数据结构和算法

应用程序 = 算法 + 数据结构

数据结构 分为 逻辑结构 和 物理结构

逻辑结构是指 数据对象中数据元素之间的相互关系

物理结构是指 数据的逻辑结构在计算机中的存储形式

逻辑结构 有四种：

1. 集合结构：同属于一个集合，没有其他关系
2. 线性关系：一对一的关系
3. 树形结构：树形结构中的数据存在一种一对多的层次关系
4. 图形结构：数据对象中存在对多对的关系。

物理结构 硬盘 软盘 光盘 一般是用文件结构来描述

1. 顺序存储结构：是把数据对象存储在地址连续的存储单元，其数据间的逻辑关系和物理关系是一致的。如 数组
2. 链式存储结构：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

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。

输入（0或多个）

输出（1个或多个）

有穷性（可以执行完）

确定性

正确性

基本要求：

没有语法错误

合法输入有正确的输出

非法的输入可以有满足满足规格的说明

故意刁难的输入 也可以 有正确的说明

可读性

健壮性

时间效率高 和 存储量低

计算机的度量方法：时间复杂度 和 空间复杂度

算法采用的策略和方案

编译产生的代码质量

问题的输入规模

机器执行指令的速度

算法的时间复杂度：在进行算法分析时，语句总得执次数T(n)是关于问题规模n的函数，进而确定T(n)随n的变化情况，并确定T(n)的数量级。记作：T(n) = O(f(n))。他表示随问题规模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，简称为时间复杂度。其中f(n)是问题规模n的某个函数。（执行次数=时间复杂度）

大O（）记法来体现算法的时间复杂度。一般情况下，随着输入规模n 的增大，T(n)增长最慢的算法是最优算法。

分析算法的时间复杂度：

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

for(int i= 0 ; i < n; i++) O(n)

while(i<n){

i= i^2

} O(logn)