

入门——数据结构

数据结构

数据：是描述客观事务的符号，是计算机可以操作的对象。

数据元素：是组成数据的、有一定意义的基本单位。

数据项：一个数据元素可以由若干个数据项组成。数据项是数据不可分割的最小单位。

数据结构：是相互之间存在一种或多种特定关系的数据元素的集合。

逻辑结构和物理结构

逻辑结构

逻辑结构：是指数据对象中数据元素之间的相互关系。

1. 集合结构：集合结构中的数据元素除了同属于一个集合之外，它们之间没有其他任何关系。
2. 线性结构：线性结构中的数据元素之间是一对一的关系。
3. 树形结构：树形结构中的数据元素之间存在一种一对多的层次关系。
4. 图形结构：图形结构的数据元素是多对多的关系。

物理结构

物理结构(存储结构)：是指数据的逻辑结构再计算机中的存储形式。

1. 顺序存储

是把数据元素存放在地址连续的存储单元里，其数据间的逻辑关系和物理关系是一致的。

2. 链式存储

是把数据元素存放在任意的存储单元里，这组存储单元可以是连续的，也可以是不连续的。这样的存储关系不能表示其逻辑关系，因此需要用一个指针存放数据元素的地址，这样通过地址就可以找到相关联的数据元素的位置。

算法

算法是解决特定问题求解步骤的描述，在计算机中表现为指令的有限序列，并且每条指令表示一个或多个操作。

算法的特性

算法具有五个基本特性：输入、输出、有穷性、确定性和可行性。

1. 输入输出

算法具有零个或多个输入；算法至少有一个或多个输出。

2. 有穷性

指算法在执行有限的步骤之后，自动结束而不会出现无限循环，并且每一个步骤在可接受的时间内完成。

3. 确定性

算法的每一步都具有确定的含义，不会出现二义性。

4. 可行性

算法的每一步都能够通过执行有限次数完成。

时间复杂度

在进行算法分析时，语句总的执行次数 $T(n)$ 是关于问题规模 n 的函数，进而分析 $T(n)$ 随 n 的变化情况并确定 $T(n)$ 的数量级。

算法的时间复杂度，也就是算法的时间度量，记作： $T(n)=O(f(n))$ 。（ $f(n)$ 是问题规模 n 的某个函数）它表示随问题规模 n 的增大，算法执行时间的增长率和 $f(n)$ 的增长率相同，称作算法的渐进时间复杂度，简称时间复杂度。

推导大O阶方法

1. 用常数1取代运行时间中的所有加法常数；
2. 在修改后的运行次数函数中，只保留最高阶项；
3. 如果最高阶项存在且不是1，则去除与这个项相乘的常数，得到的结果就是大O阶。

常用时间复杂度所耗费的时间：

$$O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(n^3) < O(2^n)$$

空间复杂度

算法的空间复杂度通过计算算法所需的存储空间实现，算法空间复杂度的计算公式记作： $S(n)=O(f(n))$ ，其中 n 为问题的规模， $f(n)$ 为语句关于 n 所占存储空间的函数。