

DS概述

copyright@分时天月

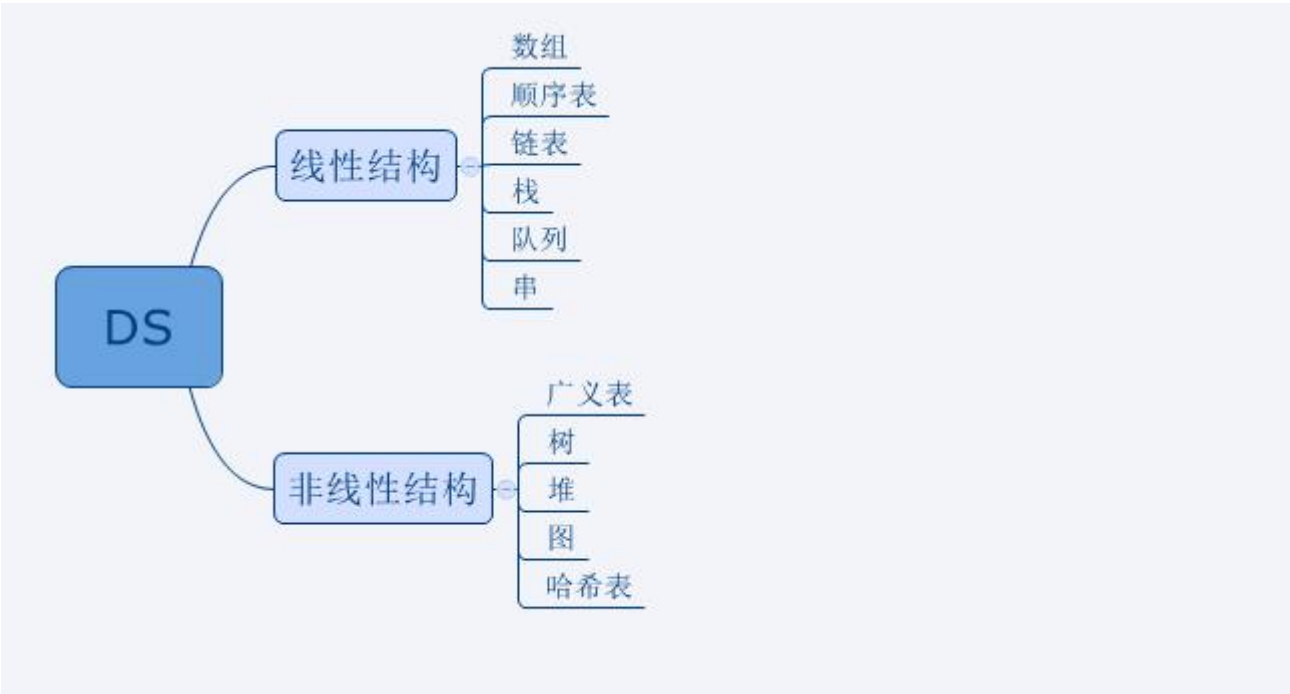
作为一个程序员，如果你是做数据库编程的，大可以不懂汇编语言，如果你是写驱动程序的，大可以不必通晓人工智能，写编译器的可以不用懂什么计算机图形学，操作系统内核高手不用精通网站架设，然而，如果你不懂数据结构和算法的基础知识，不具备数据结构和算法的基本技能，那就完全丧失了一个程序员的资格！

--CSDN副总裁孟岩

下面这段话是我个人的理解，仅供参考。

大部分初学者经常学了一阵子DS，却感到迷茫，因为不知道学这玩意儿有什么用，并且学起来还挺复杂的。甚至有一些有一定编程经验的学生会对之嗤之以鼻。因为这部分人接触到了一定程度的开发技能，有的是前端开发，学了三剑客（HTML | CSS | JavaScript）就可以开发一个差不多的网页；有的是做一些小软件/游戏，学了STL同样可以大大减少开发难度。但是这些技术都是表象的，貌似用起来没什么问题，但是一旦出现一些深层次的bug，他们就会无能为力，因为根本不知道底层是怎么搞的，自然无法从根本入手解决，只能从网上找一些类似的例子，但是又哪可能会一模一样。要想成为一个大神，肯定要对这些这些了如执掌，运用起来才更能如臂使指。

DS分类和用处概述



上图就是数据结构的基本分类情况

我在这里举几个计算机应用中用到数据结构的例子：

1. 计算机通信中(进程通信，网络通信...)经常会用到 **队列**。
2. 计算机进程调度算法的结构支持就是 **优先级队列**
3. 函数的调用实现是 **栈** 结构支持的
4. 计算机的目录结构是 **B树**（树的一种）
5. OSI七层网络模型中的数据链路层使用的寻路算法是 **图** 的遍历算法
6. 通常用于服务器编程处理高并发事件的Epoll机制用到了 **队列** 和 **红黑树**（树的一种）
7. google 搜索引擎的底层使用的布隆过滤器是 **哈希表** 和 **位图** 实现的。

.....

而各种高级结构的实现几乎都离不开 **链表**，由此见之，DS在计算机中的使用范围之广，其重要性之切！！

线性结构概述

为了能够让读者充分认识到链式存储和顺序存储的区别和优缺点，我这里会重点将顺序表和链表做一个比较。

什么是线性结构呢？

顾名思义，是说数据在逻辑结构上具有线性关系。将具有线性关系的数据存储到计算机中所使用的存储结构称为线性表。

对于线性表中的数据来说，位于当前数据之前的数据统称为“前趋元素”，前边紧挨着的数据称为“直接前趋”；同样，后边的数据统称为“后继元素”，后边紧挨着的数据称为“直接后继”。



线性表的特点：

- 存储的数据本身的类型一定保持相同，是int型就都是int型，是结构体就都是一种结构体。
- 数据一旦用线性表存储，各个数据元素之间的逻辑相对位置就固定了。

线性表的分类：

逻辑结构上相邻的数据在实际的物理存储中有两种形式：分散存储和集中存储。

逻辑结构就是程序员，也就是我们主观上理解的存储结构

物理结构就是数据实际在计算机中存储的方式

- 顺序结构，eg: 顺序表
- 链式结构，eg: 链表

