## 数据结构的一般概念

**逻辑结构：集合、线性、树形、图结构**

集合：各个元素同属于一个集合，别无其他关系 例如火锅，锅内的食材之间无关系，但都在锅中

线性：元素间一对一的关系，除了第一个元素和最后一个元素，其余元素均有前驱和后继

树形：数据间一对多的关系

图：数据间多对多的关系

**存储结构：顺序、链式、索引、散列结构**

顺序：把逻辑位置相邻的元素存储在物理位置也相邻的存储单元中 空间连续

链式：逻辑上相邻的元素在物理地址上可以不相邻，用指针表示下一个元素的存储地址

索引：用索引表来表示元素的存储地址

散列：哈希存储，通过散列表 （待扩充）

**抽象数据类型（ADT）：**

抽象数据组织及与其相关的操作。用数学化的语言定义数据的逻辑结构，定义运算。与具体实现无关。当我们定义了一个抽象数据类型的时候，我们确定了该类型的**逻辑结构**和**数据运算**，换而言之，当定义**抽象数据类型**就是定义了一个**数据结构**

例如，在统计学生信息时，经常使用姓名、学号、成绩等信息，我们可以定义这样的一个抽象数据类型student,它封装了姓名、学号、成绩三个不同类型的变量，这样操作student的变量就能够很方便的知道这些信息了。C语言中的结构体以及C++、java语言中的类等都是这种形式。

算法的特性：

有穷性：算法必须**有穷**（用有限步骤解决某个特定的问题），而程序可以是无穷的（比如说系统，可以永不停歇）

确定性：对于**相同的输入**必须有**相同的输出**

可行性：算法中描述的操作都可以通过已经实现的**基本运算执行有限次**来实现

输入：有**零个**或多个输入

输出：有**一个**或多个输出

优秀算法的特性：

正确性：能正确完成任务

可读性：**无歧义的描述**解决问题的步骤

健壮性：输入非法数据，算法能适当的做出反应

高效率：时间复杂度低

低存储量需求：空间复杂度低

**时间复杂度的函数大小对比**





