1、数据结构和算法——概念01

数据结构和算法

什么是数据结构

概念

官方定义：

数据结构是一门研究非数值计算的程序设计问题中的操作对象，以及它们之间的关系和操作等相关问题的学科。

我的理解：

程序设计 = 数据结构 + 算法

数据结构，顾名思义，就是数据之间的结构关系，或者理解成数据元素相互之间存在的一种或多种特定关系的集合。当然这些概念都是大学喜欢考的，我们没必要纠结于这个概念，有自己恰当的、并且可以为他人所接受的解释就可以。

常用结构

数组

在程序设计中，为了处理方便， 把具有相同类型的若干变量按有序的形式组织起来。这些按序排列的同类数据元素的集合称为数组。在C语言中， 数组属于构造数据类型。一个数组可以分解为多个数组元素，这些数组元素可以是基本数据类型或是构造类型。因此按数组元素的类型不同，数组又可分为数值数组、字符数组、指针数组、结构数组等各种类别。

栈

是只能在某一端插入和删除的特殊线性表。它按照先进后出的原则存储数据，先进入的数据被压入栈底，最后的数据在栈顶，需要读数据的时候从栈顶开始弹出数据（最后一个数据被第一个读出来）。

队列

一种特殊的线性表，它只允许在表的前端（front）进行删除操作，而在表的后端（rear）进行插入操作。进行插入操作的端称为队尾，进行删除操作的端称为队头。队列是按照“先进先出”或“后进后出”的原则组织数据的。队列中没有元素时，称为空队列。

链表

是一种物理存储单元上非连续、非顺序的存储结构，它既可以表示线性结构，也可以用于表示非线性结构，数据元素的逻辑顺序是通过链表中的指针链接次序实现的。链表由一系列结点（链表中每一个元素称为结点）组成，结点可以在运行时动态生成。每个结点包括两个部分：一个是存储数据元素的数据域，另一个是存储下一个结点地址的指针域。

树

是包含n（n>0）个结点的有穷集合K，且在K中定义了一个关系N，N满足 以下条件：

（1）有且仅有一个结点 K0，他对于关系N来说没有前驱，称K0为树的根结点。简称为根（root）。 （2）除K0外，K中的每个结点，对于关系N来说有且仅有一个前驱。

（3）K中各结点，对关系N来说可以有m个后继（m>=0）。

图

图是由结点的有穷集合V和边的集合E组成。其中，为了与树形结构加以区别，在图结构中常常将结点称为顶点，边是顶点的有序偶对，若两个顶点之间存在一条边，就表示这两个顶点具有相邻关系。

堆

在计算机科学中，堆是一种特殊的树形数据结构，每个结点都有一个值。通常我们所说的堆的数据结构，是指二叉堆。堆的特点是根结点的值最小（或最大），且根结点的两个子树也是一个堆。

散列表

若结构中存在关键字和K相等的记录，则必定在f(K)的存储位置上。由此，不需比较便可直接取得所查记录。称这个对应关系f为散列函数(Hash function)，按这个思想建立的表为散列表。

算法

算法的概念

是指解题方案的准确而完整的描述，是一系列解决问题的清晰指令，算法代表着用系统的方法描述解决问题的策略机制。

在我看来，算法就是求解一个问题所需要的步骤所形成的解决方法，每一步包括一个或者多个操作。无论是现实生活中还是计算机中，解决同一个问题的方法可能有很多种，在这N多种算法中，肯定存在一个执行效率最快的方法，那么这个方法就是最优算法。

算法的特性

算法具有五个基本特征：输入、输出、有穷性、确定性和可行性。

算法的设计要求

要设计一个好的算法，需要考虑以下4个特性

正确性

废话，谁会设计一个不能够解决问题的方法。

可读性

指算法无论是从设计思路上，还是从注释方面，都要能够保证算法是可读的，也就是可以被其他人员能够读懂的。其实也是废话，这是一个优秀的程序员必备的。

健壮性

通俗的讲，一个好的算法应该具有捕获异常/处理异常的能力。另外，对于测试人员的压力测试、边界值测试等刁难的测试手段，算法应该能够轻松的扛过去。

时间效率高和存储量低

这其实是两个概念，时间效率就是指的时间复杂度，存储量就是指的空间复杂度。