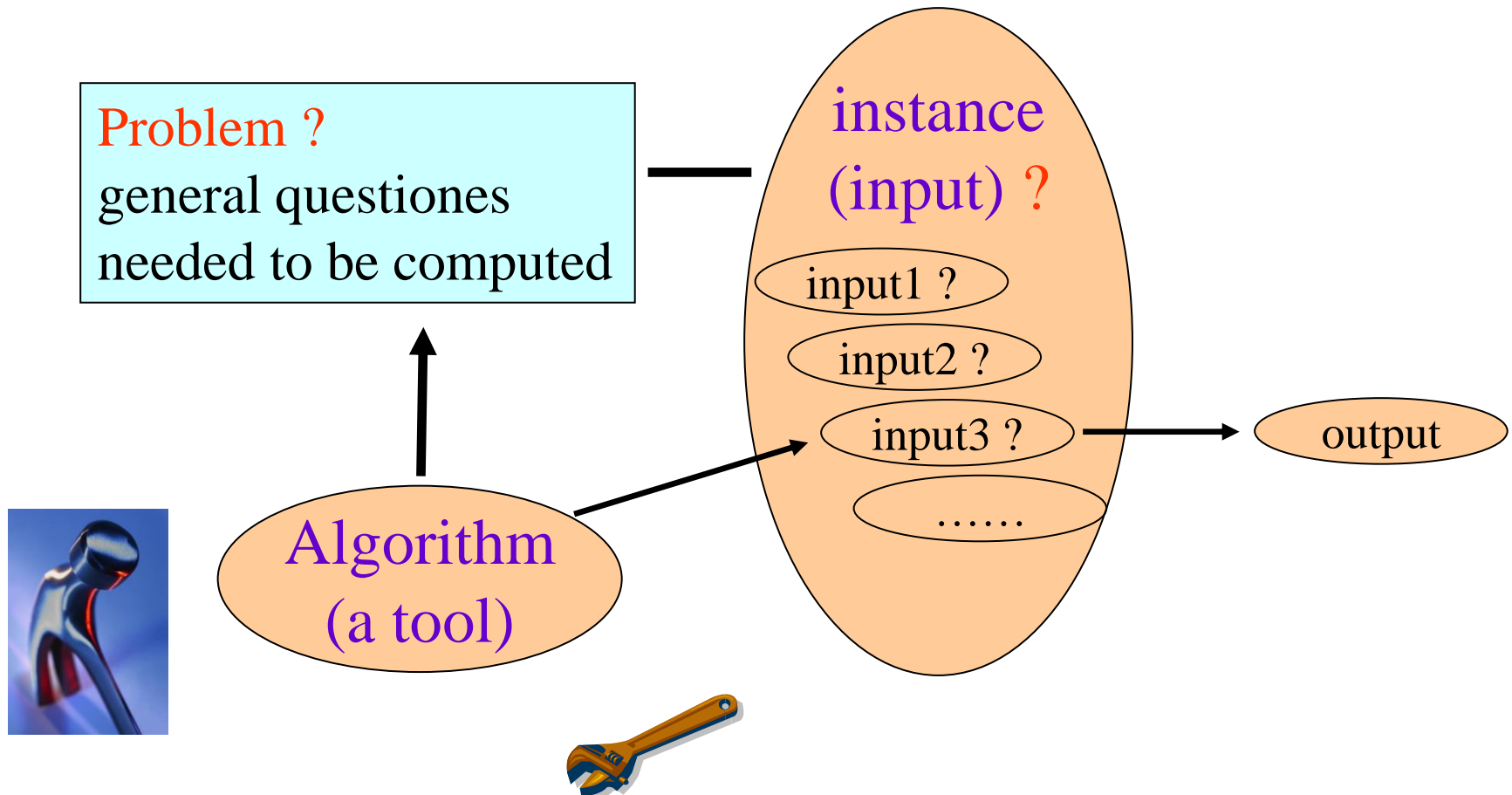
“冒泡”（选择）排序是可计算过程

$\langle 12, 26, 31, 41, 58, 59 \rangle$

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

1.1 What are algorithms

2) Algorithm is a **tool** for solving a well-specified computational problem



1.1 What are algorithms

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

Problem: to sort a sequence of numbers into nondecreasing order

Input: A sequence of n numbers $\langle a_1, a_2, \dots, a_n \rangle$

Output: A permutation (reordering) $\langle a'_1, a'_2, \dots, a'_n \rangle$ of the input sequence such that $a'_1 \leq a'_2 \leq \dots \leq a'_n$

Questions
to be computed

Algorithm
(a tool)

(input) ?

input1 ?

input2 ?

input3 ?

.....

An **instance** of the sorting problem

Input: $\langle 31, 41, 59, 26, 41, 58 \rangle$

Output: $\langle 26, 31, 41, 41, 58, 59 \rangle$

Sorting is a fundamental operation
in computer science

1.1 What are algorithms

2) Algorithm is a **tool** for solving a well-specified computational problem



无用之用为大用

1.1 What are algorithms

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.
- ◆ **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
 - might halt with an answer other than the desired one
 - 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:** $A_1 A_2 A_3 A_4$

$$A_1(A_2(A_3 A_4)); A_1((A_2 A_3) A_4); (A_1 A_2)(A_3 A_4)$$

.....

1.1.2 Data structures

- A data structure is a way to **store** and **organize** data in order to facilitate access and modifications
- 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

1.1.2 Data structures

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=MC²”**，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

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

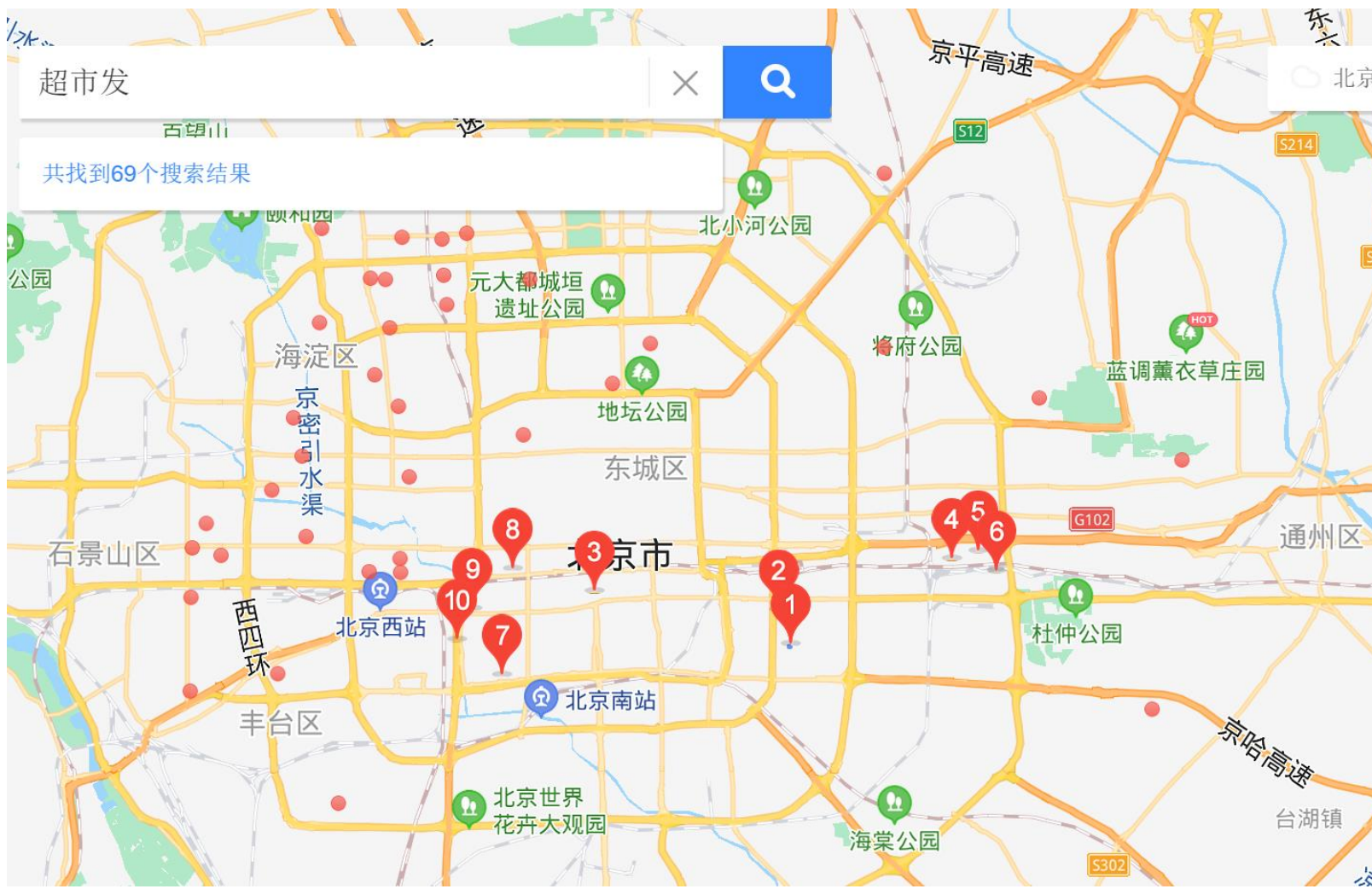
- **Grasping techniques of algorithm design and analysis (chapters and sections)**
- **Developing algorithms on your own, showing that they give the correct answer (exercises and problems)**
- **Understanding algorithm's efficiency**

1.1.4 Hard problems

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

$20! = 2432902008176640000$



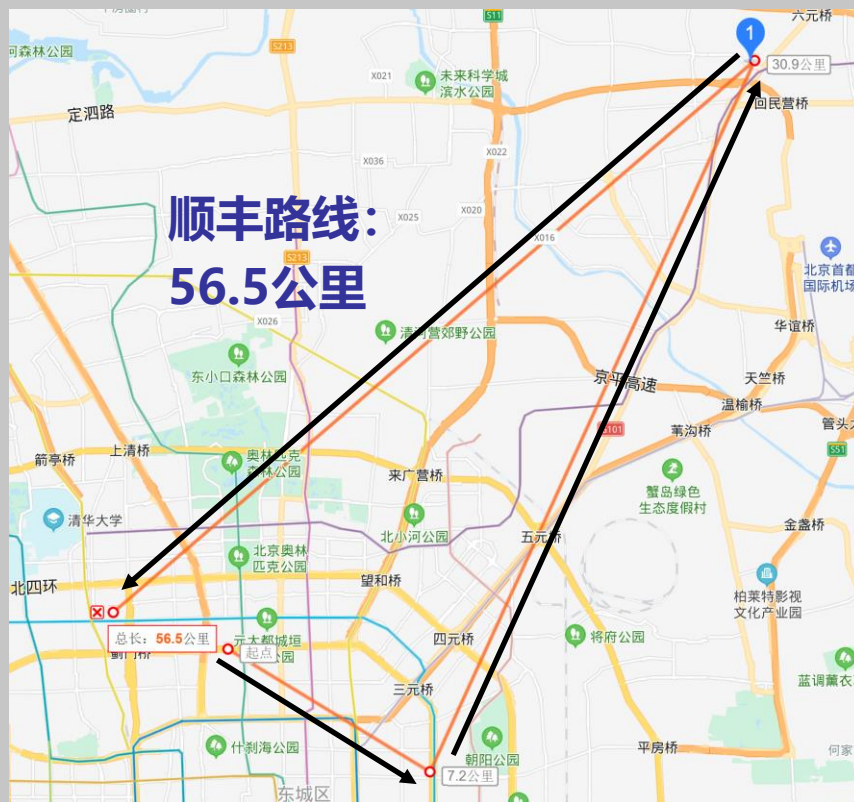
一个包裹的实例：北三环中路66号 → 北航

← 物流详情

物流状态：运输中

顺丰速运：SF1

- 快件到达【海淀学院路速运营业点】
9月8日 07:09:31
- 快件已发车
9月8日 05:55:09
- 快件在【北京南法信中转场】已装车,准备发往【海淀学院路速运营业点】
9月8日 05:54:58
- 快件到达【北京南法信中转场】
9月8日 04:00:09
- 快件已发车
9月8日 03:49:21
- 快件在【北京顺义集散中心】已装车,准备发往【北京南法信中转场】
9月8日 02:26:21
- 快件到达【北京顺义集散中心】
9月7日 21:50:31
- 快件已发车
9月7日 21:04:46
- 快件在【北京朝阳华贸营业点】已装车,准备发往【北京顺义集散中心】
9月7日 20:53:16
- 顺丰速运 已收取快件
9月7日 15:33:28



1.1.4 Hard problems

- 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）
 - ◆ 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）： $M = ? \times ?$
 - ◆

1.2 Algorithms as a technology

$$\frac{3 \times 10^9 \times 3 \times 10^9 \text{ instruc}}{10^9 \text{ instruc/s}} = 9 \times 10^9 \text{ seconds} = \frac{9 \times 10^9}{60 \times 60 \times 24 \times 365} \text{ y} \approx 285.39 \text{ years}$$

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

- 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，什么算法都行？

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

$$\frac{3 \times 10^9 \times 3 \times 10^9 \text{instruc}}{10^9 \text{instruc/s}} \approx 285.39 \text{years}$$

$$\frac{3 \times 10^9 \times \lg(3 \times 10^9) \text{instruc}}{10^9 \text{instruc/s}} \approx 94.45 \text{seconds}$$

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:** $T(n) = c_1 n^2$

- Computer A: 10^9 instruc/s
- World's craftiest programmer
- 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:** $T(n) = c_2 n \lg n$

- Computer B: 10^7 instruc/s
- Average programmer
- High-level language

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

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

	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,

- Suppose that the computer makes **one basic step per second** ($v=1$). Algorithm A and B use the same computer.
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).
- Thus if A can make n_A items, and B makes n_B items, then
$$v t = 3600 = (n_A)^2 = 100n_B$$
yields $n_A = 60$ and $n_B = 36$.
- But suppose that you replace the computer with one that is **hundred times as fast**. So you can make 360000 steps in the same time. Then
$$360000 = (n_A)^2 = 100n_B$$
yields $n_A = 600$ and $n_B = 3600$.

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

1.2.1 Efficiency

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	

A vs H+S

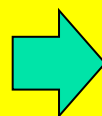
算法好比硬件好重要

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

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

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,

**For example,
a web-based service that determines how to travel
from one location to another**

- ◆ **The implementation would rely on fast hardware, a GUI, wide-area networking, and also possibly on OO**
- ◆ **It would also require algorithms for certain operations, such as finding routes, rendering maps, interpolating address** （路径寻找，地图渲染，寻址）

1.2.2 Algorithms and other technologies

(2) Other technologies rely on algorithms

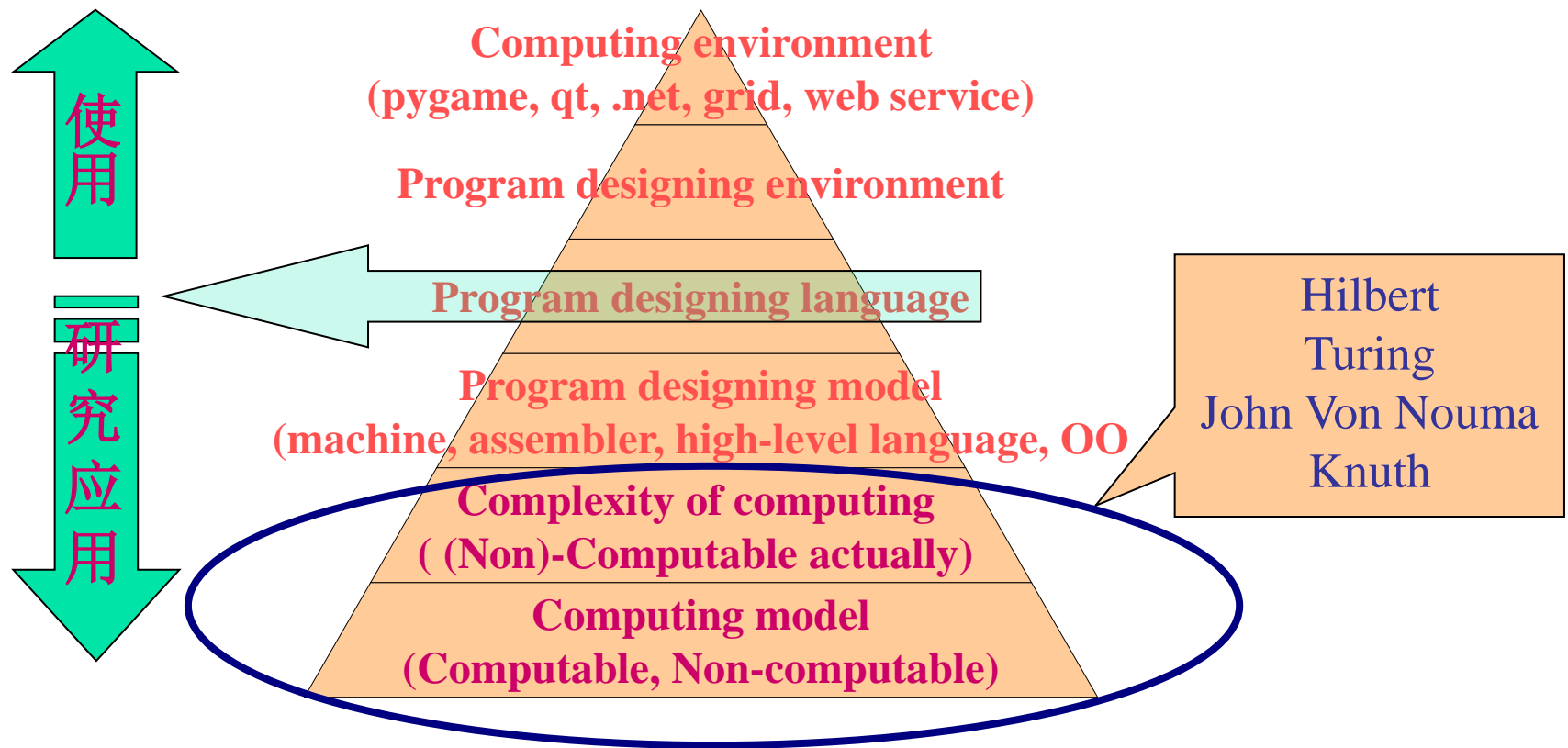
- ◆ **The hardware design used algorithms**
- ◆ **The design of any GUI relies on algorithms**
- ◆ **Routing in networks relies heavily on algorithms**

(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

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

- **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 (初学者)**
- **With modern computing technology, you can accomplish some tasks without knowing much about algorithms**
- **With a good background in algorithms, you can do much, much more !!!**

Exercises and problems

All of the exercises and problems.

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

- 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重要!)**
- 3) Efficient algorithm gains much more from the hardware improvement. (硬件重要, 有了好算法, 好硬件才更重要!)**