3.1 数据结构分类

常见的数据结构包括数组、链表、栈、队列、哈希表、树、堆、图,它们可以从"逻辑结构"和"物理结构"两个维度进行分类。

3.1.1 逻辑结构:线性与非线性

逻辑结构揭示了数据元素之间的逻辑关系。在数组和链表中,数据按照一定顺序排列, 体现了数据之间的线性关系;而在树中,数据从顶部向下按层次排列,表现出"祖先"与 "后代"之间的派生关系;图则由节点和边构成,反映了复杂的网络关系。

如图 3-1 所示,逻辑结构可分为"线性"和"非线性"两大类。线性结构比较直观,指数据在逻辑关系上呈线性排列;非线性结构则相反,呈非线性排列。

- 线性数据结构:数组、链表、栈、队列、哈希表,元素之间是一对一的顺序关系。
- 非线性数据结构: 树、堆、图、哈希表。

非线性数据结构可以进一步划分为树形结构和网状结构。

- 树形结构: 树、堆、哈希表,元素之间是一对多的关系。
- 网状结构: 图,元素之间是多对多的关系。

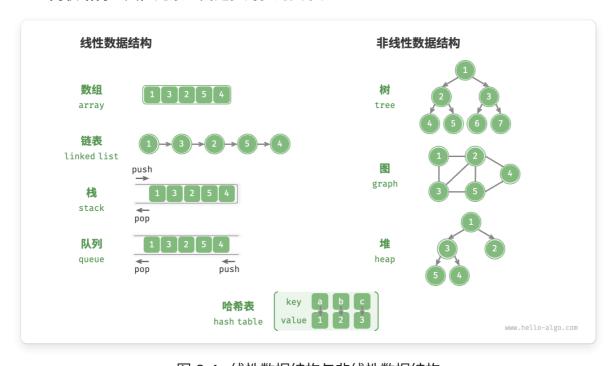


图 3-1 线性数据结构与非线性数据结构

3.1.2 物理结构:连续与分散

当算法程序运行时,正在处理的数据主要存储在内存中。图 3-2 展示了一个计算机内存条,其中每个黑色方块都包含一块内存空间。我们可以将内存想象成一个巨大的 Excel 表格,其中每个单元格都可以存储一定大小的数据。

系统通过内存地址来访问目标位置的数据。如图 3-2 所示,计算机根据特定规则为表格中的每个单元格分配编号,确保每个内存空间都有唯一的内存地址。有了这些地址,程序便可以访问内存中的数据。

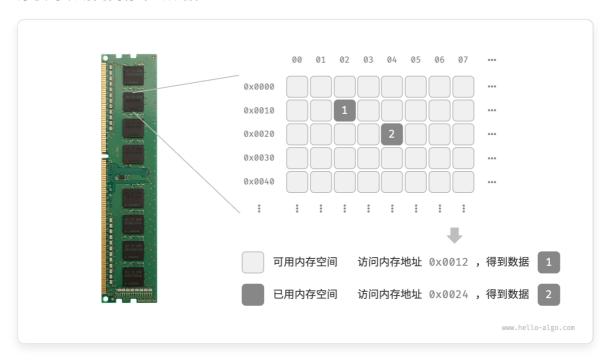


图 3-2 内存条、内存空间、内存地址



值得说明的是,将内存比作 Excel 表格是一个简化的类比,实际内存的工作机制比较复杂,涉及地址空间、内存管理、缓存机制、虚拟内存和物理内存等概念。

内存是所有程序的共享资源,当某块内存被某个程序占用时,则无法被其他程序同时使用了。**因此在数据结构与算法的设计中,内存资源是一个重要的考虑因素**。比如,算法所占用的内存峰值不应超过系统剩余空闲内存;如果缺少连续大块的内存空间,那么所选用的数据结构必须能够存储在分散的内存空间内。

如图 3-3 所示,**物理结构反映了数据在计算机内存中的存储方式**,可分为连续空间存储(数组)和分散空间存储(链表)。物理结构从底层决定了数据的访问、更新、增删等操作方法,两种物理结构在时间效率和空间效率方面呈现出互补的特点。

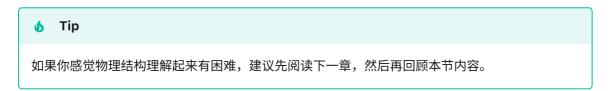


图 3-3 连续空间存储与分散空间存储

值得说明的是,**所有数据结构都是基于数组、链表或二者的组合实现的**。例如,栈和队列既可以使用数组实现,也可以使用链表实现;而哈希表的实现可能同时包含数组和链表。

- 基于数组可实现: 栈、队列、哈希表、树、堆、图、矩阵、张量(维度 ≥ 3 的数组)等。
- 基于链表可实现: 栈、队列、哈希表、树、堆、图等。

链表在初始化后,仍可以在程序运行过程中对其长度进行调整,因此也称"动态数据结构"。数组在初始化后长度不可变,因此也称"静态数据结构"。值得注意的是,数组可通过重新分配内存实现长度变化,从而具备一定的"动态性"。



上一页 下一页 **第 3 章 数据结构 3.2 基本数据类型 →**

欢迎在评论区留下你的见解、问题或建议