**一、数据结构的研究内容：**

数据结构它主要就是研究对象的特性以及相互之间的关系-----**逻辑结构。**

那我们如何将带结构的数据组织到计算机的内存中，也就是如何有效的组织计算机进行存储-------------**存储结构**

存储在计算机中的数据我们该如何运算，也就是如何有效的实现对象之间的“运算关系”------------**算法**

1. **算法的定义:**对特定问题求解方法和步骤的一种描述，它是指令的有效序列。其中每个指令表示一个或多个操作。

换句话说，算法就是解决问题的方法和步骤。比如说，第一步干什么，第二步干什么等等。

1. **算法的描述：**
2. **自然语言：英语、中文**

比如：求一元二次的根：

（1）输入方程的系数a、b、c

（2）判断a是否为零，若为0表示则提示不是一元二次方程的根。若不等于0，则执行第三步。

（3）计算d=b2-4ac

（4）判断d。如果d等于0，计算并输出两个相等的实根。如果d小于零，输出没有根，如果d大于0输出两个不相等的实根。

（5）结束

1. **使用流程图：**传统流程图，NS流程图
2. **使用伪代码：**类C语言
3. **程序代码：**c语言、Java语言
4. **算法与程序的关系：**

**算法是解决问题的一种方法或一个过程，它考虑的是如何将输入转换成输出，一个问题可以有多种算法。**

**程序是用某种程序设计语言，对算法的具体实现。**

1. **算法的特性**

**也就是我们所说的一个算法它应该满足什么要求。一般来说算法它具有五个重要的特性。**

1. **有穷性，一个算法总是在执行完有穷个步骤后结束，且每一步都在有穷的时间内完成。**
2. **确定性，算法中的每一条指令必须具有确切的含义，没有二义性，在任何条件下，只有唯一的一条执行路径，即对于相同的输入只能有相同的输出。**
3. **可行性，算法是可执行的，算法描述的操作可以通过已经实现的基本操作执行有限次来实现。**
4. **输入，一个算法有零个或多个输入**
5. **输出，一个算法有一个或多个输出**
6. **算法设计的要求**
7. **正确性：**那么什么样的程序才能算是正确的算法呢？一般来说，我们就要求算法满足问题要求，能正确解决问题。算法转换为程序后应该注意：i、程序中不含语言错误，我们就认为它是正确的可行吗？很显然要求有点低了。Ii、程序对于几组输入数据，能够得到满足要求的结果？Iii、程序对于精心选择的、典型的、苛刻且带有刁难性的组输入数据能够得出满足要求的结果。iiii程序对于一切合法的数据都能得出满足要求的结果。通常我们以第三个标准为准。
8. **可读性：**算法主要是为了人来阅读和交流，其次才是为了计算机执行，因此算法应该易于人理解。
9. **健壮性：**主要是指当输入非法数据时，算法也可以恰当的做出反应或进行相应处理，而不是产生莫名其妙的结果。
10. **高效性：**要求花费尽可能少的时间和尽量低的存储空间。
11. **算法和算法分析**

对于同一个问题，可能有许多种不同的算法。那么究竟该如何来评价这些算法的好坏呢？所以我们就需要对算法进行分析，从中挑选出最优的算法。那么我们又该如何来判断一个算法它是好还是坏呢?

我们说一个好的算法首先应该具备正确性、健壮性以及可读性。在这几方面都满足的情况下，主要考虑算法的效率，通过算法效率高低来评价一个算法的优劣程度。

通常情况下，我们从以下两个方面来考虑：

1. 时间效率：指的是算法所消耗的时间；
2. 空间效率：指的是算法在执行过程中所消耗的存储空间。

但是，时间效率和空间效率有时候又是矛盾的。

1. **算法时间效率的度量**

算法的时间效率可以用依据该算法编制的程序在计算机上执行所消耗的时间来度量。通常我们有两种度量的方法

1. 事后统计：将算法实现，测算其时间和空间开销。
2. 事前分析：对算法所消耗的时间进行估算，一个算法的运行时间是指一个算法在计算机上所消耗的时间大致可以等于计算机执行一次简单的操作所需要的时间与算法中进行的简单操作次数的乘积。

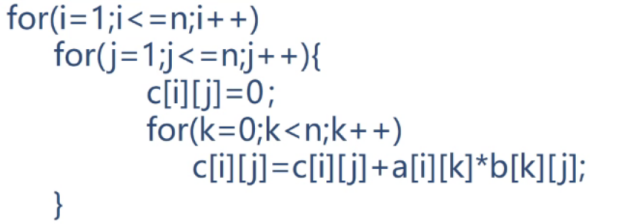
**算法运行时间=一个简单操作所需的时间\*简单操作次数**

也即算法中每条语句执行时间之和。

算法运行时间=

其中：每条语句执行的次数，我们也称为语句的频度。

那么语句执行一次需要多少时间呢？这个问题一般随机器的不同而不同，同时也与不同的编程语言而定。所以我们可以假设，执行每条语句所需要的时间均为单位时间，那么此时对算法运行时间的问题，就可以转换为讨论算法中所有语句执行的次数，即频度之和。



1. **线性表**

第一章中我们已经知道了数据结构它主要由三个部分组成，第一个就是数据的逻辑结构，也就是非数值元素之间的关系。第二个就是数据的存储结构，我们有了数据的逻辑结构以后还要进行数据的存储，我们在存储时，不仅要存储数据元素，还要存储数据元素之间的关系。最后当我们把数据元素都存储完了以后，就要进行数据的运算，这个就是我们所说的算法。

在数据的逻辑结构中，我们又可以根据数据元素之间的关系分为线性结构和非线性结构。线性结构中，元素存在着一对一的关系，而非线性关系中，数据元素存在着一对多或是多对多的关系。线性结构常用的有线性表、栈、队列、字符串、数组、广义表等。而非线性主要有树形结构和图形结构。

而数据的存储结构中我们重点关注数据的顺序存储结构和链式存储结构。

（a1,a2,...a(i-1),ai,a(i+1),...an）