# 数据结构知识点总结

大梦谁先觉i

已于 2022-06-13 16:01:10 修改 8454

收藏 121

分类专栏： 数据结构与算法 文章标签： 数据结构 c++

版权

数据结构与算法 专栏收录该内容

3 篇文章 0 订阅

订阅专栏

## 一，数据结构的一般概念

1）数据：所有能被输入到计算机并被处理的符号的集合

2）数据元素：数据的基本单位。

3）数据项：构成数据的不可分割的最小单位，一个数据元素有若干个数据项组成。

4）数据对象：相同性质的数据元素的集合，是数据的一个子集。

5）数据类型：一个值的集合和定义在这个集合上的一组操作的总成。

6）原子类型：值不可在分割的数据类型。

7）结构类型：值不可在分解成若干的数据类型。

8）抽象数据类型：抽象数据组织和与之相关的操作。

9）抽象数据类型（ADT):一个数学模型以及定义在该模型上的一组操作。其定义只与逻辑特性有关，通常采用（数据对象，数据关系，基本操作集）这样的三元来表示抽象数据类型。

10）数据结构：相互之间存在的一种或多种特定关系的数据元素的集合，包括：逻辑结构，存储结构和数据的运算。

11）数据的三要素：逻辑结构，物理结构，数据元素。

12）数据的逻辑结构：数据元素之间的逻辑关系。

13）数据的逻辑结构分类图：

集合：结构中的数据元素之间除了“同属一个集合”的关系之外，没有任何关系。

线性结构：结构中的数据元素之间只存在一对一的关系。

树形结构：结构中的数据元素之间存在一对多的关系。

图状结构或网状结构：结构中的数据元素之间存在多对多的关系。

14）数据的存储结构：数据结构在计算机中的表示，也称物理结构。

15）顺序存储：把逻辑上相邻的元素存储在物理位置上相邻的存储单元里，通过存储单元的邻接关系来表示元素之间的逻辑关系。

优点：实现随机存储，每个元素占用空间小。

缺点：只能使用相邻的一整块存储单元，会产生较多的外部碎片。

16）链式存储：不要求逻辑上相邻的元素在物理位置上也相邻，通过指针表示元素之间的逻辑关系。

优点：不会出现碎片现象，充分利用所有的存储单元。

缺点：每个元素要占用存储指针，需要多占用部分存储空间，而且只能顺序存取。

17）索引存储：存储信息的同时，建立附加的索引表，索引表中的每一项成为索引项，索引项的一般形式（关键字，地址）

优点：检索速度快。

缺点：增加索引表，占用较多存储空间，增删数据时也要修改索引表，花费较多的时间。

18）散列存储：根据元素的关键字直接计算出该元素的存储位置，也称Hash存储

优点：检索，增删节点操作都很快

缺点：散列函数不好可能会出现元素存储单元的冲突，解决冲突会增加时间 ，空间的开销。

19）算法的基本概念：

算法对特定问题求解步骤的一种描述，它是指令的有限序列，期中每一条指令都表示一个或多个操作。

20）算法的5个重要性：有穷性，确定性，可行性，输入，输出。

21）算法设计的要求：正确性，可读性，健壮性，效率与低存储需求。

22）算法效率的度量：通常用时间复杂度和空间复杂度来描述。

23）时间复杂度：算法中所有语句的频度（指该语句在算法中被重复执行的次数）之和记作T(n),时间复杂度主要分析T(n)的数量级。

算法中的基本运算(最深层循环内的语句）的频度与T(n)同数量级，所以一般采用算法中最基本的频度f(n)来分析算法时间复杂的度。即T(n)=O(f(n))

24）空间复杂度：算法耗费的存储空间，记作S(n)=O(g(n))

算法原地工作指算法所需要辅助空间是常量，即O(1）

## 二，线性表的定义和基本操作

1）线性表的定义： 具有相同数据类型的n(N>=0)个数据元素的有限序列。

2）线性表的特点：

1.除第一个元素外，每个元素有且仅有一个直接前驱，除最后一个元素外，每个元素有且仅有一个直接后继

2.表中元素个数有限

3.表中元素具有逻辑上的顺序关系

4.表中每个元素都是数据元素，每个元素都是单个元素

5.表中元素的数据类型都相同，即每个元素占有相同大小的存储空间

6.表中元素具有抽象性，即只关注与逻辑结构，不关注于元素表示的内容

7.线性表示一种逻辑结构 ，表示元素之间一对一的相邻关系；顺序表和链表是存储结构，表示物理结构

3）线性表的基本操作：

InitList(&L):初始化表

Length(L):求表长

LocateElem(L,e)：按值查找操作

GetElem(L,i):按位查找操作

ListInsert(&L,e):插入操作

ListDelete(&L,i,&e):删除操作

Print List(L)：输出操作

Empty(L):判空操作

DestroyList(&L):销毁操作

4）线性表的顺序表示

顺序表的定义：用一组连续的存储单元，一次存储线性表中的数据元素，使得逻辑上相邻的数据元素在物理位置上也相邻。

顺序表的特点：随机访问，并且存储密度高，但是增删操作需要移动大量元素

结构体描述：

基本操作相关代码：

1.插入操作

2.删除操作

3.按值查找

5）线性表的链式表示

单链表的定义：通过一组任意的存储单元来存储线性表中的数据元素，每个链表节点除了放自身的信息外，还要存放一个指向后继的指针，其中data为数据域。next为指针域。

节点类型定义

1.单链表中的元素是离散的分布在存储空间中的。所以是非随机存储结构，想找到某个元素必须从头遍历。

2.通常用头指针标识一个单链表，此外，在单链表的第一个结点之前附加一个节点，称之为头节点。头结点中可以不加任何信息，也可以记录表长等。

6）引入头结点的优点：

开始节点放在头结点的指针域中，所以链表的第一个节点位置上的操作与其他位置上的操作一致，不需要特殊处理。

若链表为空，头指针是指向头结点的非空指针（头结点的指针域为空），所以空表与非空表的处理统一。

7）单链表解决了顺序表需要大量连续存储空间的缺点，但是单链表附加了指针域，也存在浪费存储空间的缺点。

基本操作相关代码

1.建立单链表

2.按序号查找节点值

3.按值查找

4.插入节点主要代码片段

5.删除节点操作

8）双向链表：

双向链表是单链表中有一个指向后继节点的指针next的基础上，增加了一个指向前驱节点的指针prior

9）循环链表

循环单链表：在单链表的基础上，表中最后一个节点的指针不是NULL,而是改为指向头结点，整个链表形成一个环。

因为没有指针域为NULL的节点，所以，循环单链表的判空条件不是头结点的指针是否为空，而是它是否等于头指针。

插入，删除操作算法与单链表一致，只是在尾部操作不同，

10）循环双向链表：在双向链表的基础上，表中最后一个节点的指针不是NULL，而是改为指向头结点，整个链表形成一个环。

判空条件为头结点的prior域和next域都等于头结点。

11）静态链表

静态链表是借助数组来描述线性表的链式存储结构，节点也有数据域data和指针域next，不过这里的指针域指的数组下标（游标）

结点类型定义

12）顺序表和单链表的比较

1.存取方式

顺序表可以顺序存取，也可以随机存取；链表只能顺序存取

2.逻辑结构和物理结构

顺序表，逻辑上相邻的元素，物理位置上也相邻；链表，逻辑上相邻，物理位置上不一定相邻

3.查找,插入和删除操作时间复杂度

按值查找：顺序表无序时，两者时间复杂度都为O(n)；当顺序表有序时，可以采用折半查找，时间复杂度O(log2 n)

按位查找：顺序表随机访问，时间复杂度为链表平均时间复杂度为O(n)

## 三，其他线性结构

### 栈

1）栈的基本概念

栈的定义：只允许一端进行插入和删除操作的线性表

栈顶：栈允许进行插入和删除的那一端

栈底：固定的，不允许进行插入和删除的一端。

栈是一个先进后出的线性表

2）栈的基本操作

InitStack(&S)：栈的初始化

StackEmpty(S)：判断栈是否为空

Push(&S,x)：进栈

Pop(&S,&x)：出栈

GetTop(S,&x)：读栈顶元素

ClearStack(&S)：销毁栈

3）顺序栈：

栈空条件：S.top=-1；栈满条件：S.top=MaxSize-1；栈长：S.top+1

4）顺序栈的基本操作：

共享栈：利用栈底位置不变的特性，让两个栈共享一个一维数组空间，将两个栈的栈底分别设置在共享空间的两端，两个栈顶向共享空间的中间延伸

top0=-1是0号栈为空，top1=MaxSize时1号栈为空

仅当两个栈顶指针相邻时，判断为栈满

链栈：采用链式存储的栈，通常采用单链表实现，并规定所有操作在单链表的表头进行，且链栈没有头结点，Lhead指向栈顶元素

### 队列

1）队列的基本概念

队列的定义：只允许在表的一段进行插入，表的另一端进行删除

队头：允许删除的一端

队尾：允许插入的一端

空队列：不含任何元素的空表

队列是一个先进先出的线性表

2）队列的基本操作

InitQueue(&Q)：初始化队列

QueueEmpty(Q)：判队列空

EnQueue(&Q)：入队

DeQueue(&Q,&x)：出队

GetHead(Q,&x)：读队头元素

3）队列的顺序存储结构

队列的顺序存储：分配一块连续的存储单元存放队列中的元素，并设置两个指针front和rear分别指示队头元素和队尾元素的位置。

结构体描述

队头指针指向对头元素，队尾指针指向队尾元素的下一个位置

队列判空条件：Q.front=Q.rear=0

队列会出现假溢出的情况

4）循环队列

将顺序队列想象成一个环状空间，也就是逻辑上将存储队列元素的表看成一个环，即循环队列

初始时：Q.front=Q.rear=0

对首指针进1：Q.front=(Q.front+1)%MaxSize

对尾指针进1：Q.rear=(Q.rear+1)%MaxSize

队列长度：(Q.rear+MaxSize-Q.front)%MaxSize

队满判断：一般有三种

牺牲一个单元来区分队空队满，即队头指针在队尾指针的下一位置作为队满的标志

队满条件：(Q.rear+1)%MaxSize==Q.front

队空条件：Q.front==Q.front

队列中元素的个数：(Q.rear-Q.front+MaxSize)%MaxSize

类型中增设表示元素个数的数据成员size，则队空条件为Q.size=0，队满条件为Q.size=MaxSize，两种情况中都有Q.front=Q.rear

类型中增设tag数据成员，区分队空还是队满。则tag等于0的情况下，因删除导致Q.front==Q.rear则为队空，tag=1的情况下，因插入导致Q.front==Q.rear则为队满

循环队列的基本操作相关代码

5）队列的链式存储结构

队列的链式表示即链队列

结构体描述

Q.front==NULL和Q.rear==NULL时，队列为空

链队列通常设计成带头结点的单链表，这样插入删除操作统一

链队列基本操作相关代码

6）双端队列

双端队列是指允许两端都可以进行入队和出队操作的

输出受限的双端队列：允许一端进行插入和删除，但在另一端只允许插入的双端队列

输入受限的双端队列：允许一端进行插入和删除，但在另一端只允许删除的双端队列

7）栈和队列的应用

中缀转后缀表达式

若为‘(’，入栈

若为‘)’，则依次把栈中的运算符加入后缀表达式，直到出现‘(’，并从栈中删除‘(’

若为‘+’，‘-’，‘\*’，‘/’

栈空，入栈

栈顶元素为‘(’，入栈

高于栈顶元素优先级，入栈

否则，依次弹出栈顶运算符，直到一个优先级比他低的运算符或‘)’为止

\*遍历完成，若栈非空，依次弹出所有元素

### 矩阵

1）压缩矩阵：指多个值相同的元素只分配一个存储空间，对零元素不分配存储空间。其目的是为了节省存储空间

2）特殊矩阵：指具有许多相同矩阵元素或零元素，并且这些相同矩阵元素或零元素的分布有一定规律性的矩阵。常见特殊矩阵有对称矩阵、上（下）三角矩阵、对角矩阵等

特殊矩阵的压缩存储方法：找出特殊矩阵中值相同的矩阵元素的分布规律，把那些呈现规律性分布的值的多个矩阵元素压缩存储到一个存储空间中

3）对称矩阵

元素下标的对应关系

k = i \* ( i - 1 ) / 2 + j - 1 ; i >= j

k = j \* ( j - 1 ) / 2 + i - 1 ; i < j

三角矩阵

下三角矩阵元素下标的对应关系

k = i \* ( i - 1 ) / 2 + j - 1 ; i >= j

k = n \* ( n + 1 ) / 2 ; i < j

上三角矩阵元素下标的对应关系

k = ( i - 1 ) \* ( 2n - i + 2 ) / 2 + ( j - i ) ; i <= j

k = n \* ( n + 1 ) / 2 ; i > j

三对角矩阵

元素下标的对应关系

k = 3 \* ( i - 1 ) - 1 + j - i + 1 = 2i + j - 3

已知k求i、j

i = [ ( k + 1 ) / 3 + 1 ] ; j = k - 2i + 3

4）稀疏矩阵

矩阵元素个数远大于非零元素个数的矩阵

一般采用三元组（行标，列标，值）的形式存储

稀疏矩阵压缩存储后失去随机存取特性

## 四、树和二叉树

1．掌握树型结构的定义。

2．掌握二叉树的定义、性质及各种存贮结构。

3．掌握遍历二叉树、线索二叉树及其他基本操作。

4．掌握树、森林与二叉树的相互转换；理解树的遍历；掌握哈夫曼树及其应用。

## 五、图

1．掌握图的定义和术语。

2．掌握图的存贮结构；理解图的基本操作。

3．掌握图的遍历算法；了解利用图的遍历解决图的应用问题。

4．理解图的有关应用：求最小生成树、求最短路径、拓扑排序及关键路径等算法的基本思想。

## 六、查找

1．掌握静态查找表。

2．掌握二叉排序树和平衡二叉树。

3．理解B－树；了解B＋树。

4．掌握哈希表。

5．掌握各种查找方法的时间性能分析。

## 七、内部排序

1．掌握直接插入排序、希尔排序、冒泡排序、快速排序、简单选择排序、堆排序、归并排序；理解基数排序。

2．学会各种内部排序方法的比较（时间复杂度、空间复杂度、稳定性）。

----------------

版权声明：本文为CSDN博主「大梦谁先觉i」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/qq\_45228323/article/details/105831244