

算法设计与分析

第1章 算法概述（1）



学习要点

- 算法在计算机科学中的地位
- 算法的概念
- 算法分析
- 算法的计算复杂性概念
- 算法渐近复杂性的数学表述



学习要点

- 算法在计算机科学中的地位
- 算法的概念
- 算法分析
- 算法的计算复杂性概念
- 算法渐近复杂性的数学表述

1 算法是计算机科学的主题

■ 70 年代前

- 计算机科学基础的主题没有被清楚地认清。

计算机科学就是
算法的研究

■ 70 年代

- Knuth 出版了《The Art of Computer Programming》以算法研究为主线，确立了算法为计算机科学基础的重要主题

- Knuth 于1974 年获得图灵奖。

Bill Gates: “如果你认为你是一名真正优秀的程序员, 请读Knuth的《计算机程序设计艺术》, 如果你能读懂整套书的话, 请给我发一份你的简历。”

■ 70 年代后

- 算法作为计算机科学核心推动了计算机科学技术飞速发展
- 算法和绝大多数的科学、商业和技术都是相关的

1 算法是计算机科学的主题

- 伟大的智者——Don E.Knuth , 中文名：高德纳(1938-)
 - 算法和程序设计技术的先驱者
 - 《计算机程序设计艺术》被誉为算法中“真正”的圣经
 - <http://www-cs-faculty.stanford.edu/~knuth/>

2 算法设计与分析的地位

■ 计算机科学的体系

■ 可计算理论

- 计算模型
- 可计算问题/不可计算问题
- 计算模型的等价性

■ 计算复杂性理论

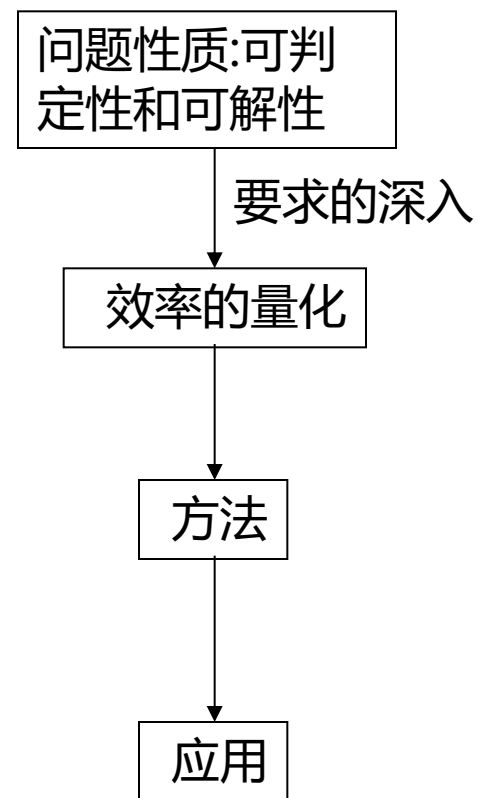
- 在给定的计算模型下研究问题的复杂性
- 复杂性问题的分类: $P=NP?$
- 抽象复杂性研究

■ 算法设计和分析

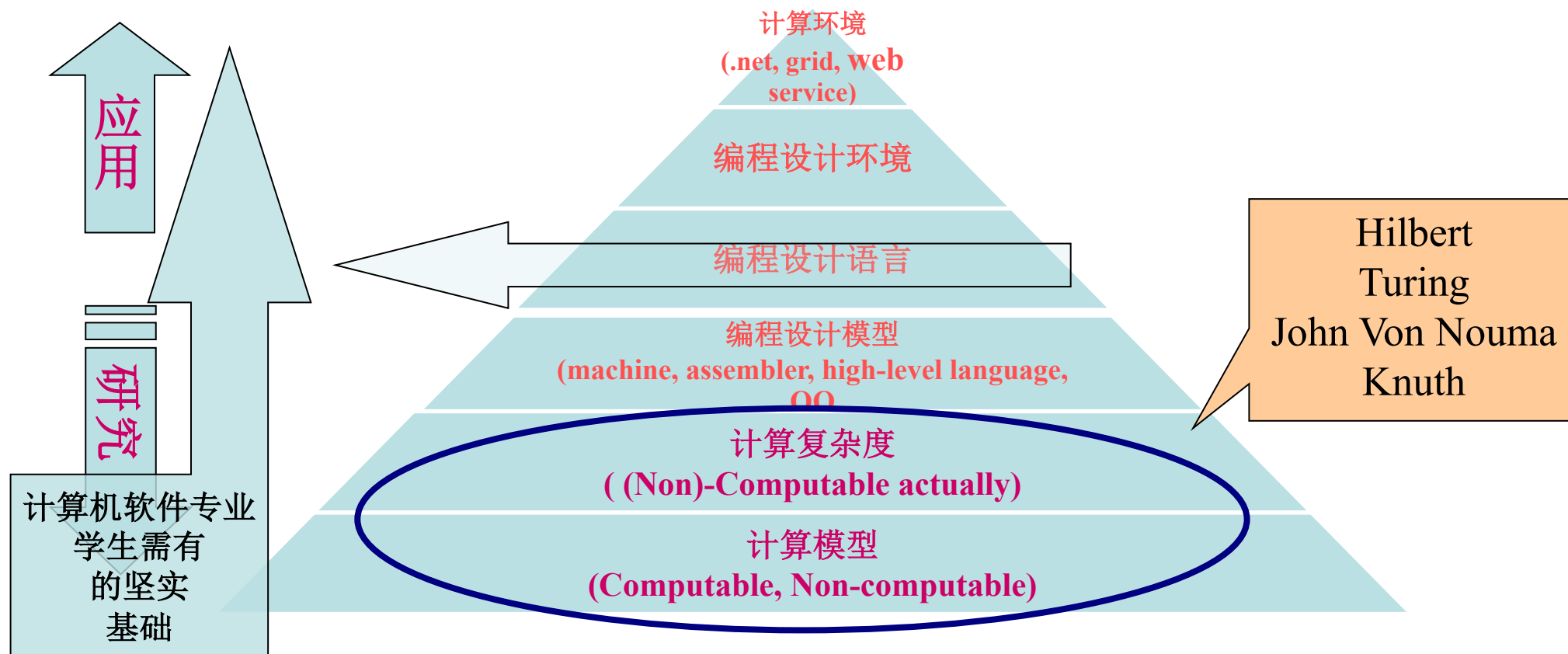
- 可计算问题的算法的设计与分析
- 设计算法的理论、方法和技术
- 分析算法的理论、方法和技术

■ 计算机软件

- 系统软件
- 工具软件
- 应用软件



3 算法与其它技术



3 算法与其它技术（续）

- 算法是当前计算机领域使用的核心技术
 - 是否具有扎实的算法知识和技术是区分skilled programmers 和the novices (初学者)的主要标准之一；
 - 有了良好的算法背景，你能do much, much more !!!



学习要点

- 算法在计算机科学中的地位
- 算法的概念
- 算法分析
- 算法的计算复杂性概念
- 算法渐近复杂性的数学表述

算法的概念

- 一个农夫带着一条狼、一头山羊和一篮蔬菜要过河,但只有一条小船
 - 乘船时,农夫只能带一样东西
 - 当农夫在场的时候,这三样东西相安无事.一旦农夫不在,狼会吃羊,羊会吃菜
- 请设计一个方案,使农夫能安全地将这三样东西带过河

算法的概念

- 数学中的算法问题：
 - “鸡兔同笼”，我国隋朝时期的数学著作《孙子算经》
 - 问题1:一个笼子里有一些鸡和兔，现在知道里面一共有35个头,94只脚，问鸡和兔各有多少只？
- 列方程解应用题步骤
 - 第1步：假设
 - 第2步：列方程
 - 第3步：求解
 - 第4步：作答

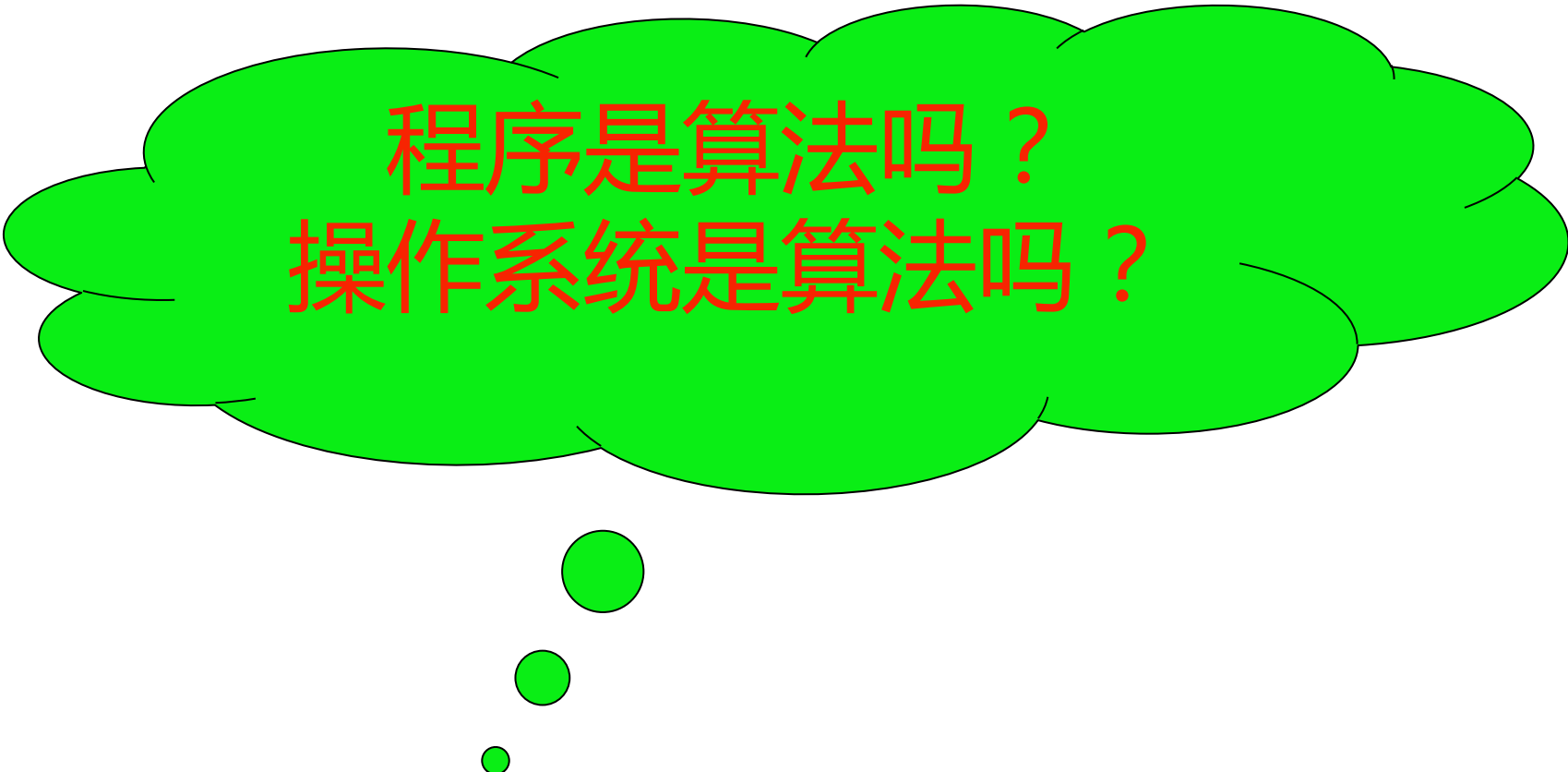
算法的概念

- 算法存在于各行业中

执行者	输入	输出
厨师	食材	美食
演奏家	乐谱	音乐
小朋友	废纸	小飞机
本科同学们	时间和精力	研究生入学通知

算法的概念★

- **算法**是指解决问题的一种方法或一个过程
- 算法是若干指令的有穷序列，满足性质：
 - **输入**：有外部提供的量作为算法的输入。一个算法有一个或多个输入,这些输入是在算法开始之前给出的量, 取自特定的对象集合
 - **输出**：一个算法产生一个或多个输出,这些输出是与输入具有某种特定关系的量
 - **确定性**：组成算法的每条指令是清晰，无歧义的。算法的每一种运算必须有确切的定义
 - 例如：一条指令： 5. 从 S_s 开始BFS遍历 G_i ，更新查询顶点 S_u 中顶点的查询距离；
 - **有限性**：
 - 算法中每条指令的执行次数是有限的
 - 执行每条指令的时间也是有限的
 - 一个算法总是在执行了有穷步骤运算之后中止



程序是算法吗？
操作系统是算法吗？

算法的概念

- 程序是算法用某种程序设计语言的**具体实现**
- 程序可以不满足算法的“有限性”性质
 - 例如**操作系统**，是一个在无限循环中执行的程序，因而不是一个算法
 - 操作系统的各种任务可看成是单独的问题，每一个问题由操作系统中的一个子程序通过特定的算法来实现。该子程序得到输出结果后便终止

算法的概念

- 算法是问题的程序化解决方案
- 算法设计技术是用算法解题的一般性方法,用于解决不同计算领域的多种问题.

问题的陈述远比它的解法重要；得解只要有数学的或实验的技巧就行。提出新的疑问、新的可能，从新的角度看老问题，需要创造性的想象力并且标志着科学的真正进步。

A.爱因斯坦

问题的定义 ✨

- “问题” 定义了**输入**和**输出**的关系
- “问题” 的描述： 一组**集合对**之间的关系
 - 设Input和Output是两个集合, 则**一个问题**是一个关系 $R \subseteq \text{Input} \times \text{Output}$
 - Input称为问题R的输入集合, Input的每个元素称为R的一个**输入**
 - Output称为问题R的输出或结果集合, Output的每个元素称为R的一个**结果**
- 例如: 过河问题
 - **输入**：农夫, 羊, 狼, 菜, 河这边, 约束条件
 - **输出**：到河那边

问题的定义

■ 例. SORT问题定义如下：

■ 输入集合

$\text{Input} = \{ \langle a_1, a_2, \dots, a_n \rangle \mid a_i \text{是整数} \}$

■ 输出集合

$\text{Output} = \{ \langle b_1, b_2, \dots, b_n \rangle \mid b_i \text{是整数}, b_1 \leq b_2 \leq \dots \leq b_n \}$

■ 问题

$\text{SORT} = \{ (\langle a_1, \dots, a_n \rangle, \langle b_1, \dots, b_n \rangle) \mid \langle a_1, \dots, a_n \rangle \in \text{Input}, \langle b_1, \dots, b_n \rangle \in \text{Output}, \{a_1, \dots, a_n\} = \{b_1, \dots, b_n\} \}$

例子：

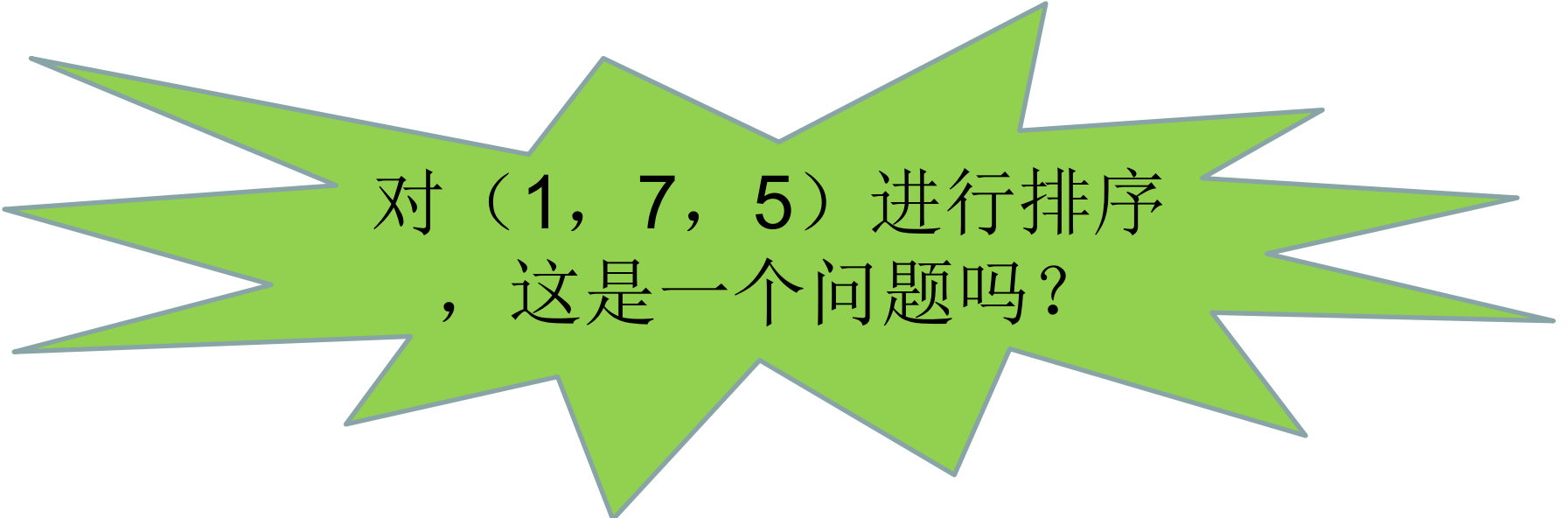
[JCS-团队生成](#)

[SIGKDD2017](#)

[KBS—竞争影响力](#)



问题的定义

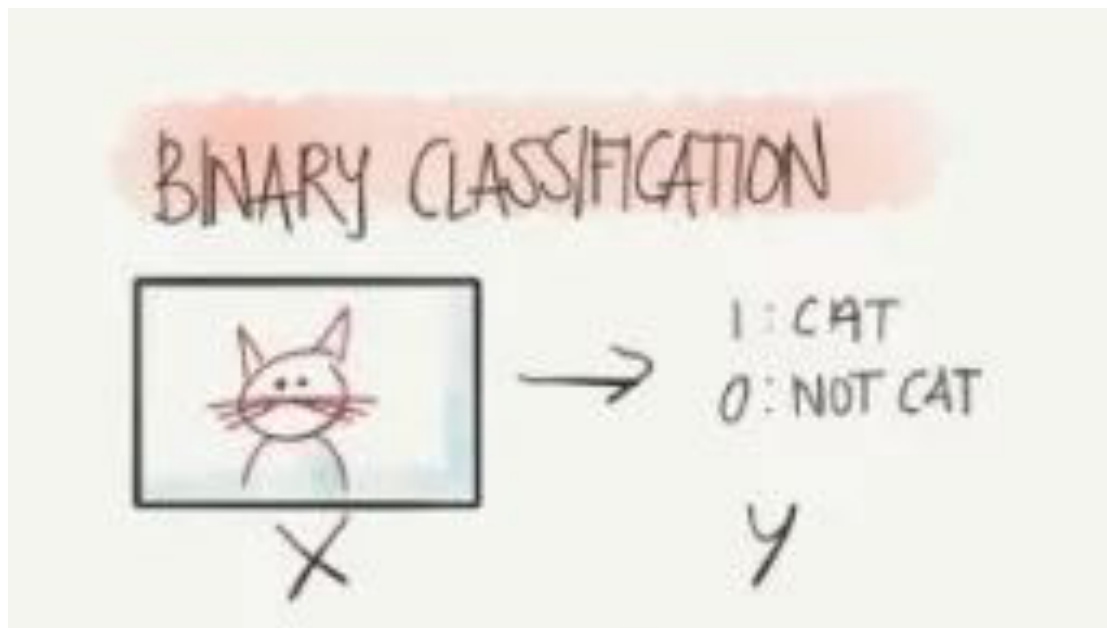


对 $(1, 7, 5)$ 进行排序
，这是一个问题吗？

问题实例

■ 二分类问题

- 输入：图像
- 输出：猫(1) 或非猫(0)



问题实例

- 算法的输入确定了该算法所解问题的一个实例
 - 问题R的一个实例是一个二元组，问题R是一个二元组的集合
 - Eg: $\langle (1,7,5), (1,5,7) \rangle$
- 注意
 - 一个算法面向一个**问题**，而不是仅求解一个问题的一個或几个实例。
 - 算法认为是一种解决问题的通用方法

思想：类与实例，关系与关系实例，特殊与普遍

问题的求解过程

- 步骤1: 问题的陈述
 - 用科学规范的语言,对所求解的问题做准确的描述
- 步骤2: 建立数学模型
 - 通过对问题的分析,找出其中的所有操作对象及操作对象之间的关系并用数学语言加以描述
- 步骤3 : 算法设计
 - 根据数学模型设计问题的计算机求解算法
- 步骤4 : 算法的正确性证明
 - 证明算法对一切合法输入均能在有限次计算后产生正确输出
- 步骤5 : 算法分析
 - 对执行该算法所消耗的计算机资源进行估算.
- 步骤6 : 算法的程序实现
 - 将算法正确地编写成机器语言程序

案例：考研问题求解

■ 求解过程：

■ Step 1 问题描述

- 输入：现阶段所学课程，清华大学招生简章
- 输出：考上研究生（各门课程上线，老师招你）

■ Step 2 数学模型

- 操作对象与关系：我，课程，大学，导师，考试

■ Step 3 设计：缺什么？补什么？怎么补？怎么找老师？

■ Step 4 正确性证明

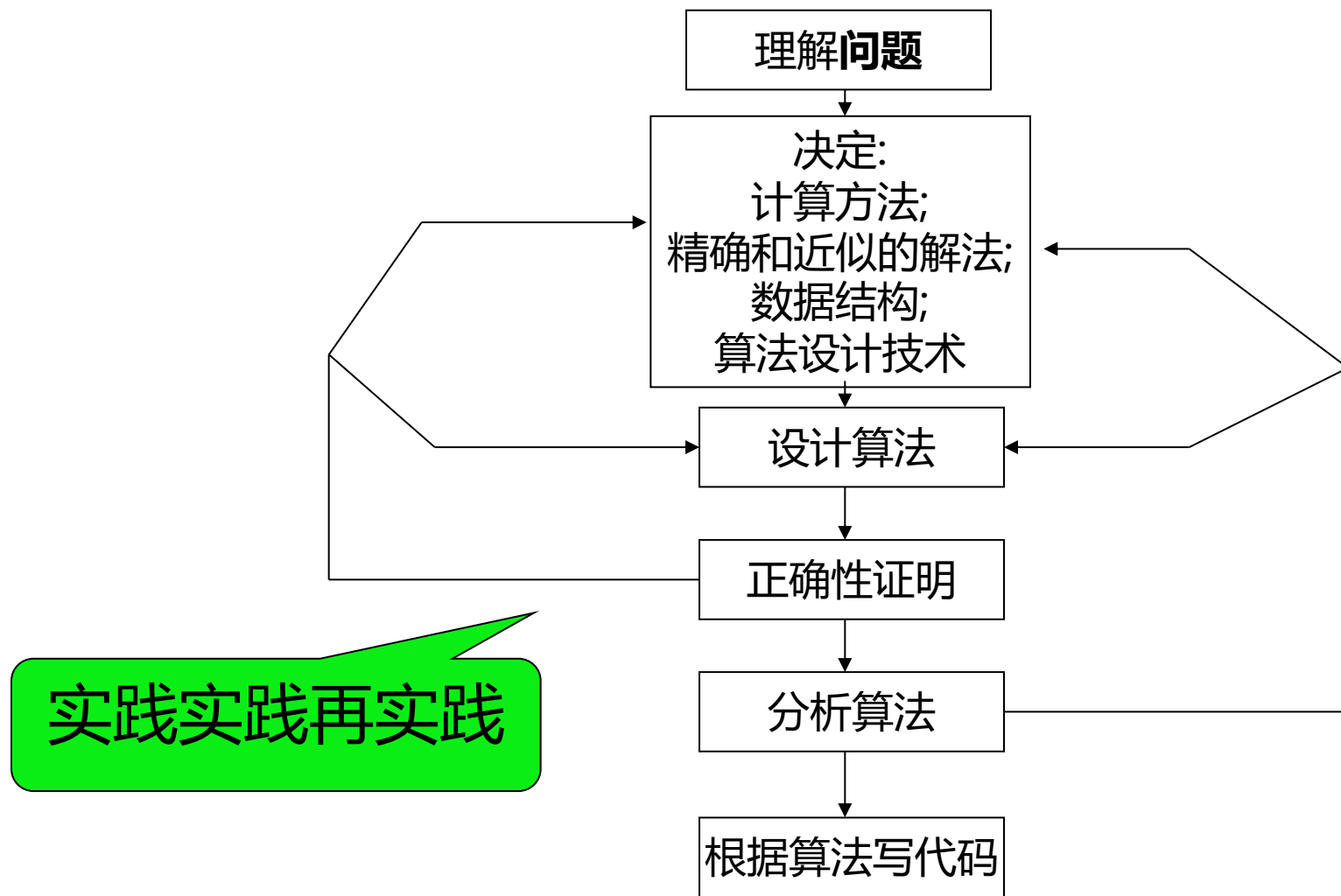
■ Step 5 分析：时间空间的最大利用，在哪里复习？经济能力

■ Step 6 实现

■ 算法：

- 步骤1，2，3...

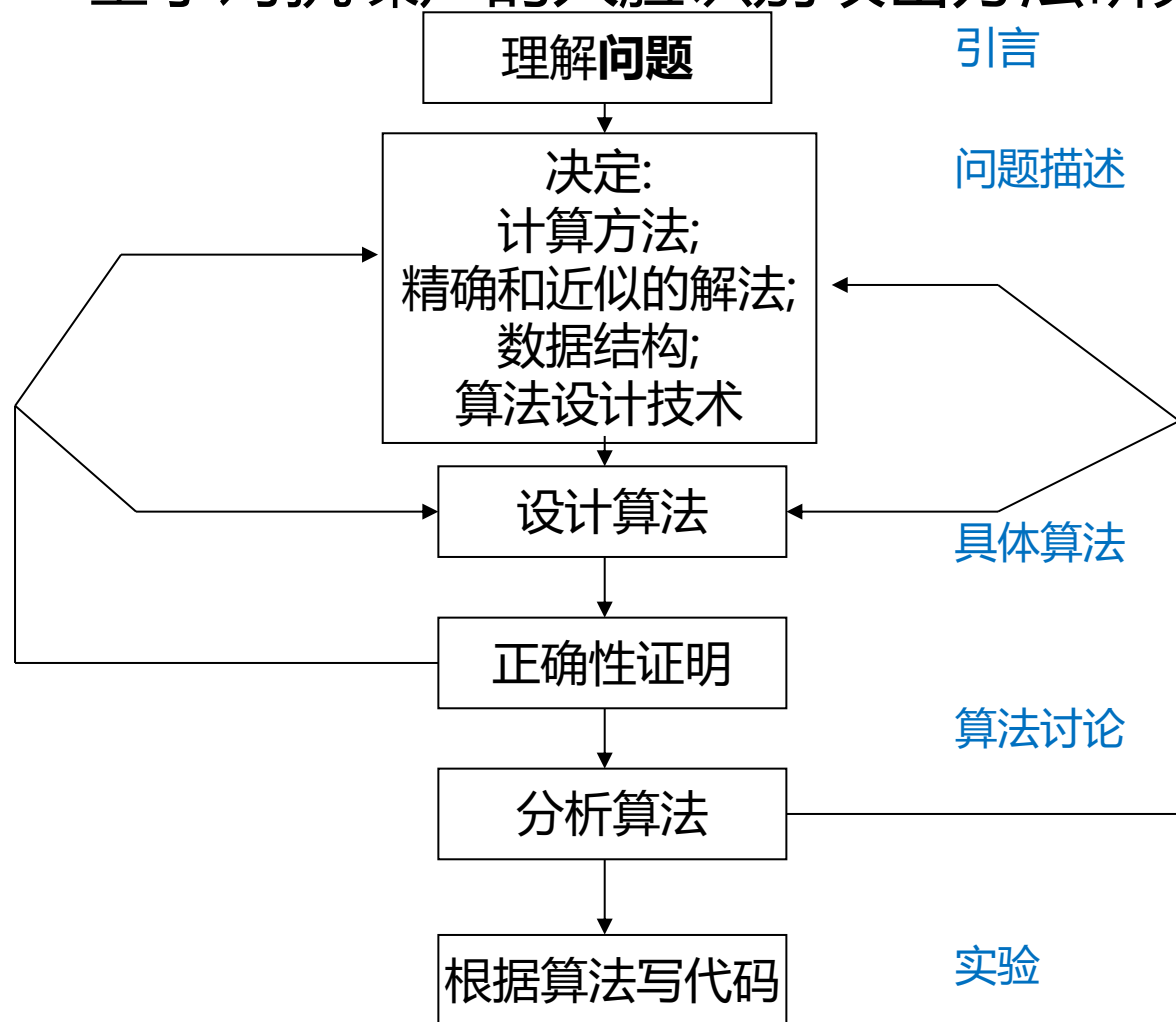
算法的设计和分析过程



毕业设计论文案例

■ 2021年校优秀

■ 基于对抗噪声的人脸识别攻击方法研究



▼ 第 1 章 绪论

1.1 研究背景

▶ 1.2 国内外研究现状

1.3 存在的问题与研究动机

1.4 主要工作内容

1.5 论文结构

▶ 第 2 章 人脸识别模型与攻击相关理论与技术

▼ 第 3 章 FGSM-Augmented (FGSM-A)对抗攻击方法

▶ 3.1 问题的提出

▶ 3.2 FGSM-Augmented (FGSM-A)攻击方法

▶ 3.3 实验设计

▶ 3.4 人脸识别模型训练与测试

▶ 3.5 实验结果分析

3.6 攻击方法讨论与分析

3.7 本章小结

▼ 第 4 章 BIM-Augmented (BIM-A)对抗攻击方法

4.1 问题的提出

▶ 4.2 BIM-Augmented (BIM-A)对抗攻击方法

4.3 Score-CAM注意力区域可视化方法

4.4 实验设计

▶ 4.5 实验结果分析

4.6 攻击方法讨论与分析

4.7 本章小结

▼ 第 5 章 基于注意力区域的Attention-BIM对抗攻击方法

5.1 问题的提出

▶ 5.2 Attention-BIM对抗攻击方法

5.3 实验设计

▶ 5.4 实验结果分析

算法书写格式 ✨

- 算法名称：农夫过河
- 输入：农夫，羊，狼，菜，河这边，约束条件
- 输出：到河那边
- 步骤1 农夫把🐑带过去
- 步骤2 空手回来
- 步骤3 农夫把🌊带过去
- 步骤4 农夫把🐑带回来
- 步骤5 农夫把菜带过去
- 步骤6 农夫空手回来
- 步骤7 农夫把🐑带过去

算法书写格式 ★

- 算法名称：CrossRiver
- 输入：
 - $\langle \text{farmer}, \text{sheep}, \text{wolf}, \text{veg} \rangle = \langle 0, 0, 0, 0 \rangle$, 不允许出现 $\langle 0, 1, 0, 1 \rangle$ 和 $\langle 0, 1, 1, 0 \rangle$
- 输出： $\langle 1, 1, 1, 1 \rangle$
- 步骤1 moveSheep()
- 步骤2 back()
- ...
- 结束

其他例子：[KBS—竞争影响力](#)

[SIGKDD2017-2](#)

算法格式：伪码

- 伪代码

- Knuth, Cormon, Kleinberg& Tardos
- 侧重于可读性的伪编程语言，不具备执行能力
- 很多方面类似于 C, Pascal, 或 Java , Python

- Python

- 可读性好，可被视为一种具有执行力的伪代码



算法格式：伪码

■ 伪代码和真实代码的区别

- 伪代码描述给定算法时清晰、准确
- 伪代码不用考虑太多技术细节，如软件工程方面的问题（ data abstraction, modularity, error handling)
- 用伪代码可以体现算法的本质
- 永远不会过时

算法格式：伪码

INSERTION-SORT (A, n) // $A[1 \dots n]$

Input: A, n ;

Output: A ;

Step:

```
1   for  $j \leftarrow 2$  to  $n$  {  
2        $key \leftarrow A[j]$   
3        $i \leftarrow j - 1$   
4       while  $i > 0$  and  $A[i] > key$  do  
5            $A[i+1] \leftarrow A[i]$   
6            $i \leftarrow i - 1$   
7        $A[i+1] \leftarrow key$   
   }
```

■ 算法格式：伪码

■ if statements

If condition then
 some action
End If

or

If condition then
 some action

算法格式：伪码

■ If...else statements

```
If condition then
    some action
else
    alternative action
End If
```

or

```
If condition then
    some action
else
    alternative action
```


算法格式：伪码

■ The for loop

```
For i=1 to n do  
    some action  
End For
```

or

```
For i=1 to n do  
    some action
```

算法格式：伪码

- The while loop

```
While condition do  
    some action  
End While
```

or

```
While condition do  
    some action
```

算法格式：伪码

- The do-until loop

<p>Do</p> <p> some action</p> <p>Until condition</p>

算法格式：伪码

- Assignment

Variable \leftarrow value

算法格式：伪码

■ Comparison

Variable = value

Variable > value

Variable < value

Variable \geq value

Variable \leq value

Variable \neq value

算法格式：伪码

- Calling a procedure

Call **procedure (...)**

问题的分类

■ 简单问题

- 存在**多项式**复杂度的算法可以解决 $P(n) = a_0 + a_1n + \cdots + a_{d-1}n^{d-1} + a_dn^d$
($a_d > 0$)

■ 复杂问题

- 只可能具有**指数复杂度**的时间或空间的算法,已经证明不可能存在多项式算法的难题

$$P(n) = a^n (a > 0 \text{ 且 } a \neq 1)$$

- 不知道是否存在多项式算法的问题

■ 不可求解的问题

■ 对于大数据问题

- 多项式时间不再是大数据计算问题易解性的标准
- 多项式时间算法也不再是大数据问题的有效求解算法

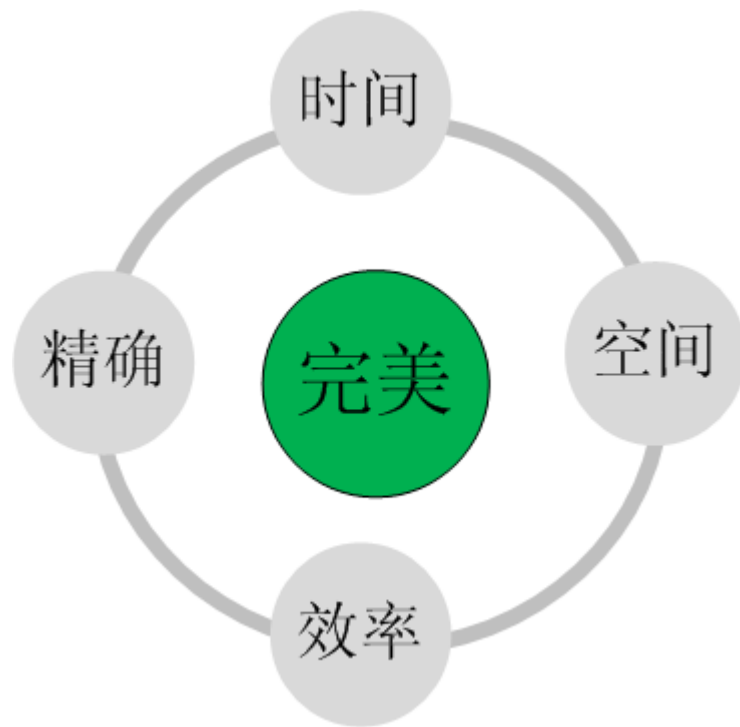
算法的应用

■ 两个公共特征

- 有很多候选的解决方案,找到真正需要的解决方案不太容易
- 有着实际的应用
 - 最短路径问题:公路, 铁路网, 网络路由

■ 算法的追求

- 没有最好, 只有更好





算法设计技术

- 直接法/枚举法/蛮力法
- 分治法
- 动态规划
- 贪心技术
- 回溯
- 分支限界法
- 其他



小结

- 主要内容
 - 了解算法在计算机科学中的地位
 - 掌握算法的概念和特征
- 重点
 - 算法的概念和特征
 - 算法的书写格式
 - 问题的定义