# 数据结构及算法基础

数据结构表示数据在计算机中的存储和组织形式，描述元素之间的位置与逻辑关系



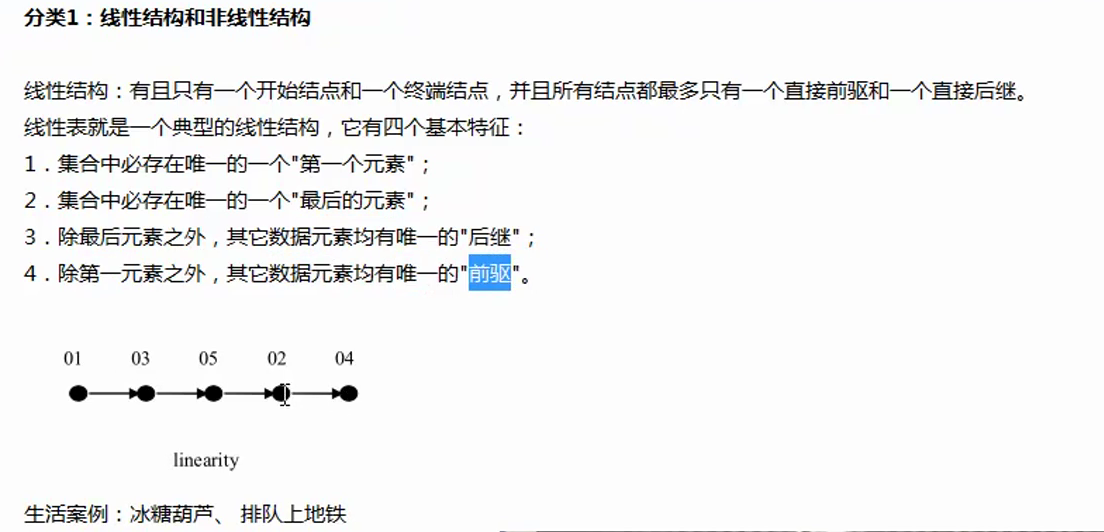
数据结构=逻辑结构+物理结构（存储结构）+各种运算和操作

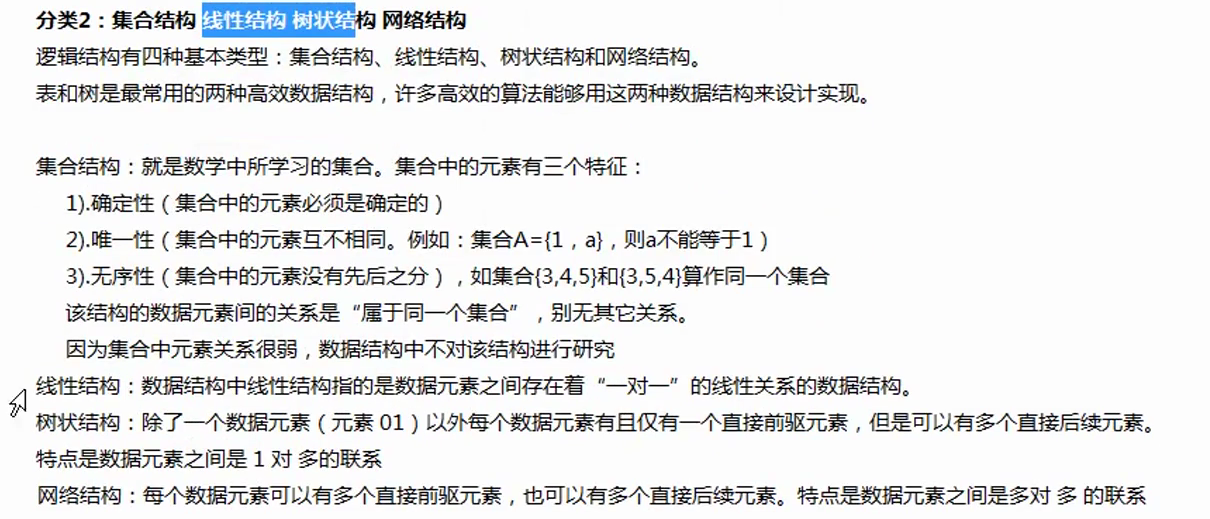
时间复杂度—运行效率

空间复杂度—存储效率

## 逻辑结构

描述数据之间的逻辑关系





1. 集合结构 元素之间同属一个总体不存在其他联系
2. 线性结构 元素之间存在一对一关系
3. 树形结构 元素之间存在一对多关系
4. 网状结构 元素之间存在多对多关系

## 存储结构

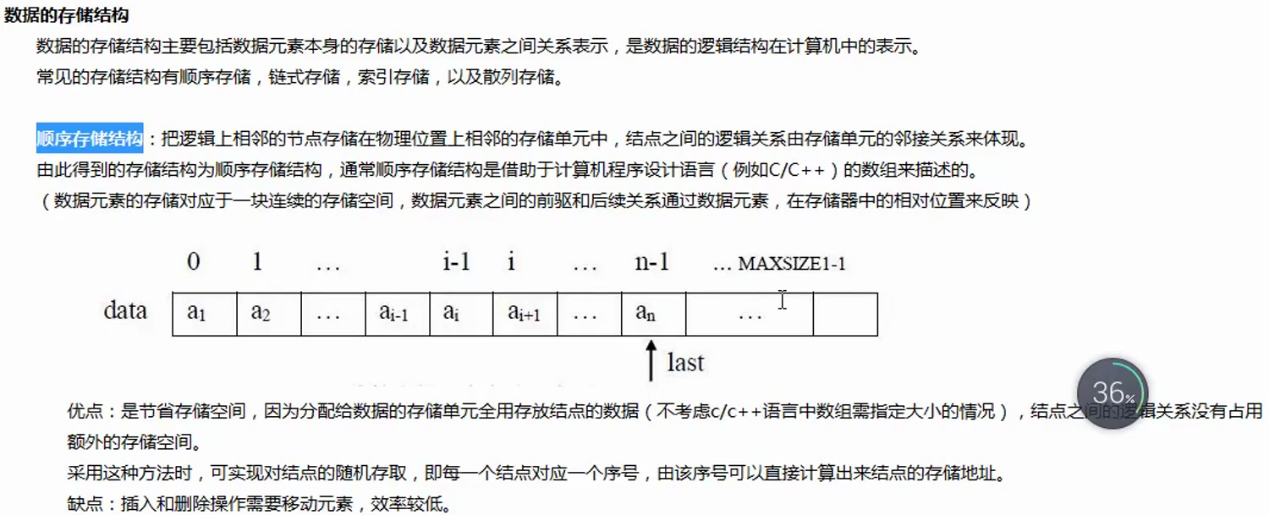
描述元素存储位置关系

1. 顺序结构

逻辑上相邻的两个元素在存储位置上也相邻，通常当数据采用顺序结构存储是需要实现申请一块连续的内存空间（如数组）

优点在于支持随机访问，已知元素下标的情况下访问速度极快

缺点是需要静态分配连续的空间，内存空间的利用率较低，插入元素和删除元素可能需要移动其他元素，效率较低

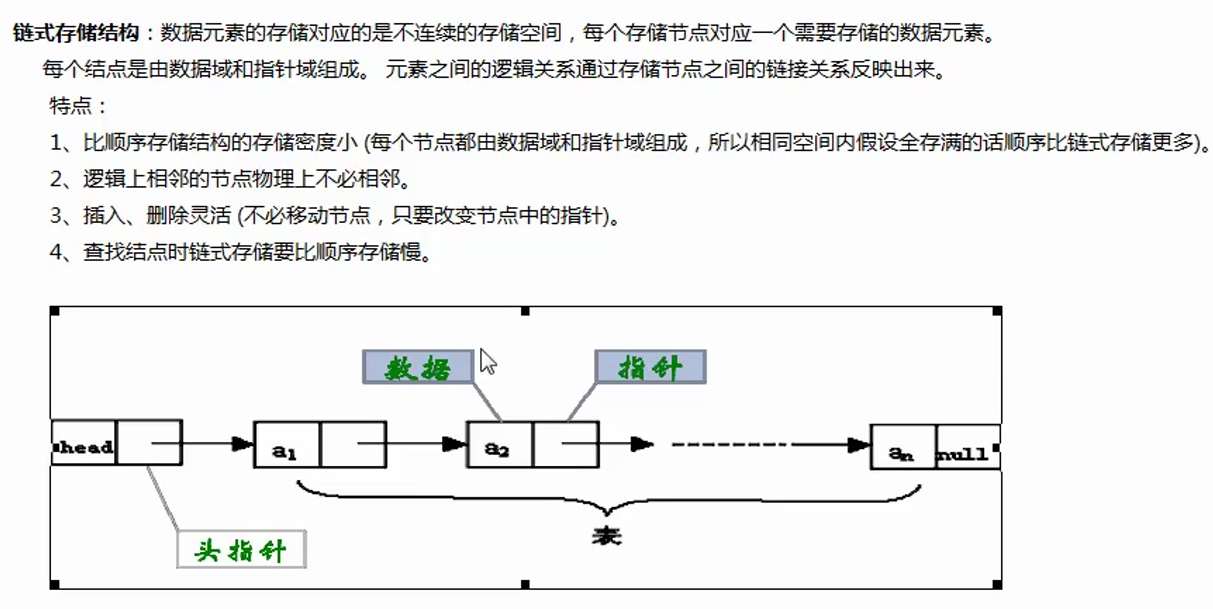


1. 链式结构

逻辑上相邻的两个元素通过指针进行关联，为每一个元素构造一个节点，节点存储元素内容及指针信息指向下一个节点

优点：空间利用率高不需要连续存储空间，插入和删除元素只需要添加和删除指针，增删效率高

缺点：需要额外空间存储指针信息，不支持利用下标对元素随机访问



1. 索引结构

除了存储元素信息之外，单独维护一个索引表存储节点的地址，索引表由若干索引项组成

优点：利用索引号查找元素存储地址，检索速度快

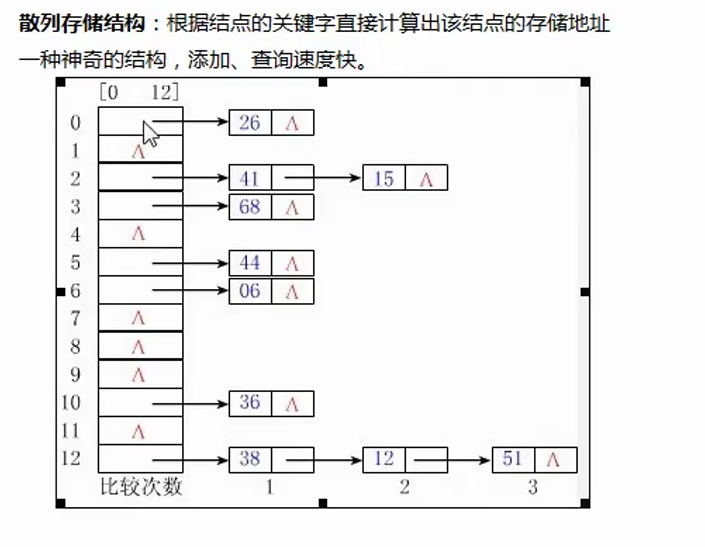
缺点：需要单独空间存储索引表

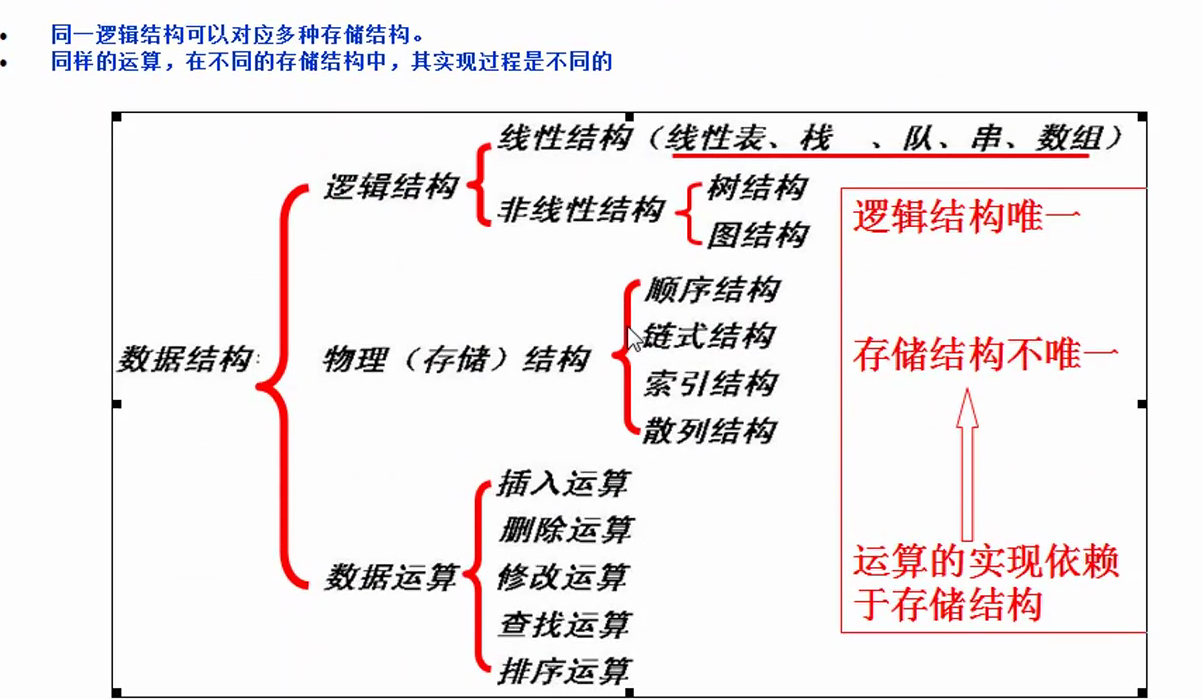
1. 散列结构

通过元素的键通过计算得出存储地址，旨在消除顺序存储结构的插入和删除的低效率与链式存储的检索低效率

优点：散列是数组存储方式的一种发展，采用存储数组中内容的部分元素作为映射函数的输入，映射函数的输出就是存储数据的位置，相比数组，散列的数据访问速度要高于数组

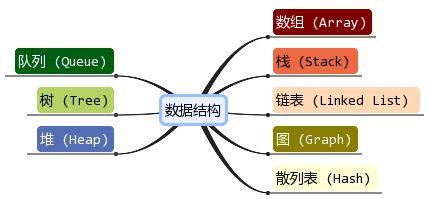
缺点：不支持排序，关键字不能重复，相比线性表存储需要更多空间。

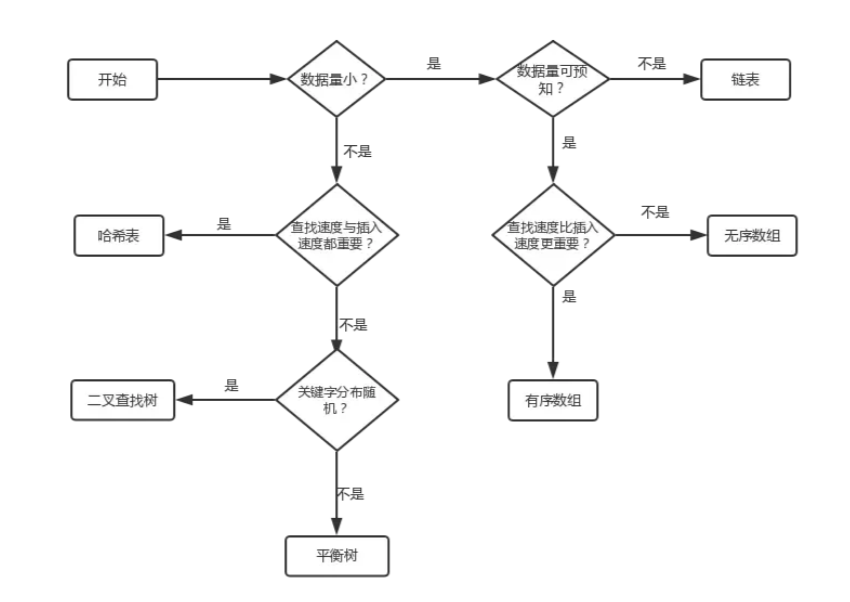




## 数据结构的选择

常见的数据结构

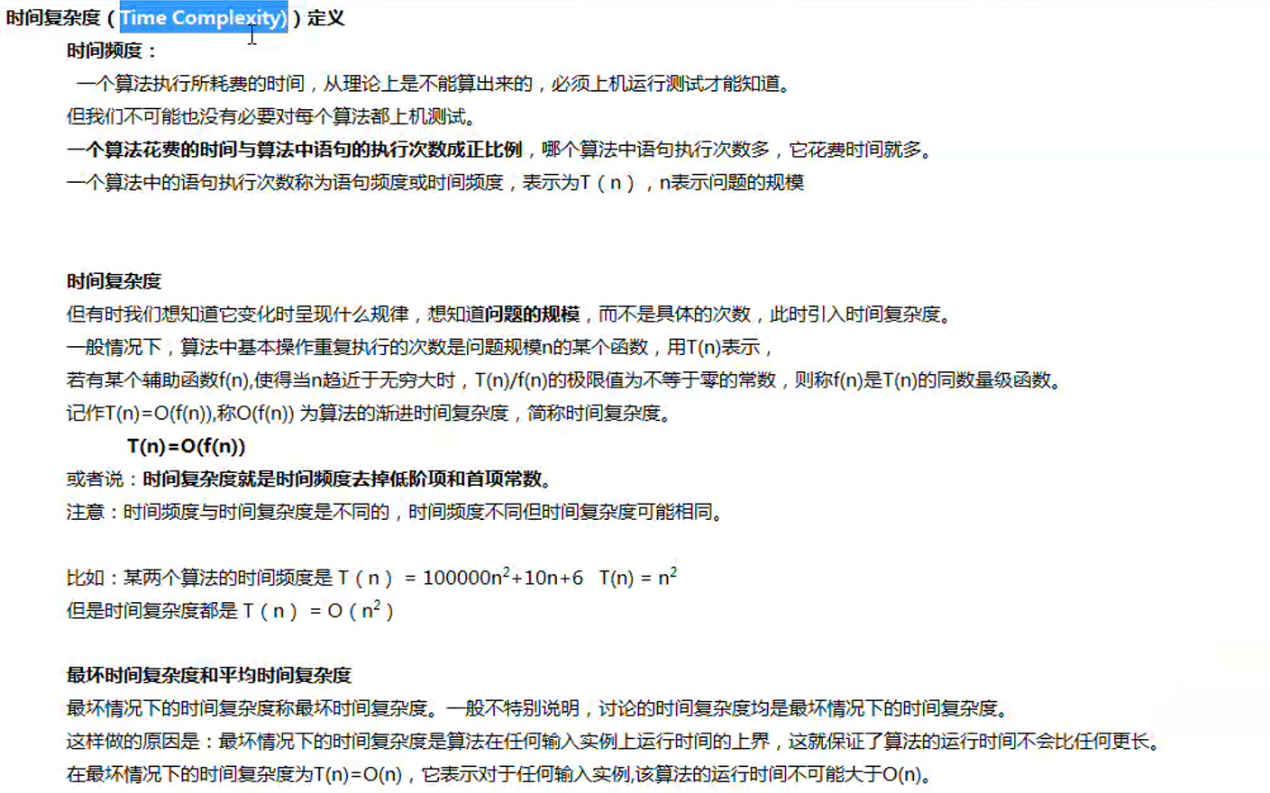


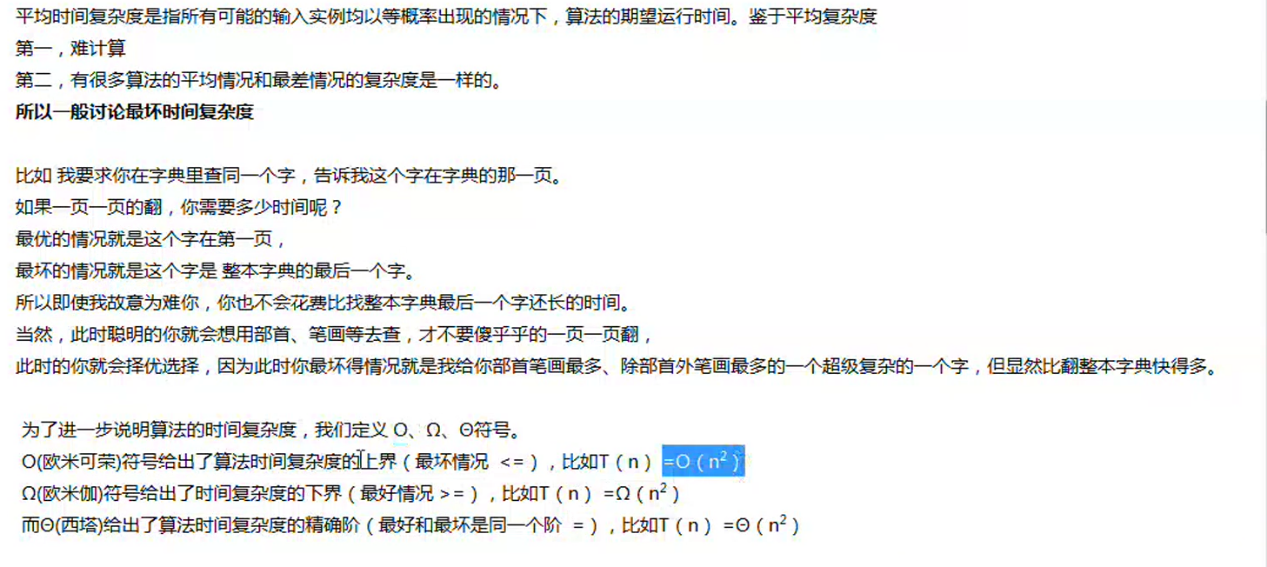


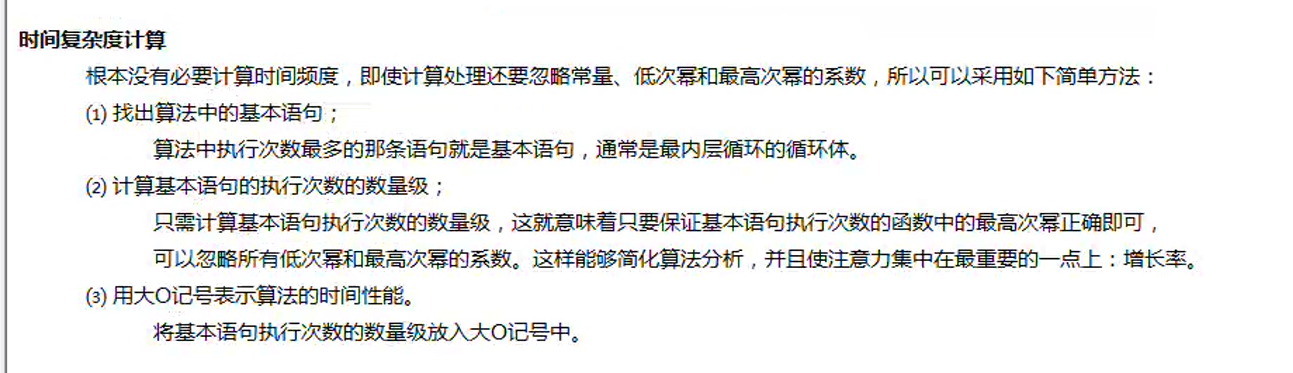
## 算法

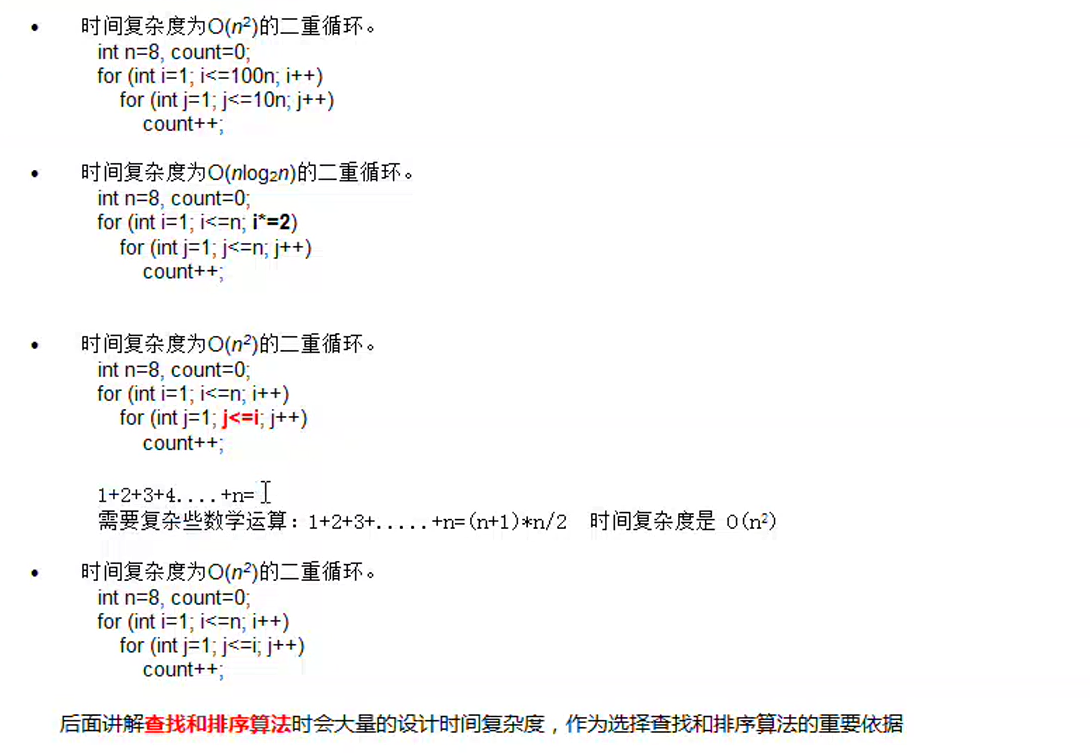


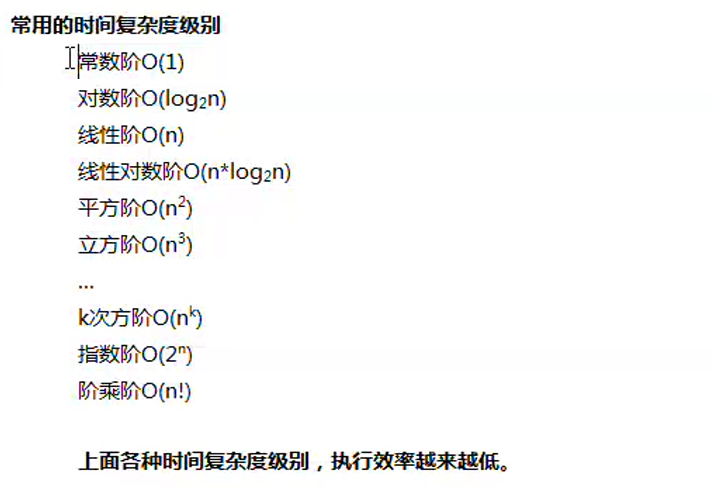
## 时间复杂度

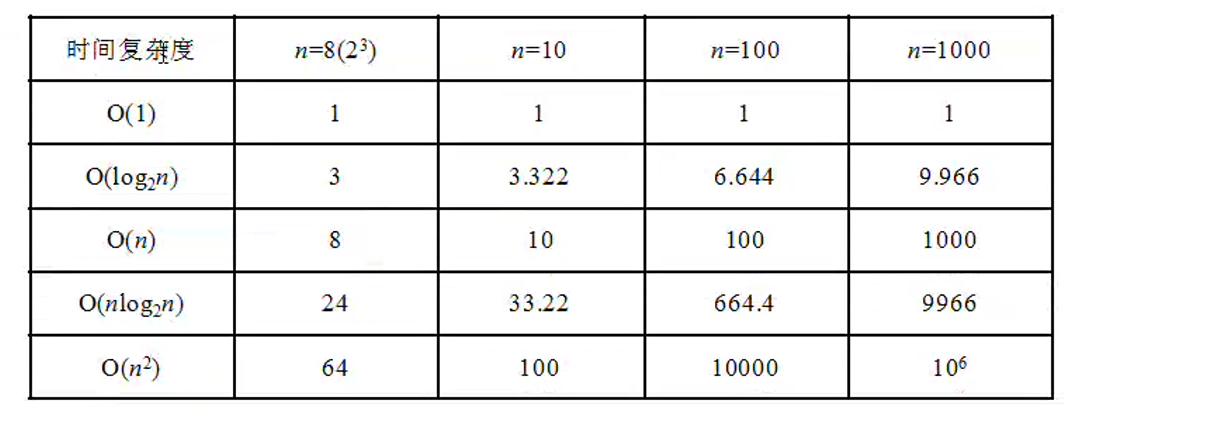


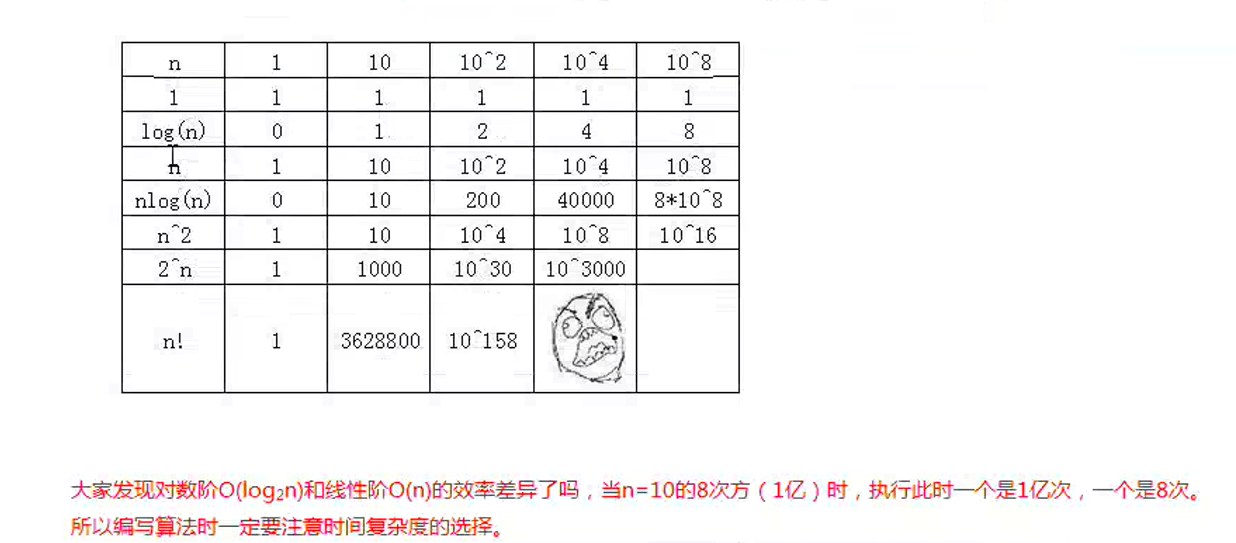




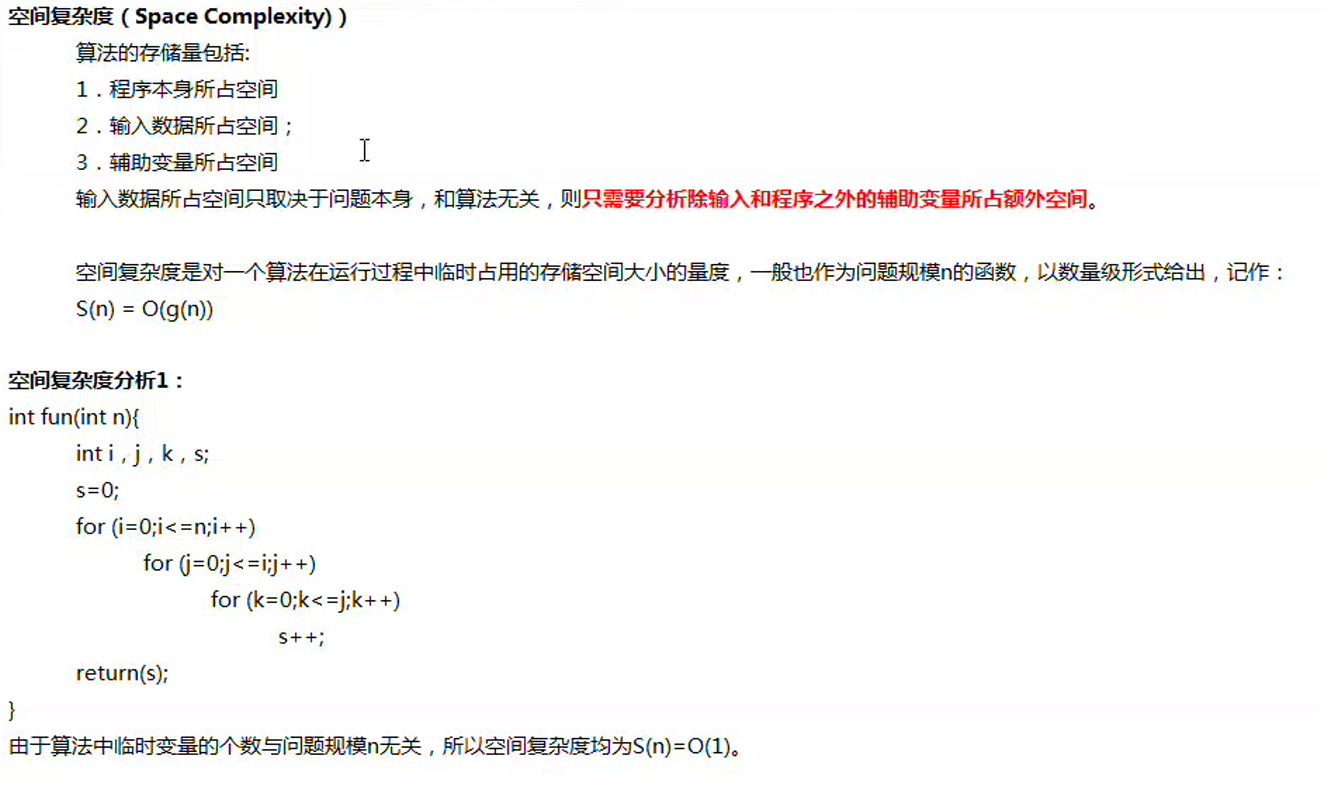


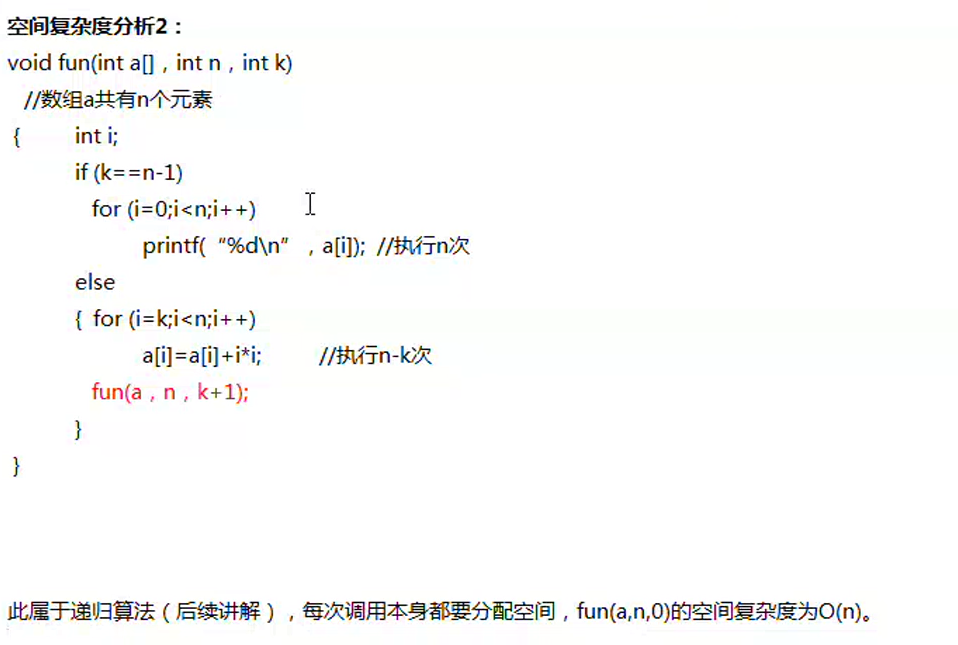






## 空间复杂度





进行到04