

数据结构

本周范围：程序性能分析、栈、队列、跳表、特殊矩阵的压缩存储

张天戈



1. 分析以下程序片段的时间复杂度。 — 共执行了 $3m$ 条语句

(1) i=s=0; while (s<n) {i++; s+=i;}

$i = s = 0$

$2n \sim n$

$$S: \overset{2}{0+1+2+3+4+\dots+n} = \frac{m(m+1)}{2} \geq n$$

$$\frac{m^2}{2} \geq n$$

$$m^2 \geq n$$

$$m \geq \sqrt{n}$$

$$\underline{O(\sqrt{n})}$$

$$\boxed{n^{\frac{1}{2}}}$$

(2) $x=n; y=0;$ while ($x \geq (y+1)*(y+1)$) $y=y+1;$

~~3~~ n ~~1+1+1~~

m ^条
 n

$$= X < |x|$$

$$O(\sqrt{n})$$

$$x=n$$

$$2 \times 2$$

$$3 \times 3$$

⋮

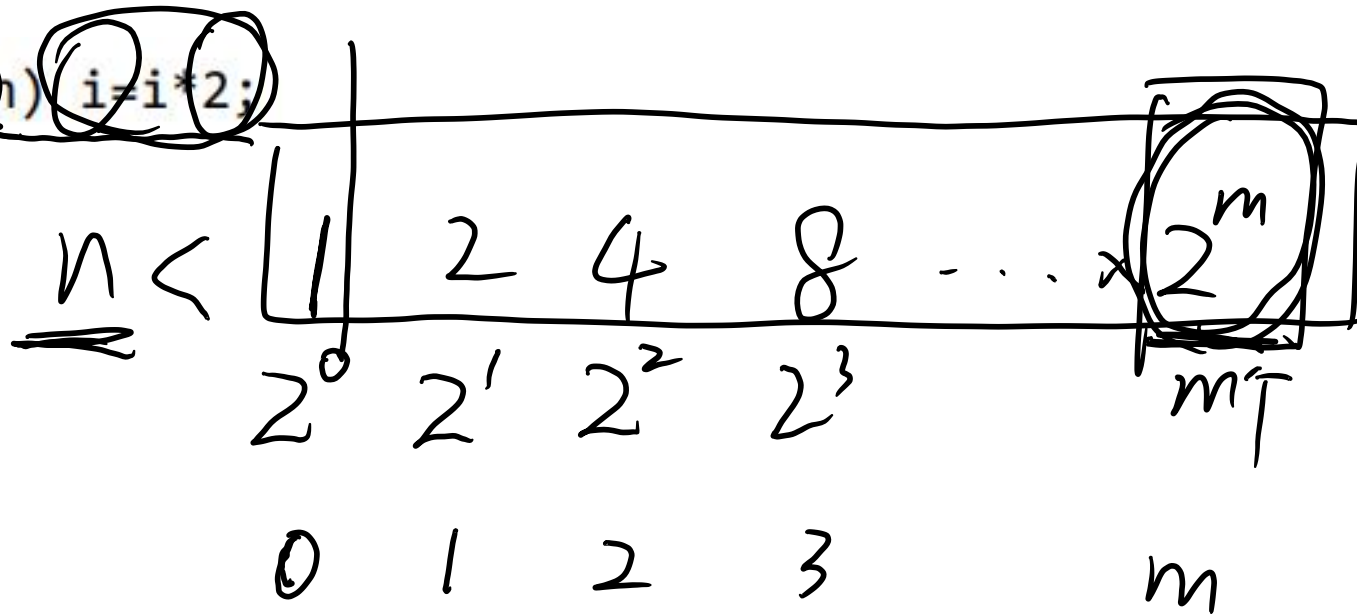
$$m \times m$$

$$n < m^2$$

$$m > \sqrt{n}$$

(3) $i=1$; while($i \leq n$) $i=i*2$;

m 条



$$\underline{2^m > n}$$

$$m > \underline{\log_2 n}$$

$$O(\underline{\log_2 n}) \quad O(\log n) \quad O(\lg n)$$

2. (1) 一个栈的进栈序列为abcde, 则不可能的出栈序列为:

A. edbca

B. decba

C. dceba

D. abcde

A

a b c
1 2 3

ab

D.

栈

A.

A.

decba

B.

C
dceba.

a b c d e

a b c d e

a b c d e

(2) 若已知一个栈的入栈顺序是 $1, 2, 3, 4, \dots, n$, 其输出序列为 $p_1, p_2, p_3, \dots, p_n$, 若 $p_1 = n$, 则 p_i 为?



$n \ n-1 \ n-2 \ \dots \ 1$

$$p_i = n - i + 1$$

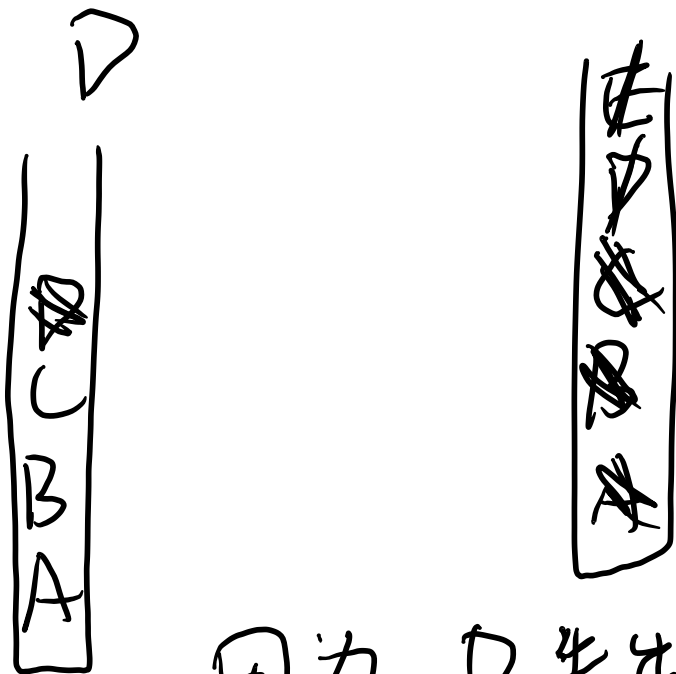
$$i = n$$

(3) 若元素的进栈序列为 A, B, C, D, E, 运用栈操作, 能否得到出栈序列 B, C, A, E, D 和 D, B, A, C, E, 为什么?

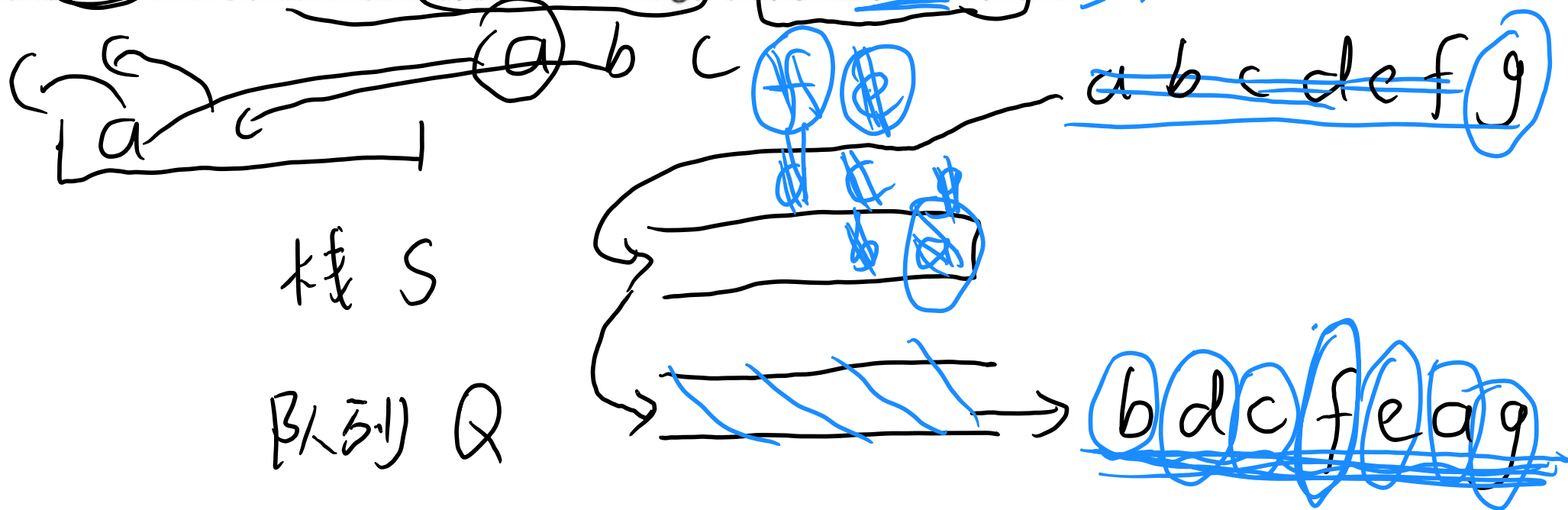
能... : , 因为 A 入栈, B 出栈, C 入栈
C A 出栈 - DE 进 - ED 出

BCAED ✓

因为 D 先出栈, 所以此时 AB C D 栈内 下一个不可能是 B,
因为 ↑

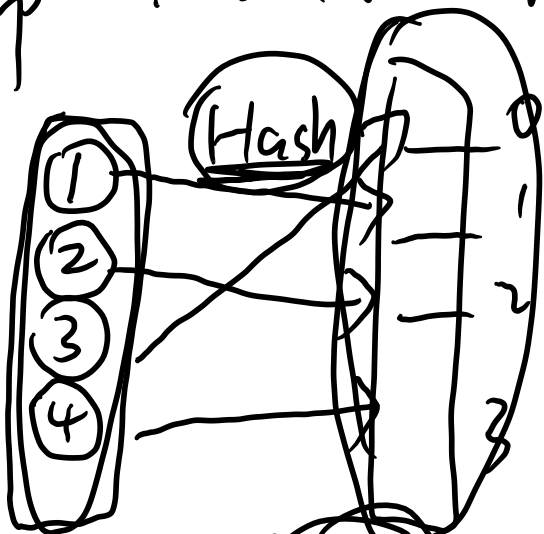


(4) 设栈S和队列Q的初始状态均为空，元素abcdefg依次进入栈S。若每个元素出栈后立即进入队列Q，且7个元素出队的顺序是bdcfeag，则栈S的容量至少是？ 3



3. 请比较跳表和哈希的区别与联系。

哈希：映射 h



① 结构决定性能

② 增删改查时

③ 遍出... 异常

跳表

二分思想

空间换时间

联系：

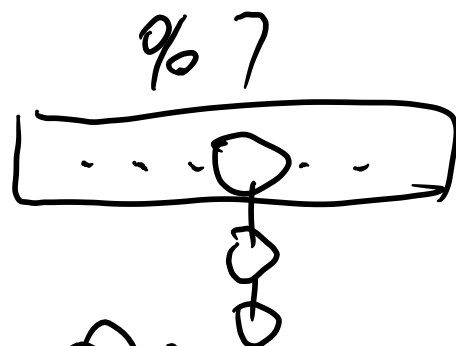
二者...

随机过优优化字典存储

区别：①

跳表

哈希



增删改查
空间

$O(\log n)$

最好 $O(1)$

最差 $O(n)$

③ 空间

④ 哈希，数组

跳... 链表

4. 请比较数组和链表的区别与联系。

	查	插	删	逆
数组	$O(n)$	<u>$O(1)$</u>	$O(n)$	
<u>有序数组</u>	$O(\lg n)$	$O(n)$	$O(n)$	
链表	$O(n)$	$O(1)$	$O(n)$	
<u>有序链表</u>	<u>$O(n)$</u>	$O(n)$	$O(n)$	

1. 存取（读写）方式

顺序表可以顺序存取，也可以随机存取，链表只能从表头顺序存取元素。例如在第 i 个位置上执行存或取的操作，顺序表仅需一次访问，而链表则需从表头开始依次访问 i 次。

2. 逻辑结构与物理结构

采用顺序存储时，逻辑上相邻的元素，对应的物理存储位置也相邻，而采用链式存储时，逻辑上相邻的元素，物理存储位置则不一定相邻，对应的逻辑关系是通过指针链接来表示的。

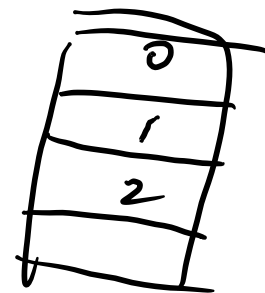
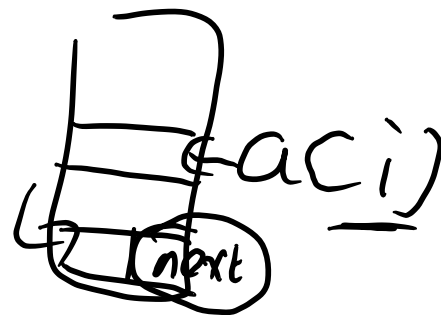
3. 查找、插入和删除操作

对于按值查找，顺序表无序时，两者的时间复杂度均为 $O(n)$ ；顺序表有序时，可采用折半查找，此时的时间复杂度为 $O(\log_2 n)$ 。

对于按序号查找，顺序表支持随机访问，时间复杂度仅为 $O(1)$ ，而链表的平均时间复杂度为 $O(n)$ 。顺序表的插入、删除操作，平均需要移动半个表长的元素。链表的插入、删除操作，只需修改相关结点的指针域即可。由于链表的每个结点都带有指针域，故而存储密度不够大。

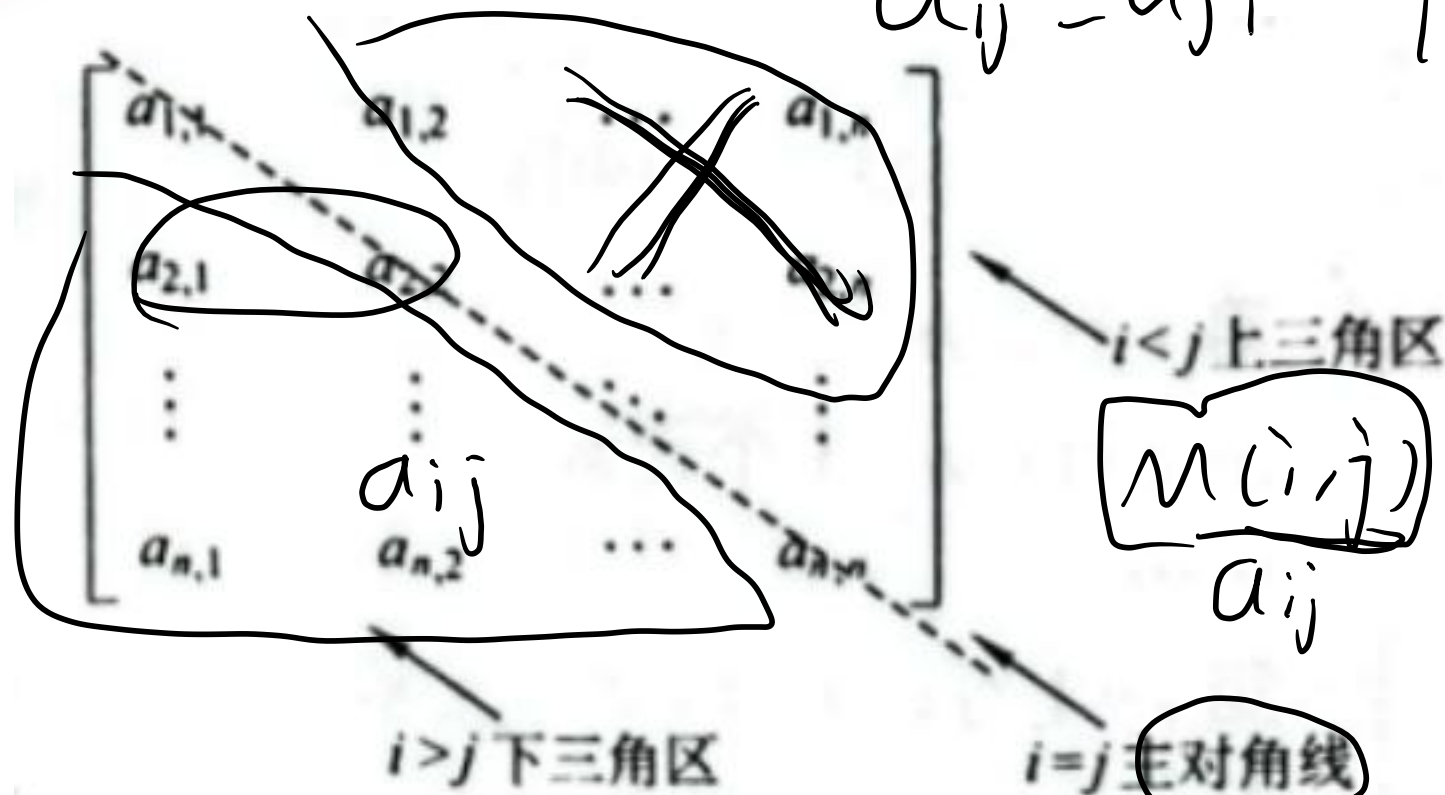
4. 空间分配

顺序存储在静态存储分配情形下，一旦存储空间装满就不能扩充，若再加入新元素，则会出现内存溢出，因此需要预先分配足够大的存储空间。预先分配过大，可能会导致顺序表后部大量闲置，预先分配过小，又会造成溢出。动态存储分配虽然存储空间可以扩充，但需要移动大量元素，导致操作效率降低，而且若内存中没有更大块的连续存储空间，则会导致分配失败。链式存储的结点空间只在需要时申请分配，只要内存有空间就可以分配，操作灵活、高效。

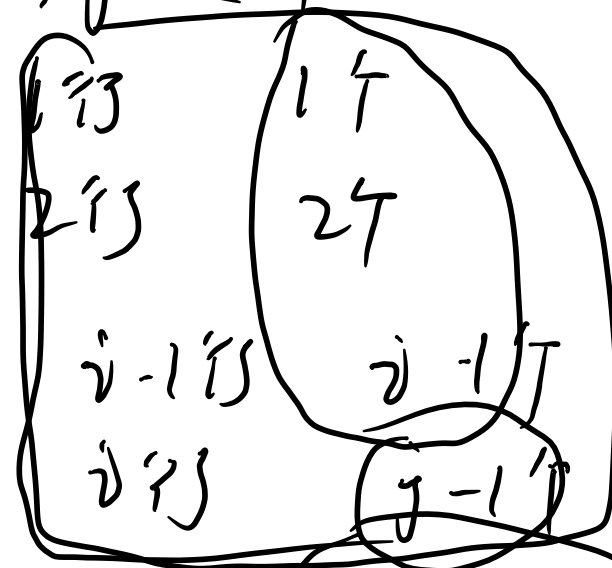


5. (总结题) 写出对称矩阵、三角矩阵、三对角矩阵和稀疏矩阵的存储映射函数。

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



$$M(i, j) = a_{ij}$$



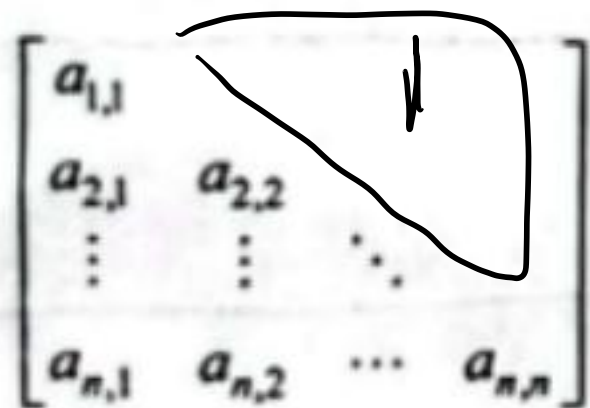
$$\frac{j(j-1)}{2} + j - 1$$

$$\frac{j(j-1)}{2} + j - 1$$

$$i \geq j$$

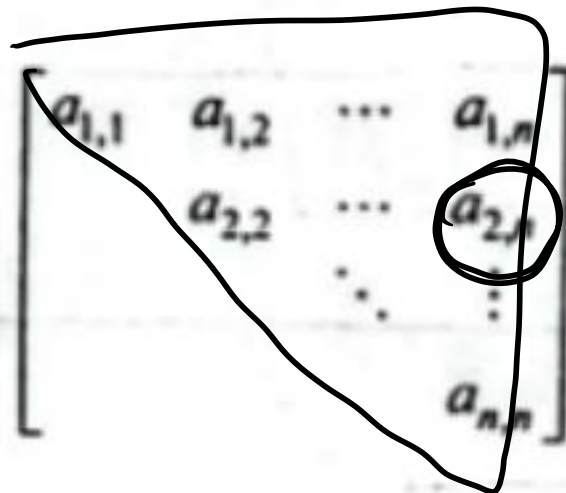
$$i < j$$





$$\begin{bmatrix} a_{1,1} & & & \\ a_{2,1} & a_{2,2} & & \\ \vdots & \vdots & \ddots & \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix}$$

(a) 下三角矩阵



$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ & a_{2,2} & \cdots & a_{2,n} \\ & & \ddots & \\ & & & a_{n,n} \end{bmatrix}$$

(b) 上三角矩阵

$$M(i, j) = \begin{cases} \frac{i(i-1)}{2} + j - 1 & i \geq j \\ \frac{n(n+1)}{2} & i < j \end{cases}$$

$$\begin{matrix} 1\ 1 \\ 2\ 2 \\ \vdots \\ i-1 \\ i \end{matrix}$$

$$\begin{matrix} n \\ n-1 \\ \vdots \\ n-i+2 \\ \textcircled{j+i} \end{matrix}$$

$$\begin{matrix} i-1 \\ i \end{matrix}$$

$$\begin{cases} \frac{i(i-1)(2n-i+2)}{2} + j - i & i \geq j \\ \frac{n(n+1)}{2} & i < j \end{cases}$$

当 $|i-j| > 1$ $a_{ij} = 0$

按行存储

a_{11}	a_{12}	a_{21}	a_{22}	a_{23}	$\vdots a_{ij}$
0	1	2	3	4	

$(i-1) \times 3 - 1 + j - i + 1$

$$M(i,j) = z \cdot i + j - 3$$

$$\begin{bmatrix}
 \underbrace{a_{11}} & \underbrace{a_{12}} & & & \\
 a_{21} & a_{22} & a_{23} & & \\
 & a_{32} & a_{33} & & \\
 & & \ddots & & \\
 & 0 & & & \\
 & & a_{n-1,n-2} & a_{n-1,n-1} & a_{n-1,n} \\
 & & \underbrace{a_{n,n-1}} & & a_{n,n}
 \end{bmatrix}$$

a_{ij}

图 3.25 三对角矩阵 A

$$M = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 6 & 0 \\ 0 & 9 & 0 & 0 \\ 0 & 23 & 0 & 0 \end{bmatrix}$$

对应的三元组

i	j	value
0	0	4
1	2	6
2	1	9
3	1	23

图 3.27 稀疏矩阵及其对应的三元组



Q & A