数据：是对客观事物的符号表示。
数据元素：是数据的基本单位，也称节点（node）或记录（record）。
数据对象：是性质相同的数据元素的集合，是数据的一个子集。
数据项：有独立含义的数据最小单位，也称域(field)。
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。
根据数据元素间关系的基本特性，有四种基本数据结构

集合：结构中的数据元素之间除了“同属于一个集合”的关系外，别无其他关系。
线性结构：结构中的数据元素之间存在一个对一个的关系。
树形结构：结构中的数据元素之间存在一个对多个的关系。
图状结构或网状结结构：结构中的数据元素之间存在多个对多个的关系。

逻辑结构：抽象反映数据元素之间的逻辑关系。（算法设计）
物理结构（存储结构）：数据结构在计算机中的表示。（算法实现）
存储结构分为：

顺序存储结构：借助元素在存储器中的相对位置来表示数据元素间的逻辑关系。
链式存储结构：借助指示元素存储地址的指针表示数据元素间的逻辑关系。

算法：对特定问题求解步骤的一种描述。
算法的五个重要特性：有穷性，确定性，可行性，输入和输出。
算法设计的原则或要求：正确性，可读性，健壮性，效率与低存储量需求。
衡量算法效率的方法：事后统计法和事前分析估算法 。
算法执行时间的增长率和 f(n) 的增长率相同，则可记作：T (n) = O(f(n))，称 T (n) 为算法的(渐近)时间复杂度
算法运行时间的衡量准则：以基本操作在算法中重复执行的次数。
栈：限定仅在表尾进行插入或删除操作线性表。入栈：插入元素的操作；出栈：删除栈顶元素的操作。
队列：只能在队首进行删除、队尾进行插入的线性表。允许插入的一端叫队尾，删除的一端叫队头。
串：由零个或多个字符组成的有限序列；空串：零个字符的串；长度：串中字符的数目；
空串：零个字符的串；子串:；串中任意个连续的字符组成的子序列；位置：字符在序列中的序号；
相等：串的值相等；空格串：由一个或多个空格组成的串，空格串的长度为串中空格字符的个数。
存储位置：LOC(i ,j)=LOC(0,0)+(b_2*i+j)L
结点:包含一个数据元素及若干指向其子树的分支；结点的度: 结点拥有的子树；
树的度:树中所有结点的度的最大值；叶子结点: 度为零的结点；分支结点: 度大于零的结点
树的深度：树中叶子结点所在的最大层次　　森林：m 棵互不相交的树的集合。

二叉树的性质：

性质 1：在二叉树的第 i 层上至多有 2^{i-1} 个结点。(i≥1)
性质 2：深度为 k 的二叉树上至多含 2^k-1 个结点。(k≥1)
性质 3：对任何一棵二叉树，若它含有 n_0 个叶子结点、n_2 个度为 2 的结点，
则必存在关系式：$n_0 = n_2+1$。
性质 4：具有 n 个结点的完全二叉树的深度为 $\lfloor \log_2 n \rfloor +1$。
满二叉树：指的是深度为 *k* 且含有 2^k-1 个结点的二叉树。
完全二叉树：树中所含的 *n* 个结点和满二叉树中编号为 *1* 至 *n* 的结点一一对应。
路径长度：路径上分支的数目。树的路径长度：树根到每个结点的路径长度之和。
树的带权路径长度：树中所有叶子结点的带权路径长度之和，记作：$WPL(T) = \Sigma w_k l_k$
带权路径 长度最小的二叉树，称为最优树二叉树或赫夫曼树。
关键路径：路径长度最长的路径。

顶点：数据元素 vi 称为顶点

边、弧：P (vi,vj)表示顶点 vi 和顶点 vj 之间的直接连线，在无向图中称为边，在有向图中称为弧。任意两个顶点构成的偶对(vi,vj)∈E 是无序的，该连线称为边。是有序的，该连线称为弧。弧头、弧尾：带箭头的一端称为弧头，不带箭头的一端称为弧尾。

顶点的度(TD)=出度(OD)+入度(ID)

图的遍历算法是求解图的连通性问题、拓扑排序和求关键路径等算法的基础。

通常有两条遍历图的路径：深度优先搜索和广度优先搜索。

排序的分类：

按待排序记录所在位置

内部排序：待排序记录存放在内存

外部排序：排序过程中需对外存进行访问的排序

按排序依据原则

插入排序：直接插入排序、折半插入排序、希尔排序

交换排序：冒泡排序、快速排序

选择排序：简单选择排序、堆排序

归并排序：2-路归并排序

基数排序