**存储结构：**

描述了数据在计算机内部的实际存储方式。它主要关注数据在计算机内存中的组织形式，以及如何在内存中存储和访问数据。存储结构通常包括以下几种：

**顺序存储结构**：数据元素在内存中按照一定的顺序依次存储，相邻的元素在内存中也是相邻的，如顺序表、数组，可以随机存取。

**链式存储结构**：数据元素通过指针相互连接，每个元素在内存中可以不连续存储，如链表。

链接存储的存储结构所占存储空间分两部分，一部分存放结点值，另一部分存放表示结点间关系的指针,单链表数据密度<1。

**索引存储结构**：除了数据元素本身外，还维护一个索引表，索引表中的每个元素指向相应数据元素的位置。

**散列存储结构**：通过散列函数将数据元素直接映射到内存地址，实现快速的查找。

存储结构与具体的数据结构有关，不同的数据结构可以选择不同的存储结构来实现。例如，数组通常使用顺序存储结构，链表则使用链式存储结构。

**逻辑结构**：

描述了数据元素之间的逻辑关系，以及对这些关系的操作。它不涉及具体的存储细节，而是关注数据元素之间的关联和组织方式。通常要求同一逻辑结构中的所有数据元素具有相同的特性，这意味着不仅数据元素所包含的数据项的个数要相同，而且对应数据项的类型要一致。逻辑结构通常包括以下几种：

**线性结构**：数据元素之间存在一对一的关系，可以通过一维的方式排列，例如线性表、栈、队列等。

**树形结构**：数据元素之间存在一对多的关系，可以用树形方式组织，例如二叉树、堆、哈夫曼树等。

**图形结构**：数据元素之间存在多对多的关系，例如图、网等。

有序表也属于逻辑结构

逻辑结构与数据的抽象关系更为密切，它们描述了数据元素之间的逻辑联系和操作规则，而不关注数据元素的存储方式。例如，队列的逻辑结构就是先进先出（FIFO）的关系，而队列的存储方式可以是数组或链表。

**排序算法：**

1. 插入排序（稳定排序）：每次循环多考虑一个数，将该“插入的”数排序到已经排好的数列中的正确位置。
2. 冒泡排序（稳定排序，原地排序）：第一次循环把最大的数移到最右侧，第二次循环把最小的数移到次右侧，以此类推；每次循环通过不断比较邻近的两个数的大小，若前者大则交换顺序来实现。
3. 选择排序（非稳定排序，原地排序）：第一次循环寻找最大的数的索引，然后与最右侧的数交换顺序，第二次循环寻找次大的数的索引，然后与次右侧的数交换顺序，以此类推。
4. 快速排序（非稳定排序）：选定最右侧的数作为pivot，通过partition函数和双指针方法确认pivot的大小位置，通过分治+递归确认每个元素的大小位置。
5. 归并（merge）排序（稳定排序）：运用递归+分治的方法，将待排序的数不断二分为单个元素，从左至右依次排序再归并起来排序。
6. 希尔（shell）排序（非稳定排序）：效率更高的插入排序，先将数列间隔开，然后对每个间隔中的相同次序的元素进行排序，然后不断缩小间隔，最终间隔缩小为1，排序完成。

