

第1节 问题与算法

本节知识

- ◆ 生活中的问题与算法
- ◆ 算法的特征及设计描述

本节活动

- ◆ “指挥机器”与“引领他人”的相似性讨论
- ◆ 算法五大特征的必要性讨论
- ◆ 绘制求解绝对值的算法流程图

随着计算机技术的发展，人们解决问题的方式也在不断地发生变化。人们越来越多地利用计算机领域的概念、思想和方法解决现实生活中的问题，并逐渐形成问题分解、抽象建模、设计算法、编写程序的解决模式。

一、生活中的问题与算法

人类社会之所以进步与发展，是因为人们在日常生活中一直不断地发现、提出各种问题，并积极寻求解决问题所需的各种策略、方法与途径。在运用计算机解决问题的实践过程中，人们逐步形成了算法的概念。比如寝室墙上贴有提示学生良好生活作息的时间表，茶叶包装袋上印有提示消费者科学泡茶的步骤，网购平台提供指导用户正确操作的方法等。

1. 引领问题解决的指令表达

人们在生活中为了引领他人精准地完成特定任务或解决指定问题，通常会采用各种恰当的方式来表达自己的指令，如图 2.1.1 所示。

春夏作息时间表

2024年3月1日起实行

内容	时间	内容	时间	内容	时间
起床	6:20	上学	7:10	散步	18:35—18:55
洗漱	6:25—6:40	放学	17:50	学习辅导	19:10—21:30
晨操 晨读	6:40—6:55 下雨时晨读	活动 洗澡	17:50—18:20 按顺序洗澡	洗漱	21:30—21:40
早餐	6:55—7:10	晚餐	18:20—18:35	就寝	21:50

(a) 作息时间表

泡/茶/方/法



撬茶 用茶针从整块茶中撬取 10g 左右的茶叶



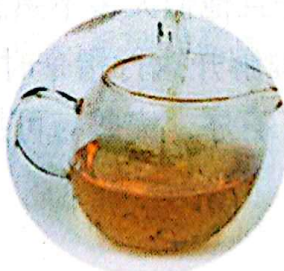
投茶 将茶叶投入已经温好的杯中



醒茶 可使茶叶充分“苏醒”，以便后期冲泡迅速出味



泡茶 将沸水沿杯壁冲入杯中，第二泡时间不宜超过 10s，此后每泡时间递增 10s 即可



出汤 将茶水倒入公道杯中



分茶 将公道杯中的茶水按量分入品茶杯中

(b) 泡茶步骤

图 2.1.1 引领他人完成任务的方式



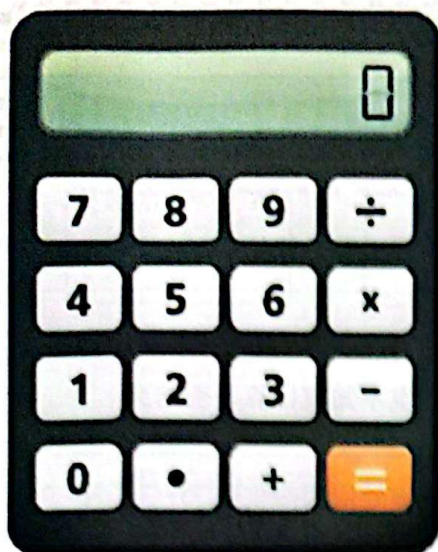
图 2.1.1 (续)

下指令者之所以采用表格、文字与图示等简单易懂方式来表达，是因为该问题需要被引领者自行解决，并且他们大多不熟悉问题解决所需的过程、步骤与安排。因此想要避免他人在理解并执行指令时出现错误或者无法完成任务，必须让指令表达尽可能地清晰、有序、完整与可执行。

2. 指挥机器执行的算法

如今使用机器替代人来解决问题，已经是一种非常普遍的现象。那些用来指挥机器解决问题或执行任务所需的一系列步骤，便是算法 (algorithm)。想要机器能够正确执行人给定的任务，就必须从它的视角来设计并描述可执行的算法。

比如让你去指挥简易计算器来完成指定的“ $5+6=?$ ”这一计算任务，就可以将执行步骤描述成如图 2.1.2 所示的内容。



- 第1步：接收按键“5”输入信号。
 第2步：接收按键“+”输入信号。
 第3步：接收按键“6”输入信号。
 第4步：接收按键“=”输入信号。
 第5步：内部运算“5+6”。
 第6步：显示器输出计算结果“11”。

图 2.1.2 指挥简易计算器计算“5+6=?”

问题讨论

请以图 2.1.2 所示的算法为例，讨论“指挥机器”与“引领他人”有哪些相似之处？

二、算法特征与设计描述

算法由一系列步骤组成，但一系列步骤组成的却不一定是算法，因为算法必须具备一些基本要求。

1. 算法的特征

一个科学、完整的算法，首先必须要有输入（或预设初始值）与输出这两个步骤，其次要规定所有的步骤必须是明确的、可以执行的并且不能无限执行的指令，从而确保任务能够在一定时间内完成。也就是说，算法要具备输入、输出、确定性、可执行性及有穷性这五大特征，其具体内涵与简易计算器的实例解析如表 2.1.1 所示。

表 2.1.1 算法特征及其解析

特征	具体内涵	实例解析
输入	必须有预设初始值或输入数值	必须要输入“5”“+”“6”“=”的数据，否则计算器不知道要计算什么
输出	必须有一个或多个数据输出	必须输出计算结果“11”，否则计算没有意义，用户也不知道计算是否结束
确定性	要有明确的执行对象及行为	确定要从按键处依次接收到“5”“+”“6”“=”的数据，否则不知道怎么做
可执行性	每个步骤都必须可以执行	要确保图 2.1.2 中的 6 个步骤都可以执行，否则无法完成任务
有穷性	执行步骤的次数必须有限	必须在有限的步骤内完成并结束任务，如果步骤无限，将永远无法结束

问题讨论

计算机的算法，为什么必须要具备这五大特征？

2. 算法的设计

在需要计算机解决的众多问题中，有些是属于常见的特定问题。如排序、查找与插入等问题，人们很早就对它们开展研究，并提出多种可靠、高效的算法。当我们再次遇到这类问题时，就可以采用先学习再评价的方式来从中挑选，而无须再另行设计。

还有一些是属于非特定或是个性化的问题，则通常需要我们自行设计算法。简单的算法设计过程，如图 2.1.3 所示。一般是先从理解问题入手，即要根据解决问题的目标，来明确输入、输出与约束条件。以图 2.1.2 中的算法为例，向简易计算器输入的是“5”“+”“6”和“=”，其输出的是计算结果，约束条件是必须依次接收输入的“5”“+”“6”和“=”。然后再用适当方式对解决该问题的步骤进行描述。最后还要通过编写代码方式，来对算法进行实践检

验、评估与优化等。

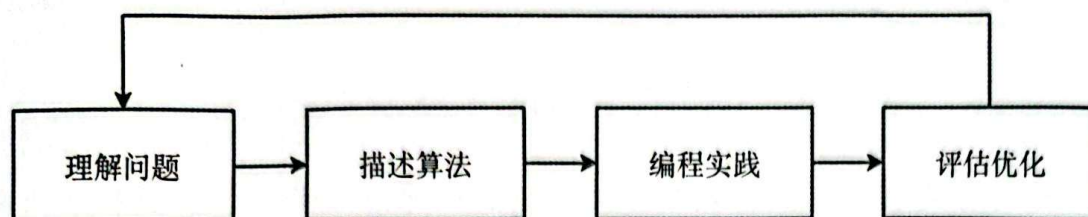


图 2.1.3 简单的算法设计过程

3. 算法的描述

跟引领他人解决日常生活问题的指令表达类似，计算机算法也有一些约定成俗的描述方式。用以保障你所描述的内容，能够被他人所理解，并促进算法自身的交流与提升。常用来描述算法的方式，有自然语言、流程图与伪代码等。

这三种描述方式，有各自的优势与不足。以计算圆面积算法为例，如图 2.1.4 所示。用自然语言描述，具有简单、通俗易懂的优点，但也容易产生歧义而给后续编写代码带来不必要的困扰，比如“计算面积”这一步，就没有给出具体的计算公式；用流程图描述则有清晰、直观与简洁的优点，虽然画图比较费时，但对专业性要求不高，因而较受人欢迎；而伪代码则是一种近似程序代码的算法表达方式，它方便程序员快速编程实现，但专业性要求相对较高，比如算法中的每条指令，都必须占用一行并用编号来表示步骤等。

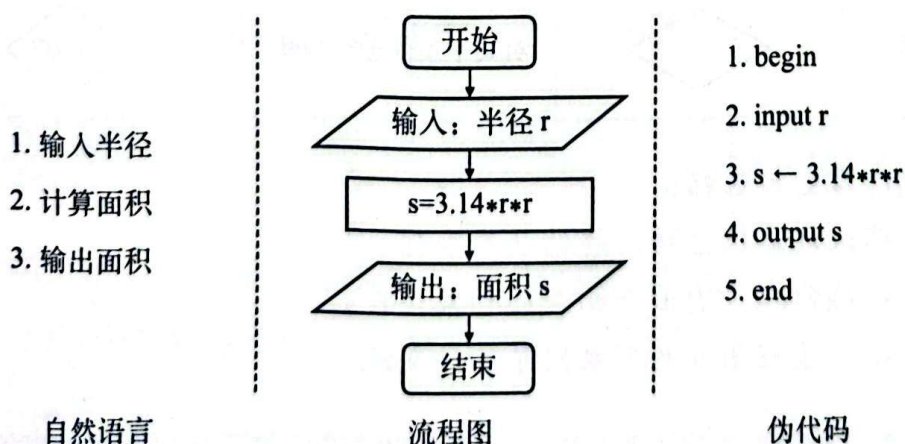


图 2.1.4 计算圆面积的三种算法描述



实践活动

用流程图描述求解绝对值的算法

绝对值是指一个数在数轴上的对应点到原点的距离，正数和0的绝对值是它本身（即 $|x|=x$ ），而负数的绝对值则是它的相反数（即 $|x|=-x$ ）。如果让计算机来自动求解，则必须设计相应的算法。

请用流程图描述计算机求解一个数的绝对值的算法。流程图的符号如表2.1.2所示。

表 2.1.2 流程图的符号

名称	符号	意义	范 例
起止框		程序的开始或结束	开始 结束
处理框		对数据的处理	$r \leftarrow 5$
流程线		流程的进行方向	
输入/输出框		数据的输入或输出	输入运算数 输出计算结果
判断框		对某个条件进行判断	$r \geq 0?$

实践内容主要包括：

- (1) 确认计算机自动求解的基本步骤。
- (2) 明确输入、输出与相关的约束条件。
- (3) 绘制流程图并与邻桌同学进行交流。

生活经验告诉我们，解决同一个问题，往往会有多种方案。比如想要从贵阳前往遵义，就有如表2.1.3所示的多种出行方案，并且它们的耗时与出行

成本各不相同。如果你对时间与出行成本比较在意，那么火车将是一种最优的选择。

表 2.1.3 从贵阳到遵义的几种出行方案

出行方案	耗 时	出行成本
汽车（出租车，非高速）	约 3 小时	最高
汽车（自驾，高速）	约 2 小 15 分钟	较高
火车（高铁）	约 2 小时	较低
自行车	约 12 小时	很低

当你选择运用计算机解决问题时，也同样会面临算法选择的挑战，因此需要对相关的各种算法进行评价。评价的主要指标有正确性、可读性、健壮性、高效性与低存储性等。在这些指标中，高效性因容易被人们直观感受而备受关注，它主要体现在用算法指挥计算机的执行次数方面。

以如图 2.1.5 所示的两种算法为例，在实现运用计算机解决“在 1 ~ 50 范围内，猜中 50 这个数字”这个问题过程，用“折半猜测”算法只要执行 6 次就猜中，而用“逐个猜测”算法则需执行 50 次。因此在解决该问题上，算法 2 要比算法 1 来得高效。

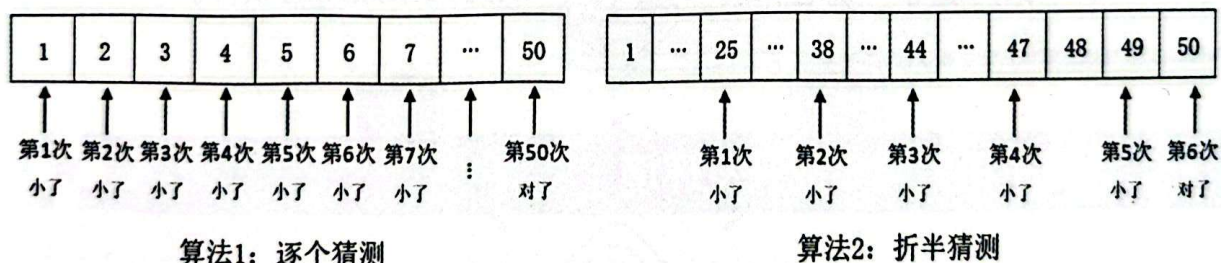


图 2.1.5 “猜数游戏”的不同算法比较

项目实施

小清团队准备使用流程图来描述要开发的软件的算法。
你会采用哪种方式描述算法，请填写在下面的横线上。
