**数据：**是描述客观事物的符号，是计算机中可以操作的对象，是能被计算机识别，并输入给计算机处理的符号集合。

**数据元素：**是组成数据的、有一定意义的基本单位，再计算机中通常作为整体处理。也被称为记录。

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

**数据对象：**是性质相同的数据元素的集合，是数据的子集。

****

**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。**

按照视点的不同，我们把数据结构分为**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。

逻辑结构分为以下四种：集合结构、线性结构、树形结构、图形结构。

1. 集合结构：集合结构中的数据元素除了同属于一个集合外，它们之间没有其他关系。
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**：是指数据的逻辑结构在计算机中的存储形式。

数据元素的存储结构形式有两种：**顺序存储**和**链式存储**。

**顺序存储结构**：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。

**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。

逻辑结构是面向问题的，而物理结构就是面向计算机的，其基本的目标就是将数据及其逻辑关系存储到计算机的内存中。

**数据类型**：是指一组性质相同的值的集合及定义在此集合上的一些操作的总称。

抽象数据类型(Abstract Data Type，ADT)：是指一个数学模型及定义在该模型上的一组操作。

**算法**是解决特定问题求解步骤的描述，再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。

**算法具有五个基本特性**：输入、输出、有穷性、确定性和可行性。

**算法设计的要求**：正确性、可读性、健壮性、时间效率高和存储量低。

**算法效率的度量方法**：事后统计方法(不科学、不准确)、事前分析估算方法。

**函数的渐进增长**：给定两个函数f(n)和g(n)，如果存在一个整数N，使得对于所有的n>N，f(n)总是比g(n)大，那么，我们说f(n)的增长渐近快于g(n)。

判断一个算法的效率时，函数中的常数和其他次要项常常可以忽略，而更应该关注主项(最高阶项)的阶数。

某个算法，随着n的增大，它会越来越优于另一算法，或者越来越差于另一算法。这其实就是事前估算方法的理论依据，通过算法时间复杂度来估算算法时间效率。

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，也就是算法的时间量度，记作：T(n)=O(f(n))。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，简称为时间复杂度。其中f(n)是问题规模n的某个函数。

这样用大写O()来体现算法时间复杂度的记法，我们称之为大O记法。





**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。