

- 一、串
- 二、KMP算法
- 三、矩阵
- 四、广义表
- 五、作业习题

本系列博客为《数据结构》（C语言版）的学习笔记（上课笔记），仅用于学习交流和自我复习

数据结构合集链接：[《数据结构》C语言版（严蔚敏版） 全书知识梳理（超详细清晰易懂）](#)

这一章实在没什么东西，我就放一些有点用的ppt和作业题吧(这一章的作业题写着是真的烦)

一、串

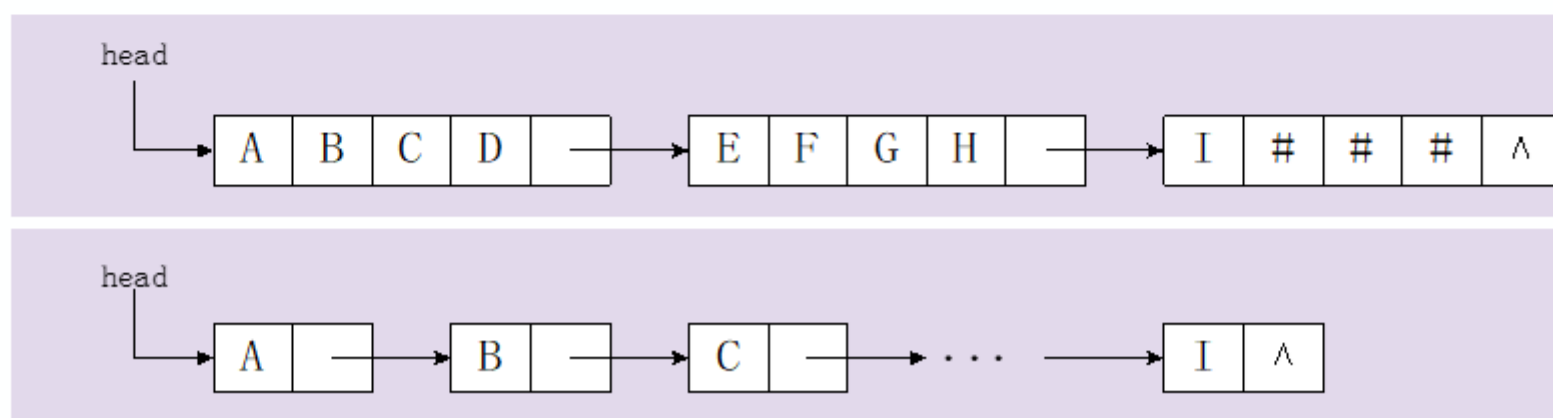
链式存储表示

优点：操作方便

缺点：存储密度较低

$$\text{存储密度} = \frac{\text{串值所占的存储位}}{\text{实际分配的存储位}}$$

可将多个字符存放在一个结点中，以克服其缺点



https://blog.csdn.net/weixin_45697774

二、KMP算法

0x15.基本数据结构 — 字符串（KMP算法（含详细证明）和最小表示法）

三、矩阵

1

什么是压缩存储?

若多个数据元素的值都相同, 则只分配一个元素值的存储空间, 且零元素不占存储空间。

2

什么样的矩阵能够压缩?

一些特殊矩阵, 如: 对称矩阵, 对角矩阵, 三角矩阵, 稀疏矩阵等。

3

什么叫稀疏矩阵?

矩阵中非零元素的个数较少 (一般小于5%)

1. 对称矩阵

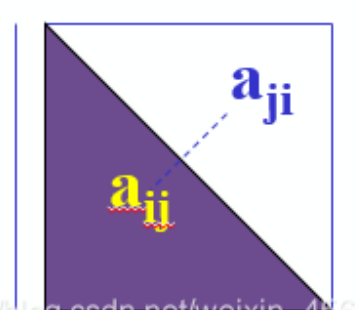
[特点] 在 $n \times n$ 的矩阵 a 中, 满足如下性质:

$$a_{ij} = a_{ji} \quad (1 \leq i, j \leq n)$$

[存储方法] 只存储下(或者上)三角(包括主对角线)的数据元素。共占用 $n(n+1)/2$ 个元素空间。

sa	a_{11}	a_{21}	a_{22}	a_{31}		$a_{ij}(a_{ji})$		a_{nn}
k	1	2	3	4				$n(n+1)/2$

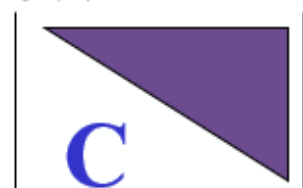
$$k = \begin{cases} i(i-1)/2 + j & \text{当 } i \leq j \\ j(j-1)/2 + i & \text{当 } i > j \end{cases}$$



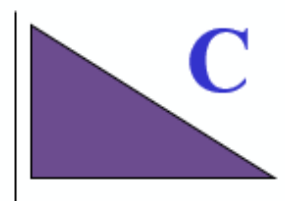
https://blog.csdn.net/weixin_45697774

2. 三角矩阵

[特点] 对角线以下(或者以上)的数据元素(不包括对角线)全部为常数 c 。



上三角矩阵



下三角矩阵

[存储方法] 重复元素 c 共享一个元素存储空间, 共占用 $n(n+1)/2 + 1$ 个元素空间: $sa[1.. n(n+1)/2 + 1]$

上三角矩阵

$$\begin{cases} k = (i-1) \times (2n-i+2)/2 + j - i + 1 & i \leq j \\ n(n+1)/2 + 1 & i > j \end{cases}$$

下三角矩阵

$$\begin{cases} k = i \times (i-1)/2 + j & i \leq j \\ n(n+1)/2 + 1 & i > j \end{cases}$$

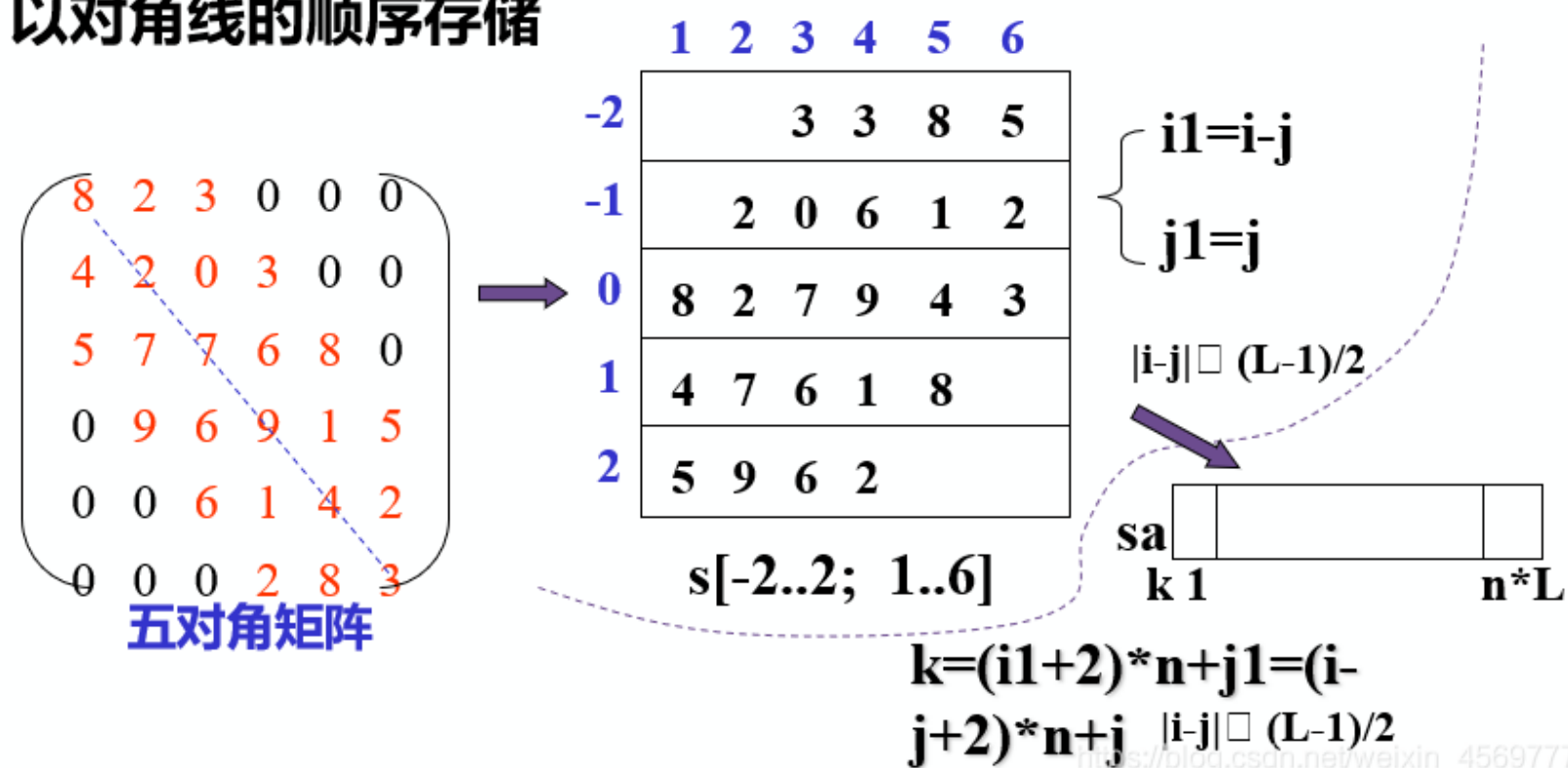
https://blog.csdn.net/weixin_45697774

▶▶▶ 3. 对角矩阵（带状矩阵）

[特点] 在 $n \times n$ 的方阵中，非零元素集中在主对角线及其两侧共 L (奇数) 条对角线的带状区域内 — L 对角矩阵。

[存储方法]

以对角线的顺序存储

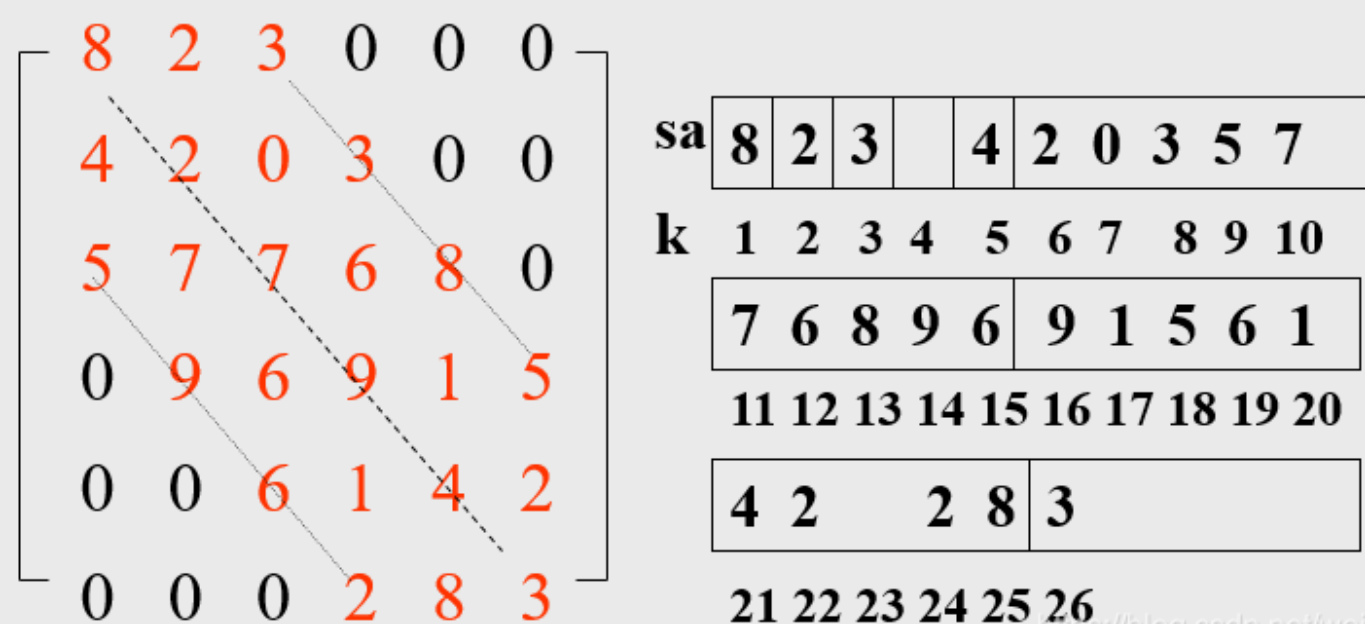


▶▶▶ 3. 对角矩阵（带状矩阵）

只存储带状区内的元素

除首行和末行，按每行 L 个元素，共 $(n-2)L+(L+1)$ 个元素。 $sa[1..(n-1)L+1]$

$$k=(i-1)L+1+(j-i) \quad |i-j| \leq (L-1)/2$$



▶▶▶ 稀疏矩阵

[特点]

大多数元素为零。

[常用存储方法]

只记录每一非零元素 (i, j, a_{ij})

节省空间，但丧失随机存取功能

顺序存储：三元组表

链式存储：十字(正交)链表

15	0	0	22	0	-15
0	11	3	0	0	0
0	0	0	-6	0	0
0	0	0	0	0	0
91	0	0	0	0	0
0	0	28	0	0	0

https://blog.csdn.net/weixin_45697774 6 □

四、广义表

▶▶▶ 广义表的基本运算

求表头GetHead(L)

01

非空广义表的第一个元素，可以是一个单元素，也可以是一个子表

求表尾GetTail(L)

04

非空广义表除去表头元素以外其它元素所构成的表。表尾一定是一个表

https://blog.csdn.net/weixin_45697774

▶▶▶ 练习

A=()



GetHead和GetTail均无定义

A=(a,b)



GetHead(A)=a GetTail(A)=(b)

A=(a)



GetHead(A)=a GetTail(A)=()

A=((a))



GetHead(A)=(a) GetTail(A)=()

A=(a,b,(c,d),(e,(f,g)))

GetHead(GetTail(GetHead(GetTail(GetTail(A)))))

d

https://blog.csdn.net/weixin_45697774

五、作业习题

1.串是一种特殊的线性表，其特殊性体现在（）

- A.可以顺序存储
- B.数组元素是一个字符
- C.可以连续存储
- D.数据元素可以是多个字符

答案：B

解析

串又称为字符串，是一种特殊的线性表，其特殊性体现在数据元素是一个字符，也就是说串是一种内容受限的线性表。（栈和队列是操作受限的线性表）

2.若串S= “software” ，其子串的个数是（ ）。(2分)

- A.8
- B.37
- C.36
- D.9

答案：B

解析：

串中n个不同的字符，共有 $(n(n+1))/2$ 个真子串，空串也算的话再+1。

题中8个字符， $(8 \times 9) \div 2 = 36$ ，加上空串就是37。

3.假设以行序为主序存储二维数组A=array[1..100, 1..100]，设每个数据元素占2个存储单元，基地址为10，则LOC[5, 5]=()。

- A. 808
- B. 818
- C. 1010
- D. 1020

答案

B

解析

公式： $Loc(A_{ij}) = 10(\text{基地址}) + [(5 - 1) * 100 + (5 - 1)] * 2 = 818$ 。

4.数组A[0..4,-1..-3,5..7]中含有元素的个数()

- A55
- B45
- C36
- D16

答案：B

解析：三维数组：

长的边 个数：4-0+1=5

宽的 边 个数： (-1) - (-3) +1=3

高的边个数： 7-5+1=3

总个数： 5*3*3

5.设有一个10阶的对称矩阵A，采用压缩存储方式，以行序为主存储，a_{1,1}为第一元素，其存储地址为1，每个元素占一个地址空间，则a_{8,5}的地址是__。

- A. 13
- B. 33
- C. 18
- D. 40

答案

B

答案解析

这里数组下标从1开始，对称矩阵，只存储其**下三角形元素**，在a8,5的前面有7行，第1行有1个元素，第2行有2个元素，...，第7行有7个元素，这7行共有 $(1+7) \times 7 / 2 = 28$ 个元素，在第8行中，a8,5的前面有4个元素，所以，a8,5前有 $28+4=32$ 个元素，其地址为33。

6.数组A[0..5,0..6]的每个元素占5个字节,将其按列优先次序存储在起始地址为1000的内存单元中,则元素A[5,5]的地址是

答案：1175

这类题目分为两类：行优先和列优先（假设数组为A [n] [m]，数组下标从0开始）

1.行优先：基址 + (行数) * m + 列数) * 每个元素所占内存单位

2.列优先：基址 + (列数 * n + 行数) * 每个元素所占内存单位，即 $1000 + (5 \times 6 + 5) \times 5 = 1175$

7.广义表((a,b,c,d))的表头和表尾分别是什么？

显然,广义表((a,b,c,d))中只有1个元素,

即(a,b,c,d)

表头是(a,b,c,d),一个子表

表尾是空表()长度为0

8.设广义表L= ((a, b, c)), 则L的长度和深度分别是 ()

- A.1和1
- B.1和3
- C.1和2
- D.2和3

答案：C

只有一个元素长度是1.唯一的元素里嵌套一层，所以深度是2

9.下面的说法不正确的是 ()。

- A.广义表的表头总是一个广义表
- B.广义表的表尾总是一个广义表
- C.广义表难以用顺序存储结构
- D.广义表可以是一个多层次的结构

答案：A

10.已知广义表LS=((a,b,c),(d,e,f)),运用head和tail函数取出LS中原子e的运算是

tail操作是除掉头的所有，包含括号

LS= ((a,b,c),(d,f))故:

tail(LS)= ((d,e,f))

head(tail(LS))=(d,e,f)

$\text{tail}(\text{head}(\text{tail}(\text{Ls}))) = (e, f)$

$\text{head}(\text{tail}(\text{head}(\text{tail}(\text{ls})))) = e$

11. 设有数组 $A[i, j]$ ，数组的每个元素长度为 3 字节， i 的值为 1 到 8， j 的值为 1 到 10，数组从内存首地址 BA 开始顺序存放，当用以列为主存放时，元素 $A[5, 8]$ 的存储首地址为（ ）。(2分)

A. BA+141

B. BA+180

C. BA+222

D. BA+225

答案：B

这里是以列为主存放

本题 $A[5, 8]$ 以列为主，该元素处于第八列，前七列是满的每列8个元素，该元素处于第五行，他的前一个元素 $A[4, 8]$ 的结束地址就是所求的开始，最后，每个元素占3。所以有公式：

$$(7 * 8 + 4) * 3 = 180$$

12. 若有 n 阶对称矩阵 A ，以行序为主序方式，将其下三角形的元素(包括主对角线上所有元素)依次存放于一维数组 $B[1..(n(n+1))/2]$ 中，则在 B 中确定 $a[i, j]$ ($i < j$) 的位置 k 的关系为（ ）。(2分)

A. $i * (i - 1) / 2 + j$

B. $j * (j - 1) / 2 + i$

C. $i * (i + 1) / 2 + j$

D. $j * (j + 1) / 2 + i$

答案：B

n 阶对称矩阵中的元素满足下述条件： $a_{ij} = a_{ji}$ ，($1 \leq i, j \leq n$)。对称矩阵中的每一对数据元素可以共用一个存储空间，因此可以将 n^2 个元素压缩存储到 $n(n+1)/2$ 个元的空间中，即可以一维数组保存。

假设用一维数组 $B[n(n+1)/2]$ 作为对称矩阵 A 的存储结构，则 $B[k]$ 和矩阵元素 a_{ij} 的下标 i, j 的对应关系为：

当 $i > j$ 时， $k = i(i-1)/2 + i$;

当 $i < j$ 时， $k = j(j-1)/2 + i$;

(以上公式是针对 a_{ij} 和 a_{ji} 保存在同一个单元中的情况)

因为存储下三角元素，所以 $i < j$ ， $k = j(j-1)/2 + i$ 。

13. 二维数组 A 的每个元素是由 6 个字符组成的串，其行下标 $i = 0, 1, \dots, 8$ ，列下标 $j = 1, 2, \dots, 10$ 。若 A 按行先存储，元素 $A[8, 5]$ 的起始地址与当 A 按列先存储时的元素() 的起始地址相同。设每个字符占一个字节。

A. $A[8, 5]$

B. $A[3, 10]$

C. $A[5, 8]$

D. $A[0, 9]$

正确答案

B

答案解析

二维数组就是平常所说的矩阵，也可以看成是数据元素是一维数组的一维数组。由于计算机内部存储器的地址是一维线性排列的，故在存储数据时，必须将二维数组转换为计算机的内部存储形式才可以。一般有两种存储方式：以行为主的存储方式(row-major)和以列为主的存储方式(column-major)。

设二维数组 $A[1:U_1, 1:U_2]$ ，该数组有 $(U_1 - 1) + 1 = U_1$ 行，有 $(U_2 - 1) + 1 = U_2$ 列。设每个元素占 d 个空间，数组的起始地址为 L_1 。

①以行为主(row-major)：也称行优先存储。其特点为：以每一行为单位，一行一行地放入内存，如C语言、PASCAL语言等都是如此处理二维数组的。

②以列为主(column-major)：也称列优先存储。其特点为：以每一列为单位，一列一列地放入内存，如Fortran语言就是如此处理二维数组的。二维数组A可以看成由 U_2 个一维数组组成，每个一维数组有 U_1 个元素。同 1)类似，有：

- 第j列数据元素的起始地址为： $(j - 1) \times U_1 \times d + L_1$
- 数组元素 $A[i, j]$ 的地址为： $Loc(A[i, j]) = L_1 + (j - 1) \times U_1 \times d + (i - 1) \times d$

重要说明：二维数组 $A, m \times n$ 的含义是：该数组有 m 行($0 \sim m - 1$ 第一维)，有 n 列($0 \sim n - 1$ ，第二维)，占用 $m \times n$ 个存储空间。假设每个数据元素占用 L 个存储单元(可理解为字节)，则二维数组A中任意一个元素 a_{ij} 的存储位置为：

行优先： $LOC(i, j) = LOC(0, 0) + (n \times i + j) \times L$

列优先： $LOC(i, j) = LOC(0, 0) + (m \times j + i) \times L$

二维数组是一种随机存储结构，因为存取数组中任一元素的时间都相等。(单链表则不是，因为需要顺链访问，访问时间同数据元素的存储位置有关)。

回到本题中，二维数组 $A[0:8, 1:10]$ ，设起始地址为 0，数组元素 $A[i, j]$ 按行存储公式为：

$Loc(A[i, j]) = L_1 + (i - 1) \times U_2 \times d + (j - 1) \times d$ ，数组元素 $A[i, j]$ 按列存储公式为：

$Loc(A[i, j]) = L_1 + (j - 1) \times U_2 \times d + (i - 1) \times d$ ，可得 $i = 3, j = 10$ 。

14. 设二维数组 $A[1..m][1..n]$ (即 m 行 n 列) 按行存储在数组 $B[1..m \times n]$ 中，则二维数组元素 $A[i][j]$ 在一维数组 B 中的下标为()

- A. $(i - 1) * n + j$
- B. $(i - 1) * n + j - 1$
- C. $i * (j - 1)$
- D. $j * m + i - 1$

$A[i, j]$ 在 i 行前有 $i - 1$ 行，就有 $(i - 1) * n$ 个元素，再加上它是在 j 列，所以就是 $(i - 1) * n + j$ ，注意这里数组下标是从 1 开始的，所以不需要减 1