

数据结构与算法

引言



张晓平

武汉大学数学与统计学院



2017 年 9 月 28 日

目录

1. 快速开始
2. 什么是计算机科学
3. 什么是编程
4. 为什么要学习数据结构和抽象数据类型
5. 为什么要学习算法

1. 快速开始

2. 什么是计算机科学

3. 什么是编程

4. 为什么要学习数据结构和抽象数据类型

5. 为什么要学习算法

- ▶ 计算技术的变化为计算机科学家提供了越来越多的工具和平台。
- ▶ 计算机的快速发展，诸如快速处理器、高速网络和大容量存储器，已经让计算机科学家陷入高度复杂螺旋中。

在所有这些快速演变中，有一些基本原则会保持不变。计算机科学关注用计算机来解决问题。

问题以及解决方案的复杂性可能会掩盖求解过程中的一些基本思想。

快速开始

简要回顾计算机科学、算法和数据结构所必须适应的框架，研究其原因，以及如何理解它们以有助于更好的解决问题。

1. 快速开始

2. 什么是计算机科学

3. 什么是编程

4. 为什么要学习数据结构和抽象数据类型

5. 为什么要学习算法

什么是计算机科学

所谓计算机科学，即

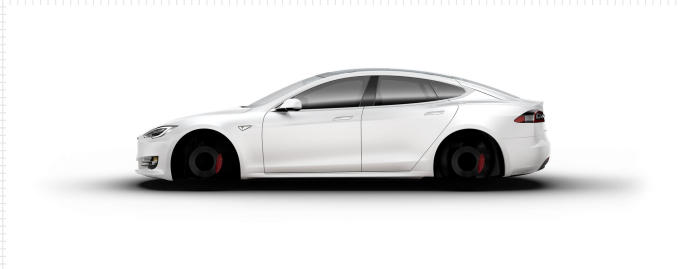
- ▶ 研究问题
- ▶ 解决问题
- ▶ 生成解决问题的方案

给定一个问题，目标是开发算法，编写程序，用于解决可能出现的问题。

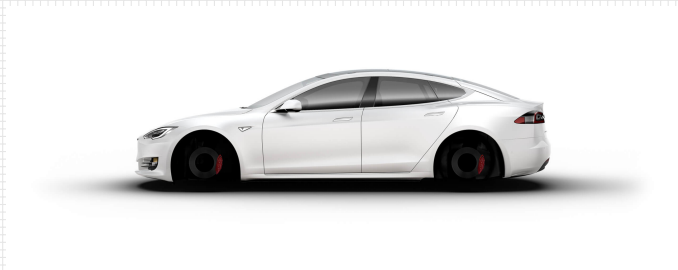
什么是计算机科学

计算机科学，因涉及问题解决过程本身，是关于抽象的研究。抽象使我们能从逻辑视角和物理视角来分别看待问题及其解决方案。

什么是计算机科学

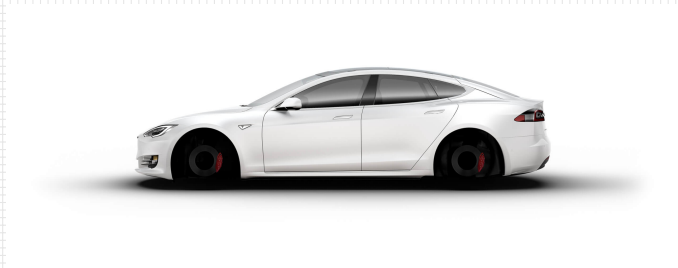


什么是计算机科学



- ▶ 作为司机，你会与车有一些互动，如上车、插钥匙、点火、换挡、制动、加速、转向、下车等。你只需要了解车一些基本功能，就可以操纵它。此时，你所看到的是汽车的“逻辑视角”，这些功能通常称为“接口”。

什么是计算机科学



- ▶ 作为司机，你会与车有一些互动，如上车、插钥匙、点火、换挡、制动、加速、转向、下车等。你只需要了解车一些基本功能，就可以操纵它。
此时，你所看到的是汽车的“逻辑视角”，这些功能通常称为“接口”。
- ▶ 作为修车师傅，看待汽车的视角就会截然不同。你不仅需要知道如何开车，还必须知道汽车的一些内部细节，比如说发动机如何工作、变速箱如何变速、温度如何控制等等。
这就是“物理视角”，细节发生在“引擎盖下”。

什么是计算机科学



什么是计算机科学



- ▶ 作为用户，你可以编写文档、收发邮件、上网冲浪、播放音乐、存储图像和玩游戏等，但你并不知道这些 APP 工作的细节。
此时，你是从逻辑或用户角度看待计算机。

什么是计算机科学



- ▶ 作为用户，你可以编写文档、收发邮件、上网冲浪、播放音乐、存储图像和玩游戏等，但你并不知道这些 APP 工作的细节。

此时，你是从逻辑或用户角度看待计算机。

- ▶ 作为计算机科学家、程序员、技术支持人员和系统管理员，你看待计算机的角度会截然不同。你必须知道操作系统如何工作、如何配置网络协议、如何编写控制功能的各种脚本。

总之，你必须能够控制底层的细节。

什么是计算机科学

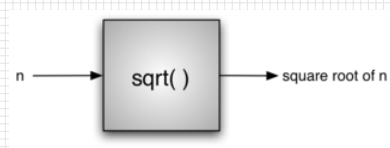
作为用户，你不需要知道细节，只需了解接口的工作方式。接口是用户与底层沟通的窗口。

什么是计算机科学

看一个抽象的例子—Python 数学模块。一旦导入模块，就可以执行计算

```
>>> import math
>>> math.sqrt(16)
4.0
```

你无需知道计算平方根的细节，只需知道 `sqrt` 函数的功能及其使用方式。这就像一个“黑盒子”，其接口可描述为：函数名、参数、返回值，其细节隐藏在内部。



1. 快速开始
2. 什么是计算机科学
3. 什么是编程

4. 为什么要学习数据结构和抽象数据类型
5. 为什么要学习算法

什么是编程

编程是将算法转换为程序语言的过程，以便能被计算机所执行。

- ▶ 首先要有解决方案，亦即算法；
- ▶ 然后在选择合适的编程语言实现算法。

计算机科学不研究编程，但编程却是计算机科学家的重要能力。编程通常是解决方案的表达方式。

什么是编程

- ▶ 算法描述了依据实际问题所生成的解决方案和产生预期结果所需要的一套步骤。
- ▶ 编程语言必须提供一种表示方法来表示对应的过程和数据。为此，它提供了**控制结构**和**数据类型**。

什么是编程

控制结构允许以方便而明确的方式表示算法步骤。至少，算法需要执行顺序处理、决策选择和控制迭代。只要语言提供这些基本语句，它就可以表达算法。

什么是编程

在计算机中，所有数据项都由一串一串的二进制数表示。为了让这些二进制串有意义，就需要有**数据类型**。

- ▶ 数据类型为二进制数据提供解释，以便我们能够根据实际问题来思考数据。这些底层的内置数据类型（有时称为原始数据类型）为算法开发提供了基础。

什么是编程

在计算机中，所有数据项都由一串一串的二进制数表示。为了让这些二进制串有意义，就需要有**数据类型**。

- ▶ 数据类型为二进制数据提供解释，以便我们能够根据实际问题来思考数据。这些底层的内置数据类型（有时称为原始数据类型）为算法开发提供了基础。例如，大多数编程语言提供整数类型。内存中的二进制数据可解释为整数，并且给予一个与整数（如 23, 654 和 -19）相关联的含义。

什么是编程

在计算机中，所有数据项都由一串一串的二进制数表示。为了让这些二进制串有意义，就需要有**数据类型**。

- ▶ 数据类型为二进制数据提供解释，以便我们能够根据实际问题来思考数据。这些底层的内置数据类型（有时称为原始数据类型）为算法开发提供了基础。例如，大多数编程语言提供整数类型。内存中的二进制数据可解释为整数，并且给予一个与整数（如 23, 654 和 -19）相关联的含义。
- ▶ 此外，数据类型还提供数据项所参与操作的描述。对于整数，提供诸如加法、减法和乘法的操作。

什么是编程

然而，通常我们遇到的困难是问题及其解决方案非常复杂。由语言提供的简单的结构和数据类型，虽然可以表示复杂的解决方案，但在实际中却不好用。我们需要一些方法控制这种复杂性，以助于形成更好的解决方案。

1. 快速开始
2. 什么是计算机科学
3. 什么是编程

4. 为什么要学习数据结构和抽象数据类型

5. 为什么要学习算法

为什么要学习数据结构和抽象数据类型

为管理问题的复杂性及解决过程，计算机科学家使用抽象使他们能够专注于“大局”而不会迷失在细节中。

通过对问题进行建模，我们能够更好和更有效地解决问题。

这些模型允许我们以更加一致的方式来描述我们的算法。

为什么要学习数据结构和抽象数据类型

- ▶ **过程抽象**：隐藏特定函数的细节，以允许用户或客户端在高层查看它。

为什么要学习数据结构和抽象数据类型

- ▶ **过程抽象**：隐藏特定函数的细节，以允许用户或客户端在高层查看它。
- ▶ **数据抽象，即抽象数据类型 (ADT)**：对数据和允许操作的逻辑描述，不用考虑如何实现它们。

这意味着我们只关心数据表示什么，而不关心它最终将如何构造。通过提供这种级别的抽象，我们围绕数据创建一个封装。通过封装实现细节，我们将它们从用户的视图中隐藏。这称为信息隐藏。

为什么要学习数据结构和抽象数据类型



图: 展示了 ADT 是什么以及如何操作。用户与接口交互, 使用抽象数据类型指定的操作。抽象数据类型是用户与之交互的 shell。实现隐藏在更深的底层。用户不关心实现的细节。

为什么要学习数据结构和抽象数据类型

ADT 的实现要求我们使用一些程序构建和原始数据类型的集合来提供数据的物理视图。

“逻辑”与“物理”两个视角的分离，允许我们将问题定义复杂的数据模型，而不给出关于模型如何实际构建的细节。这提供了独立于实现的数据视图。

由于通常有许多不同的方法来实现抽象数据类型，所以这种实现独立性允许程序员在不改变数据的用户与其交互的方式的情况下切换实现的细节。

用户可以继续专注于解决问题的过程。

1. 快速开始
2. 什么是计算机科学
3. 什么是编程

4. 为什么要学习数据结构和抽象数据类型

5. 为什么要学习算法

为什么要学习算法

计算机科学家经常通过经验学习。我们通过看别人解决问题和自己解决问题来学习。接触不同的问题解决技术，看不同的算法设计有助于我们面对下一个具有挑战性的问题。通过思考许多不同的算法，我们可以开始开发模式识别，以便下一次出现类似的问题时，我们能够更好地解决它。

为什么要学习算法

算法通常彼此完全不同。例如，对于 `sqrt`，完全可能存在很多不同的方式来实现其细节。

自然地，我们建议使用一个更高效，或者一个只是工作更快或使用更少的内存的算法。

为什么要学习算法

作为计算机科学家，除了需要具备解决问题的能力，还需要掌握解决方案的评估技术。

通常有很多方法来解决问题，所以当我们找到一个解决方案时，需要一遍又一遍地进行比较，然后决定它是否是一个好的方案。