

计算思维引导

陶先平

南京大学计算机软件研究所

找假币---何谓“计算思维”？

- 给你70个外观完全一样的金币,但是你知道其中有一个是假币,其重量比真币轻。给你一架没有砝码的天平,你可以在天平两边摆任意多个金币,比较他们的轻重。
- 请设计一种方法,通过若干次称量,确定哪一个假币。



第一个解法

总共称7次

- 先将70个金币平均分两份，放到天平两边。

假币必在轻的那一侧的35个中。

- 将那35个加上另一侧中任意一个，再平分称量。

- 将轻的一侧的18个平分再称量。

- 将轻的一侧的9个加另外任意一个再平分称量。

- 将轻的一侧的5个加另外任意一个再平分称量。

- 将轻的一侧的3个加另外任意一个再平分称量。

- 将轻的一侧的2个分别放到天平两边，轻的是假币。完成

第一个解法稍稍改进

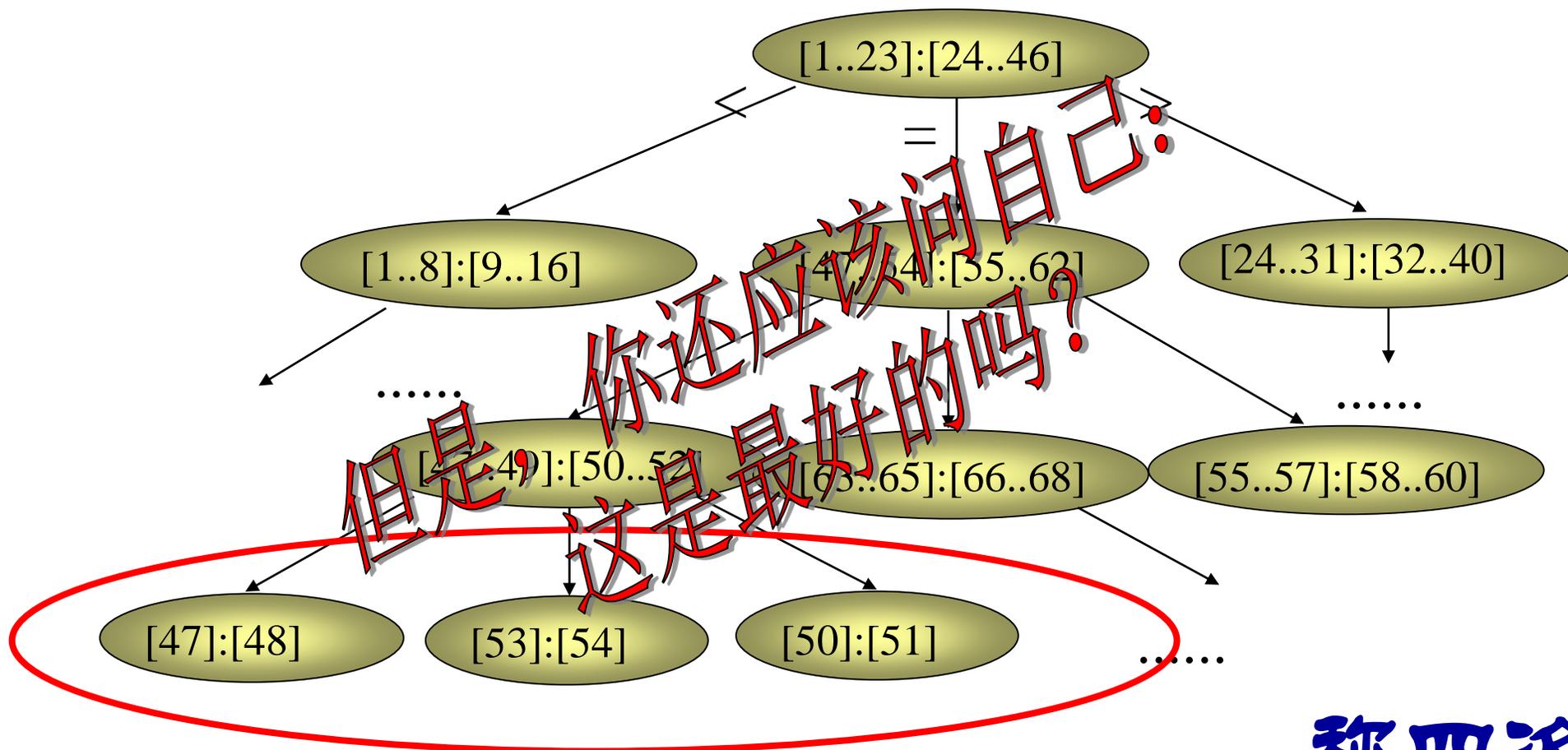
总共称6次

- 先将70个金币平均分两份，放到天平两边。

假币必在轻的那一侧的35个中。
- 将那35个中任意一个放一旁，其余再平分称量，每边17个。
- 若相等则旁边的是假币，否则将轻的一侧的17个中任意一个放旁边，其余的再平分再称量，每边8个。
- 若相等则旁边的是假币，否则将轻的一侧的8个平分再称量，每边4个。
- 将轻的一侧的4个再平分称量，每边2个。
- 将轻的一侧的2个再平分称量，每边1个。轻的是假币。完成

但是，你还能更好吗？
你还应该问自己！

一个更好的办法



称四次!

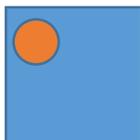
我们为什么会这样思考来找到最快的方法？

问题空间：

我们所观察到的对象
(每个对象的状态及其变化)



第一种方案：



几乎每次压缩问题空间到一半

第二种方案：



几乎每次压缩问题空间到三分之一



如何表达我们的这个思想？写个程序！

```
Procedure FindIt(n) {
    if n=2 {
        天平上翘起的是假币；程序结束；
    }
    if n=3 {
        任取两个放在天平上；
        如果平衡，假币为第三个，否则为翘起的天平端硬币；程序结束；
    } else {
        将硬币分为几乎数量相同的三堆n1,n2,n3；
        称量其中数量相同的两堆；
        如果两堆不同重 {
            FindIt(n1);
        } else FindIt(n3);
    }
}
```

//从n个硬币中找出一个较轻的假币
//只有两个硬币

//有三个硬币

//有多于三个硬币
//其中必定有两堆数量相同
//不妨假设n1=n2
//不妨假设n1堆轻

到此为止，这个问题我们解决了吗？

No!

我们还应该至少回答这些问题：

你能证明你的解法是正确的吗？

你能证明你的解法是最优的吗？

你能证明你的程序没有错误吗？

再一个互动游戏：

- 统计到场人数：
 - 0，所有人都站起来，每个人都握有一个数字：1
 - 1，每两个人组成一组，将手中数字相加，并记住。其中一人坐下；
 - 2，重复第一步，直到教室中只有一人；
 - 3，最后一人，大声报出数字；

这个游戏，给了我们什么启发？

- 1, 依然是“压缩问题空间”：
 - N压缩到 $n-1$ ==》 n压缩到 $n/2$
 - 三人或者四人或者.....都是一种可能的选择，只要一次统计能够被“简单”完成
- 2, 如果每次分组（两人组）后，组内的统计、累计都可以在组内完成，那么：我就只需要完成分组、同步和最后数据的收集工作
 - 每个小组，可以并行完成组内工作
 - 每个小组都是一个小型计算机系统
 - N个人，如果小组规模是 m ，那么我只需要进行约 $\log_m n$ 次的分组、同步工作
 - 我是一个管理了多个可并行运行的计算机系统的“并行计算机系统”
 - 多核系统是一个典型案例
- 分治法+并行处理：极大提高了问题求解的效率

如何表达我们的解题过程呢？

- 假设我们有 $p+1$ 个处理器(0,...,p),其中第0号是master, 其它是slave

• Parallel Procedure count(n) {

 if (I'm the master){

 将n个数据分为p份: n_1, n_2, \dots, n_p

 for (i=1 to p step 1){

 count(n_i);

 }

 for (i=1 to p step 1){

 receive value from p_i ;

 sum = sum+value;

 }

 }else{

 }else{

 //slaves

 接收master给予的数据;

 for (i=1 to n/p step 1){

 value= GetValue(i);

 sum = sum+value;

 }

 send sum to master;

 }

}

是什么导致了我们的独特视角？

关于计算思维的一些理解

- 计算思维是我们认知计算的过程中积累形成的思考“模式”
 -
- 计算思维教学需要传递计算给我们带来的可能性以及实现这些可能的基本方法
- 算法是解读计算思维的最佳载体

1, 计算思维是我们认知计算过程中积累的思考“模式”

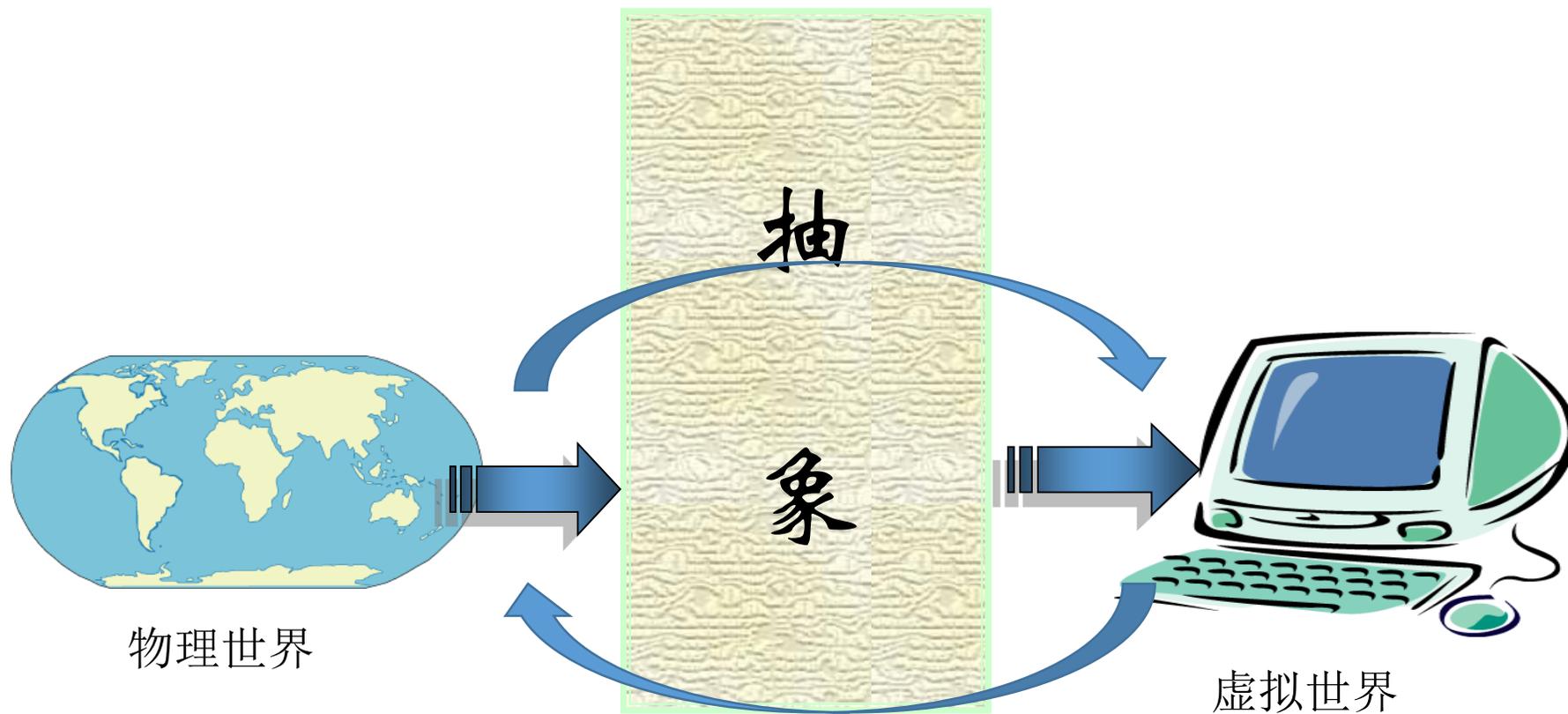
- 思维是一种认知过程
- 计算思维是我们认知计算过程中若干层面的抽象及其实现中“沉淀”下来的一些.....

计算思维： 抽象化+自动化：

三个层面的抽象过程及相应的自动化过程

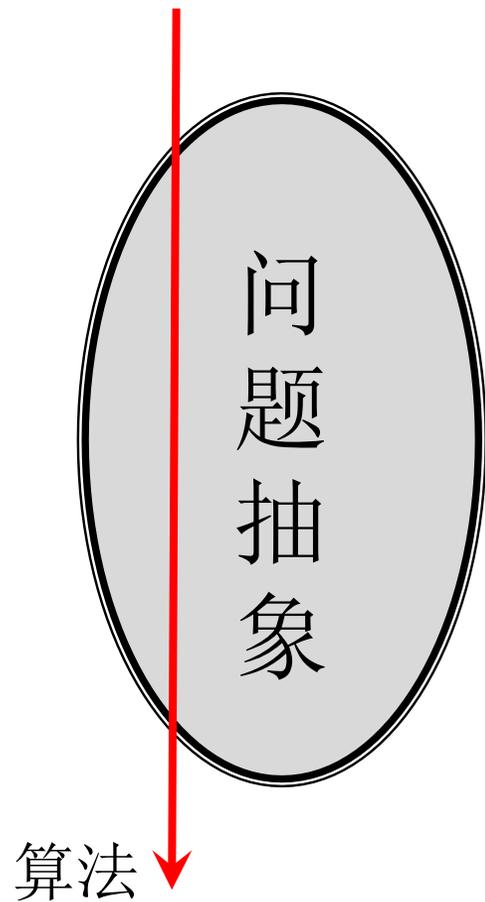
如何去“传递”抽象化+自动化？

我们用计算机干什么？



走向物理世界与虚拟世界的无缝连接

问题抽象



算法是计算思维的核心概念

:

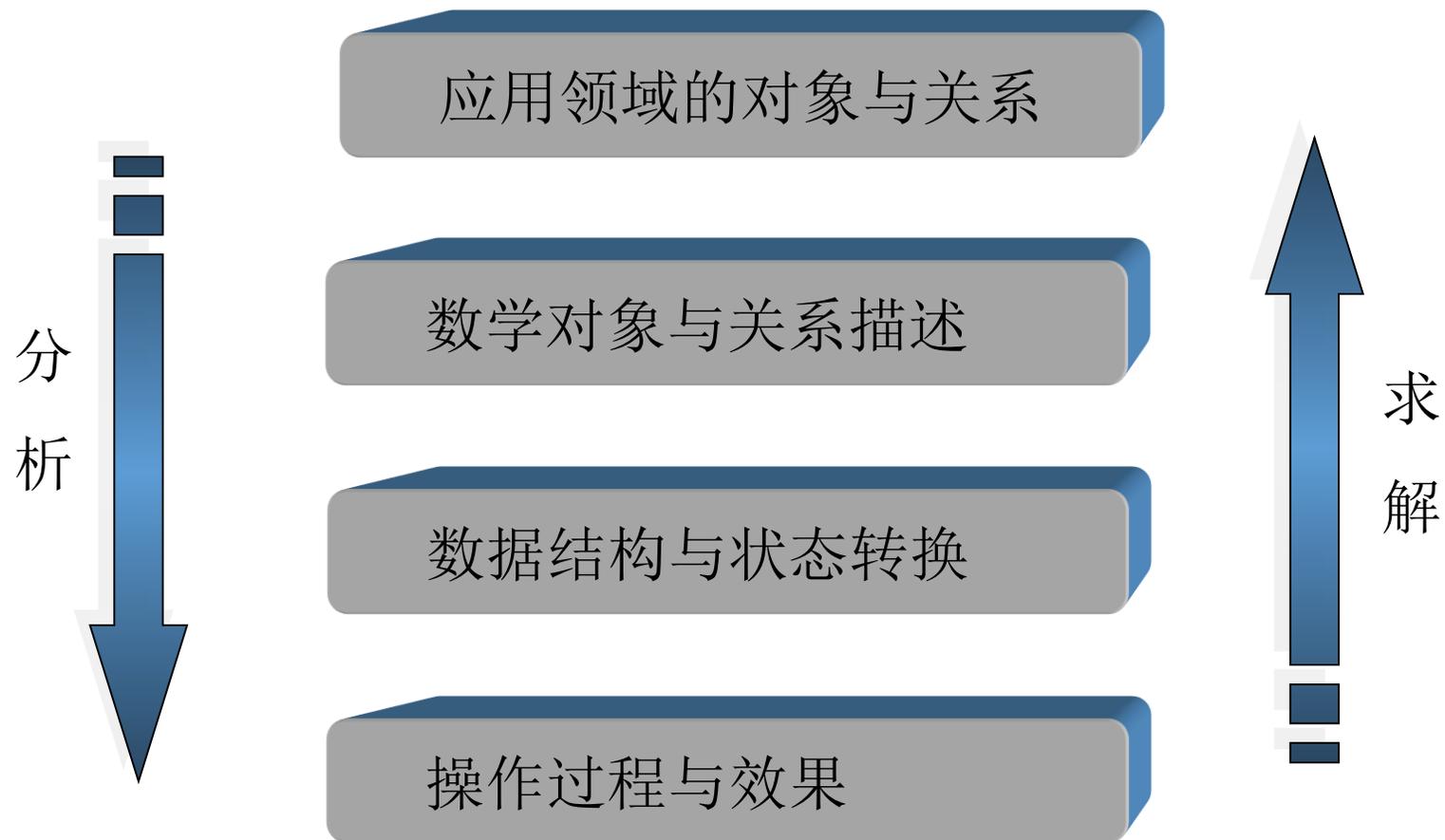
方法层：算法

表示层：编程

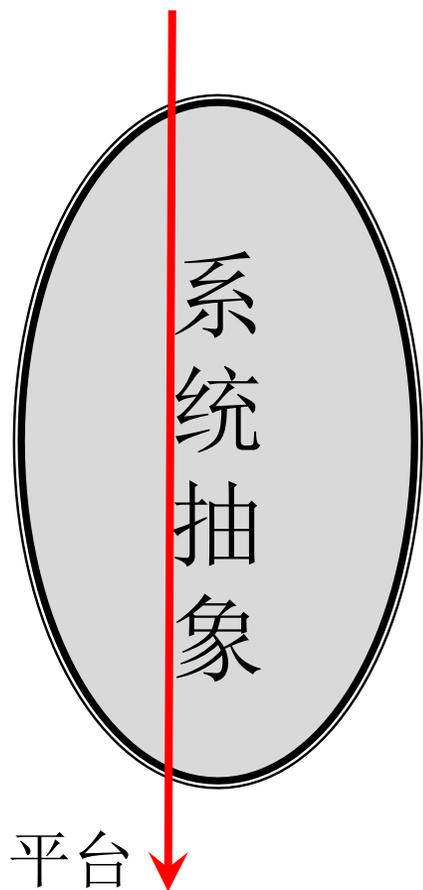
实现层：机器

这差不多也
就是计算机
科学的主要
内容了

问题抽象的分层映射



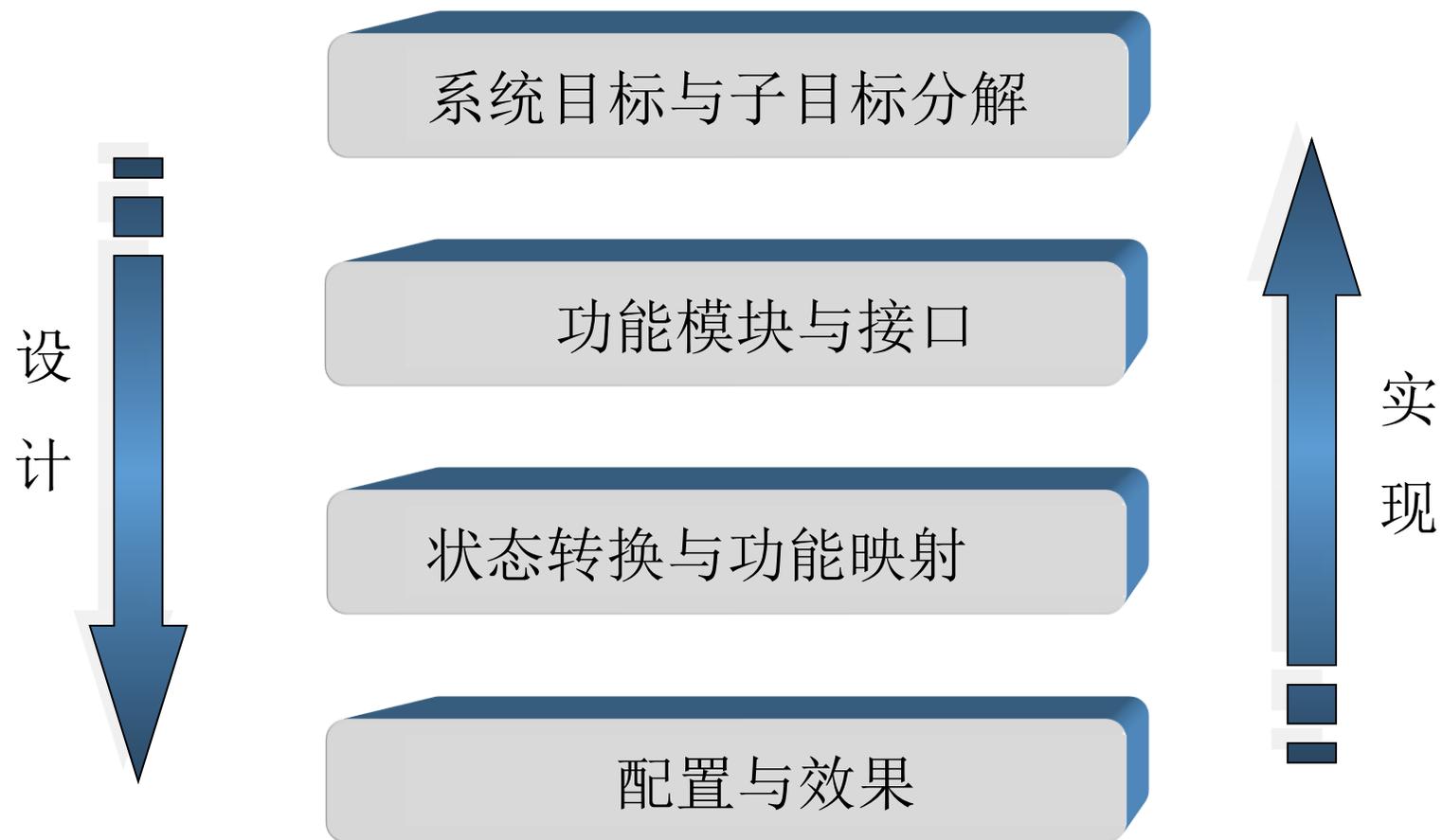
系统抽象



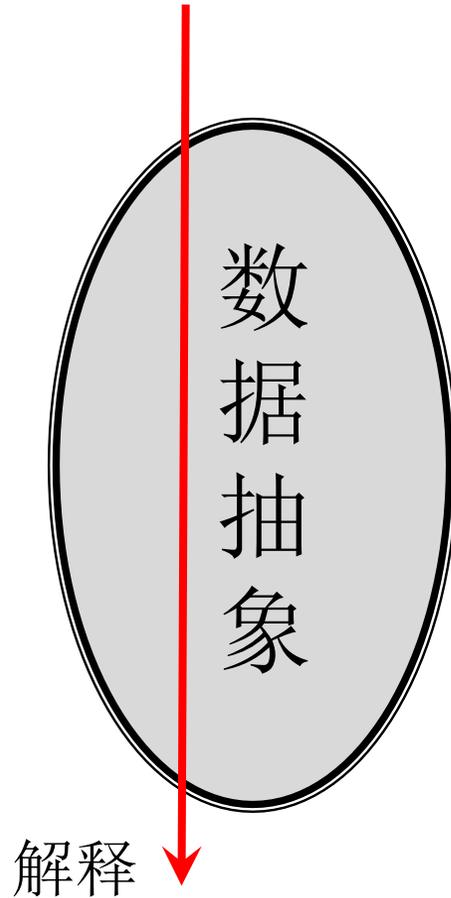
核心概念：

系统模型、功能逻辑、
接口、实现

系统抽象的分层映射

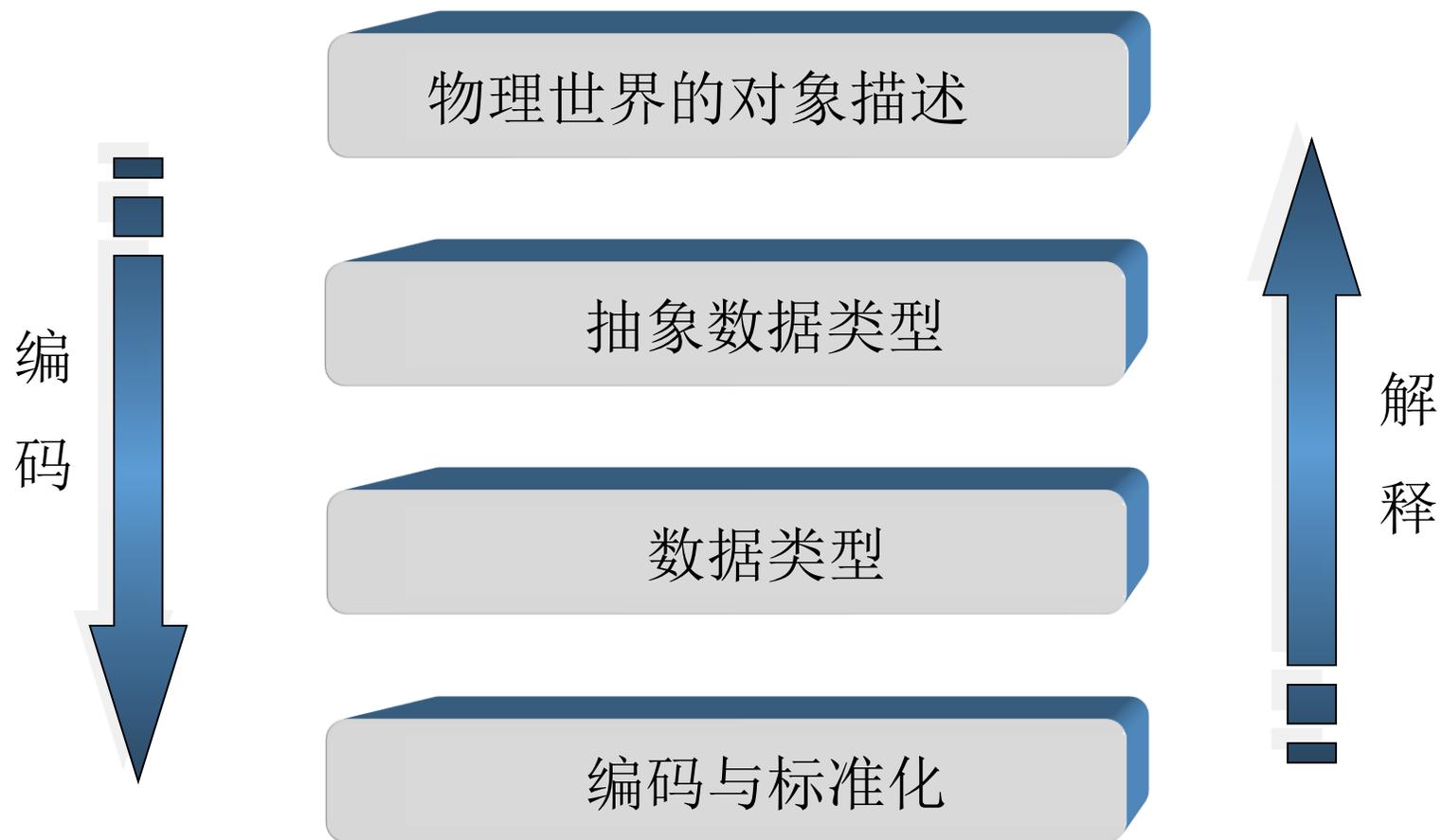


数据抽象



核心概念：
信息形态、信息组织、
存储、检索与利用

数据抽象的分层映射



2, 计算思维教/学需要传递/感悟计算给我们带来的可能性以及实现这些可能的基本方法

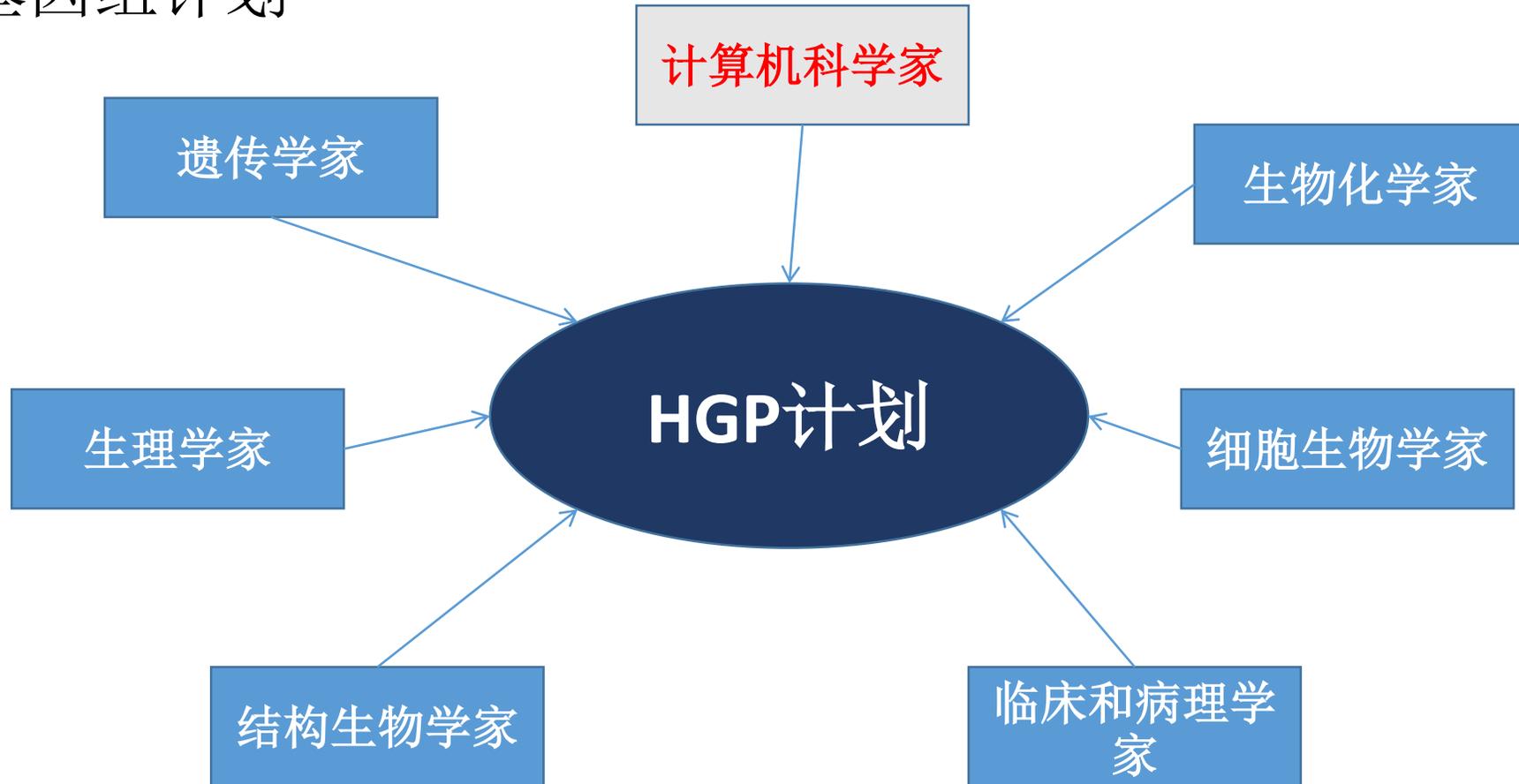
- 想以前想不到之事
- 做以前做不到之事
- 做以前做不好之事

想以前想不到之事

- 全球脉动(Global Pulse)计划：
 - 联合国已经推出的新项目，希望利用“大数据”来促进全球经济发展
 - 进行所谓的“情绪分析”，使用软件来对社交网站和文本消息中的信息作出分析
 - 帮助预测某个给定地区的失业率、支出削减或是疾病爆发等现象
 - 目标在于利用数字化的早期预警信号来提前指导援助项目，以阻止某个地区重新陷入贫困等困境

做以前做不到之事

- 人类基因组计划



做以前做不好之事

- ERP系统

- 针对物资资源管理（物流）、人力资源管理（人流）、财务资源管理（财流）、信息资源管理（信息流）而开展的集成一体化的企业管理
- 信息技术带给了我们庞大的处理能力：
 - 复杂业务模型、复杂管理要求、
 - 复杂合作关系、复杂时空数据类型.....

- 计算给这个世界带来的不是这个和那个技术，也不是这个或者那个炫目应用
- 计算带给我们的是无限的想象空间和强有力的实现手段

人有多大胆
地有多大产

试图给出计算思维的定义

美国卡内基-梅隆大学教授Jeannette M. Wing(周以真)领导世界上可能是唯一的“计算思维研究中心”，并大力推动这一概念被认可。以下内容摘自她的文章：

[Computational Thinking: What and Why?](#) Link Magazine, 2010

Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent [CunySnyderWing10]

Informally, computational thinking describes the mental activity in formulating a problem to admit a computational solution. The solution can be carried out by a human or machine, or more generally, by combinations of humans and machines.

<http://www.cs.cmu.edu/~CompThink/papers/TheLinkWing.pdf>

结束语

计算思维看不见，摸不着，但影响着你的决策！

计算思维：当你面临一个要解决的问题时，如果你的第一感觉是去寻找一个数学模型对问题和解进行建模，去尝试着通过算法来寻找解，并尝试着思考如何用一个辅助工具开展计算时，计算思维已经在影响你了！

有很多的计算过程中沉淀下来的模式，被封装为计算思维，可以被我们直接使用

作业

请你为某个型号的电子词典，设计一个查找单词的递归算法(伪代码)

- 提示：
- 1，电子词典已经按照词典序排好了序，词典中共有 n 个单词；
 - 2，你可以直接使用`compare(x,y)`函数来判断单词 x 和 y 是否相同；`compare`函数在单词 x 排在 y 之前时，得到值 -1 ，相同时得到值 0 ，之后时得到值 1 ；
 - 3，请自行查阅“折半查找法”，并从中获得帮助；
 - 4，查找的结果是：“没有发现”或者“发现”
 - 5，不允许抄袭！