

链表的建立、插入和删除 (一) PDF转换可能丢失图片或格式, 建议阅读原文

[https://www.100test.com/kao\\_ti2020/136/2021\\_2022\\_\\_E9\\_93\\_BE\\_E8\\_A1\\_A8\\_E7\\_9A\\_84\\_E5\\_c97\\_136227.htm](https://www.100test.com/kao_ti2020/136/2021_2022__E9_93_BE_E8_A1_A8_E7_9A_84_E5_c97_136227.htm) 数组作为存放同类数据的集合, 给我们在程序设计时带来很多的方便, 增加了灵活性。但数组也同样存在一些弊病。如数组的大小在定义时要事先规定, 不能在程序中进行调整, 这样一来, 在程序设计中针对不同问题有时需要30个大小的数组, 有时需要50个数组的大小, 难于统一。我们只能够根据可能的最大需求来定义数组, 常常会造成一定存储空间的浪费。我们希望构造动态的数组, 随时可以调整数组的大小, 以满足不同问题的需要。链表就是我们需要的动态数组。它是在程序的执行过程中根据需要有数据存储就向系统要求申请存储空间, 决不构成对存储区的浪费。链表是一种复杂的数据结构, 其数据之间的相互关系使链表分成三种: 单链表、循环链表、双向链表, 下面将逐一介绍。7.4.1 单链表图7-3是单链表的结构。单链表有一个头节点head, 指向链表在内存的首地址。链表中的每一个节点的数据类型为结构体类型, 节点有两个成员: 整型成员(实际需要保存的数据)和指向下一个结构体类型节点的指针即下一个节点的地址(事实上, 此单链表是用于存放整型数据的动态数组)。链表按此结构对各节点的访问需从链表的头找起, 后续节点的地址由当前节点给出。无论在表中访问那一个节点, 都需要从链表的头开始, 顺序向后查找。链表的尾节点由于无后续节点, 其指针域为空, 写作为NULL。图7-3还给出这样一层含义, 链表中的各节点在内存的存储地址不是连续的, 其各节点的地址是在需要

时向系统申请分配的，系统根据内存的当前情况，既可以连续分配地址，也可以跳跃式分配地址。看一下链表节点的数据结构定义：`struct node{int num.struct node *p.}`。在链表节点的定义中，除一个整型的成员外，成员p是指向与节点类型完全相同的指针。在链表节点的数据结构中，非常特殊的一点就是结构体内的指针域的数据类型使用了未定义成功的数据类型。这是在C中唯一规定可以先使用后定义的数据结构

。#8226. 单链表的输出过程有以下几步1) 找到表头。2) 若是非空表，输出节点的值成员，是空表则退出。3) 跟踪链表的生长，即找到下一个节点的地址。4) 转到2)。[例7-5] 创建一个存放正整数（输入-999做结束标志）的单链表，并打印输出。`#include /包*含malloc()的头文件*/#include struct node /*链表节点的结构*/{int num.struct node *next.} .m a i n ( ){struct node *creat(). /*函数声明*/void print().struct node *head. /*定义头指针*/head=NULL. /*建一个空表*/head=creat(head). /*创建单链表*/print(head). /*打印单链表*/} 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com`