

建立 X 射线粉晶衍射数据库 的排序

沈建都 卓肇琨

矿物 X 射线物相鉴定, 实际上就是把欲鉴定试样的 X 射线粉晶衍射数据与已有的标准 X 射线粉晶衍射数据进行对比。为了使这种对比过程有规律地进行, 一个合乎逻辑的方法是利用电子计算机使检索过程完全自动化。

广泛利用电子计算机进行矿物 X 射线物相鉴定, 是今后发展的必然趋势。它可以大大提高效率, 而且只要程序编制恰当, 其结果也是可靠的。

本文着重介绍使用 APPLE II 型微机建立矿物 X 射线粉晶衍射数据库的排序问题。

数据库的排序原则

使用微机自动检索进行矿物物相鉴定, 是根据选定的检索方法编制程序实现的。我们用 MBASIC 语言编制的名为 SEARCH 的矿物 X 射线物相鉴定程序, 采用了类似哈那瓦尔特 (Hanawalt) 的检索方法, 其原理和方法的过程与人工检索并无多大差异, 不同的只是由微机来作出判断, 选出最佳的配对, 从而作出鉴定。

为满足 SEARCH 程序的设计要求, 首先必须建立一个标准矿物 X 射线粉晶衍射数据库。储存的数据主要是晶面间距 d 和相对强度 I/I_1 的数值, 储存的其他数据还应包括诸如试样的化学组成等有关的基本信息。储存用的数据一般都取自粉晶衍射标准联合委员会 (JCPDS) 出版的一套《粉晶衍射卡片》(《PDF》), 因为《PDF》包括的标准数据最多而且可靠性也大。

SEARCH 程序数据库的排序原则, 是从输入微机的一个矿物的《PDF》LN 条衍射线中, 提取八条强线的衍射数据排在前边, d 值按 I/I_1 值由大到小的顺序排列, 第八条强线之后的衍射线仍只按 d 值从大到小排列。《PDF》数据是按 d 值从大到小排列的, 所以将数据以顺序文本文件存入磁盘之前, 必须对输入微机的数据依排序原则进行处理, 这就是所谓排序问题。通过名为 EDPDF 的建库程序, 可完成排序和建库工作。

排序程序与实例

下面列出 EDPDF 程序中与排序问题有关的一段, 以石英的《PDF》前二十条衍射线

的数据为例, 结合程序说明排序过程。

```

1  DATA 20, 4.26, 35, 3.343 , 100, 2.458, 12, 2.282, 12, 2.237, 6, 2.128,
    9, 1.980, 6, 1.817, 17, 1.801, 1, 1.672, 7, 1.659, 3, 1.608, 1,
    1.541, 15, 1.453, 3, 1.418, 1, 1.382, 7, 1.375, 11, 1.372, 9, 1.288,
    3, 1.256, 4
510 DIM LN, DR (45), IR (45)
720 READ LN
730 FOR I = 1 TO LN
740 READ DR (I), IR (I)
750 NEXT I
755 DR (0) = DR (1); IR (0) = IR (1)
760 FOR I = 1 TO 8: A = IR (I): A2 = I
770 FOR J = I + 1 TO LN
780 IF A < IR (J) THEN A = IR (J): A2 = J
785 NEXT J
786 C = DR (A2): IF A2 = I THEN GOTO 810
790 FOR H = A2 TO I + 1 STEP -1: IR (H) = IR (H - 1): DR (H)
    = DR (H - 1): NEXT H
800 IR (I) = A
805 DR (I) = C
810 NEXT I

```

程序中几个变量的含义:

LN为输入的一个矿物《PDF》衍射线条数,

DR (I) 为第 I 条衍射线的 d 值,

IR (I) 为第 I 条衍射线的 I / I_1 值。

为将八条强线的 d、 I / I_1 数据调到前边, 程序中利用了循环语句的嵌套, 依靠微机内部堆栈指针, 借用中间变量 A、A2、C、H, 执行语句 755—810, 完成对输入数据的排序工作。

语句 510 为数组说明语句, 定义数组大小。执行语句 720—750, 微机就从语句 1 读入需要的数据, 并依次赋值给 LN、DR (I) 和 IR (I)。语句 760 说明要从输入的 LN 条衍射线中提取八条强线, 且表示外循环次数。语句 770 说明被比较的衍射线范围, 且表示内循环次数。

程序中给出了两个数据比较后的变换位置条件, 即 $A < IR (J)$ 。

最初微机读入的数据是语句 1 中的排列次序。第一个数 20 为输入的衍射线条数。

当执行到循环语句时, 微机先从原先输入的第一个 I / I_1 数据开始, 依次与后面十九个 I / I_1 数据比较, 用中间变量 A 将其中最大的 I / I_1 值保存起来。随之从最大的 I / I_1 值前一个 I / I_1 值开始, 直至第一个 I / I_1 值之间的所有 I / I_1 值, 依次向后移动一个位置,

然后用A的值取代第一个 I/I_1 值。这样在第一个位置上出现的 I/I_1 值就是所有输入的衍射线中最大的 I/I_1 值了。

接着微机从当前第二个 I/I_1 值开始,依次与后面十八个 I/I_1 值比较,再把其中最大的 I/I_1 值调到第二个 I/I_1 值的位置上。此时,在第二个位置上出现的 I/I_1 值就是所有输入的衍射线中第二强的了。如果当前开始比较的 I/I_1 值大于后面所有 I/I_1 值,微机就从下一个 I/I_1 值开始,与其后的强度数据比较。如此,微机每完成一次外循环,指针自动加1,直到从第八个 I/I_1 值开始,依次与排在后边的 I/I_1 值比较,在调动 I/I_1 值的同时,调动相应的d值,这样就将八条强线的d、 I/I_1 值调到了前边。

例如,当外循环 $I=1$ 时,令 $A=IR(1)$, $A2=1$,即从第一个 I/I_1 值开始, $A=35$;内循环则从2开始, $J=2$ 时满足换位条件,即 $35<100$,此时再令 $A=IR(2)=100$, $A2=J=2$,其后没有比100再大的 I/I_1 值了。令 $C=DR(A2)=DR(2)=3.343$,这样用中间变量A和C把要调到前边的数据100和3.343保存起来,再经中间变量H将 $DR(1)$ 和 $IR(1)$ 的值4.26和35赋值给 $DR(2)$ 和 $IR(2)$,最后将中间变量A和C保存的数据赋值给 $DR(1)$ 和 $IR(1)$ 。第一次外循环完成后数据的排列次序是:

```
3.343 100 4.26 35 2.458 12 2.282 12 2.237 6 2.128 9 1.98 6
1.817 17 1.801 1 1.672 7 1.659 3 1.608 1 1.541 15 1.453 3
1.418 1 1.382 7 1.375 11 1.372 9 1.288 3 1.256 4
```

当外循环 $I=3$,内循环 $J=8$ 时,又满足了换位条件,从下面数据的排列次序可以明显地看清语句790的执行情况。

```
3.343 100 4.26 35 1.817 17 2.458 12 2.282 12 2.237 6 2.128 9
1.98 6 1.801 1 1.672 7 1.659 3 1.608 1 1.541 15 1.453 3 1.418
1 1.382 7 1.375 11 1.372 9 1.288 3 1.256 4
```

第四次外循环完成后数据的排列次序是:

```
3.343 100 4.26 35 1.817 17 1.541 15 2.458 12 2.282 12 2.237 6
2.128 9 1.98 6 1.801 1 1.672 7 1.659 3 1.608 1 1.453 3 1.418
1 1.382 7 1.375 11 1.372 9 1.288 3 1.256 4
```

第七次外循环完成后数据的排列次序是:

```
3.343 100 4.26 35 1.817 17 1.541 15 2.458 12 2.282 12 1.375 11
2.237 6 2.128 9 1.98 6 1.801 1 1.672 7 1.659 3 1.608 1 1.453
3 1.418 1 1.382 7 1.372 9 1.288 3 1.256 4
```

第八次外循环完成后数据的排列次序是:

```
3.343 100 4.26 35 1.817 17 1.541 15 2.458 12 2.282 12 1.375 11
2.128 9 2.237 6 1.98 6 1.801 1 1.672 7 1.659 3 1.608 1 1.453
3 1.418 1 1.382 7 1.372 9 1.288 3 1.256 4
```

可见,已完成了要求的排序工作,可以把这样排列的数据储存到磁盘上去了。即便是更多条衍射线,只要是在数组定义范围内,均可利用该程序顺利、快速地完成排序工作。