

五、桥本分数式——优美的智慧

25 逆序乘积式

1. 问题提出

选择数字完成以下逆序乘积式

$$DE \times FG = ED \times GF \quad \text{①}$$

$$DEF \times GHK = FED \times KHG \quad \text{②}$$

$$DEFG \times K = GFED \quad \text{③}$$

每一式中的每一个字母代表一个数字，不同的字母代表不同的数字。

式①表述为：用 4 个不同的数字组成两个 2 位数，这两个 2 位数的乘积等于这两个 2 位数的逆序数的乘积。

式②表述为：用 6 个不同的数字组成两个 3 位数，这两个 3 位数的乘积等于这两个 3 位数的逆序数的乘积。

式③表述为：用 5 个不同的数字组成一个 4 位数与一个 1 位数，这个 4 位数乘以 1 位数的乘积等于这个 4 位数的逆序数。

试找出所有符合要求的逆序乘积式。为避免重复，约定式①中的 4 个数字中 D 为最小；式②中的 4 个百位数字 D, G, F, K 中 D 为最小。

2. 2 位逆序乘积式求解

(1) 求解思路

由 $de*fg=ed*gf$ 且 d 最小可知 e, g 必位于 d, f 之间。于是必有 $d \leq 6$ 且 $f \geq d+3$ 。由此建立如下 4 重循环：

d 从 1 递增到 6；

f 从 9 递减到 d+3；

e 从 d+1 递增到 f-1；

g 从 d+1 递增到 f-1；

若 $e! = g$ 且 $de*fg=ed*gf$ 则输出结果，并用 n 统计解的个数。

(2) 程序实现

* 二位逆积式 f251



```

? " 二位逆序式: DE*FG=ED*FG"
? ""
n=0
for d=1 to 6           && 设置 4 个数字循环
for f=9 to d+3 step -1
for e=d+1 to f-1
for g=d+1 to f-1
if e!=g and (10*d+e)*(10*f+g)=(10*e+d)*(10*g+f)
n=n+1
?? str(n,5)+": "+str(d*10+e,2)+"*"+str(f*10+g,2)+"="
?? str(e*10+d,2)+"*"+str(g*10+f,2)
if n%2=0
? ""
endif
endif
endif
endfor
endfor
endfor
endfor
? " 共有 "+str(n,2)+"个解。 "
return

```

(3) 程序运行结果

二位逆序式: DE*FG=ED*GF

```

1: 12*84=21*48      2: 14*82=41*28
3: 12*63=21*36      4: 13*62=31*26
5: 23*96=32*69      6: 26*93=62*39
7: 23*64=32*46      8: 24*63=42*36
9: 34*86=43*68      10: 36*84=63*48

```

共有 10 个解。

3. 3 位逆序乘积式求解

(1) 求解思路

为了寻求两个没有重复数字的三位数之积等于它们的逆序数之积, 设置 a, b 两个三位数循环, 分离出 a, b 的 6 个数字存放的 c 数组, 并分别计算 a, b 的逆序数 a1, b1。

根据题意, 若等式 $a*b=a1*b1$ 不成立, 或 D 大于 G, F, K, 则直接返回试下一组。否则, 应用 i, j 二重循环比较分离的 6 个数字是否有相同数字:

若存在相同, 则标注 t=1, 不作打印。

若不存在相同, 保持原有的 t=0, 作打印输出。

为确保不重复, 包括一边两个乘数交换的重复, 等号两边交换的重复。为此要求 a

的百位数字 $c(1) < c(4)$, $c(3)$, $c(6)$ 。

(2) 程序实现

```
* 三位逆序式 f252
dime c(6)
? " 三位逆序式: DEF*GHK=FED*KHG"
n = 0
for a=102 to 987
for b=a+1 to 999
    c(1)=int(a/100)
    c(2)=int(a/10)%10
    c(3)=a%10
    c(4)=int(b/100)
    c(5)=int(b/10)%10
    c(6)=b%10
    a1=c(3)*100+c(2)*10+c(1)
    b1=c(6)*100+c(5)*10+c(4)
    if a*b#a1*b1 or c(1)>c(4) or c(1)>c(3) or c(1)>c(6)
        loop
    endif
    t=0
    for i=1 to 5
    for j=i+1 to 6
        if c(i)=c(j)
            t=1
            exit
        endif
    endfor
endfor
if t=0
    n=n+1
    ? str(n,5)+" : "+str(a,3)+"*"+str(b,3)+"="
    ?? str(a1,3)+"*"+str(b1,3)
endif
endfor
endfor
return
```

(3) 程序运行结果

三位逆序式: DEF*GHK=FED*KHG

1: 134*862=431*268

2: 143*682=341*286

3: 314*826=413*628



4. 4 位逆序乘积式求解

(1) 求解思路

设置 n 循环穷举四位数, k 循环穷举倍数。

设置 i 循环, 循环 4 次, 每次实施 $m=m*10+t\%10$, $t=\text{int}(t/10)$, 求取 n 的逆序数 m 。

应用 $\text{at}()$ 函数判别 n 中是否存在相同数字, 判别 k 是否与 n 中数字相同。

应用 $m=k*n$ 进行判别, 作打印输出。

(2) 4 位逆序乘积式程序设计

* 4 位逆序乘积式 f253

```
set talk off
j=0
for n=1023 to 4987          && 在[1023, 4987]中穷举 n
for k=2 to 9                && k 穷举倍数
if at(str(k,1),str(n,4))>0
loop                        && 如果 k 与 n 某数字相同, 返回
endif
m=0
t=n
for i=1 to 4
x=t%10
if at(str(x,1),str(n,4),2)>0
m=0                        && 若 n 中有重复数字, 退出
exit
endif
m=m*10+x
t=int(t/10)                && 求整数 n 的逆序数 m
endfor
if m=k*n
? str(n,4)+"*"+str(k,1)+"="+str(m,4)
endif
endfor
endfor
return
```

(3) 程序运行结果

运行程序, 得

2178*4=8712

可知, 4 位逆序乘积式只有以上唯一解。

26 巧妙的三组平方

1. 问题提出

把 1, 2, ..., 9 这九个数字分成三个组, 每组三个数字, 使得这三个组中的三个数字分别能排列成平方数 a , b , c ($a < b < c$)。

设计程序, 求出满足要求的所有分法。

2. 设计要点

该题曾作为北京中学生智力竞赛试题。

设置 $a=a_1*a_1$, a_1 在 $11 \sim \sqrt{798}$ 中循环取值。

$b=b_1*b_1$, 因 $a > b$, 则 b_1 在 $a_1+1 \sim \sqrt{897}$ 中循环取值。

$c=c_1*c_1$, 而 $c > b$, 则 c_1 在 $b_1+1 \sim \sqrt{987}$ 中循环取值。

这样设置三重循环, 确保 a , b , c 都是三位平方数。

把 a , b , c 的 9 个数字转换为字符串 hd , 应用字符运算 $\$$ 判定 hd 中数字 1~9 是否各出现一次: 若 i ($i=1, 2, \dots, 9$) 出现在字符串 hd 中, 则表达式 $\text{str}(i, 1) \$ hd$ 为真; 若 i 不在 hd 中, 则 $\text{str}(i, 1) \$ hd$ 为假, 此时标注 $m=1$ 。

3. 程序实现

```
* 三组三数字平方 f261
set talk off
n=0
? "三组平方(每组三个数字)的所有分法为:"
for al=11 to sqrt(798)
  for bl=a1+1 to sqrt(897)
    for cl=b1+1 to sqrt(987)
      a=a1*a1          && 确保 a, b, c 均为三位平方数
      ad=ltrim(str(a))
      b=b1*b1
      bd=ltrim(str(b))
      c=c1*c1
      hd=str(c)+ad+bd
      m=0
      for i=1 to 9
        if !(str(i, 1) $ hd) && 测试每个数字是否为 1 次
          m=1
      exit
```



```

endif
endfor
if m=0
    n=n+1
    ? ltrim(str(a))+"("+ltrim(str(a1))+"^2)  "
    ?? ltrim(str(b))+"("+ltrim(str(b1))+"^2)  "
    ?? ltrim(str(c))+"("+ltrim(str(c1))+"^2)"
endif
endif
endfor
endfor
endfor
? "共"+ltrim(str(n))+ "种分法."
return

```

4. 程序运行结果

三组平方 (每组三个数字) 的所有分法为:
 361 (19²) 529 (23²) 784 (28²)
 共 1 种分法.

5. 一般三组平方

把 1, 2, ..., 9 这九个数字分成三个组, 每组至少一个数字, 使得这三个组中的所有数字分别能排列成平方数 a, b, c ($a < b < c$)。

设计程序, 求出满足要求的所有分法 (若某一分法中某组的数字能排列成不同的平方数, 只算一种分法)

(1) 设计要点

因 $a < b < c$, $a = a_1 * a_1$, $b = b_1 * b_1$, $c = c_1 * c_1$, 则 $a_1 < b_1 < c_1$ 。

注意到 a_1, b_1 都可取一位数, 于是设置 a_1 从 1 ~ sqrt(798) 循环取值, b_1 从 $a_1 + 1$ ~ sqrt(8976) 循环取值。因三组共 9 个数字, 由 a, b 的位数可确定 c 的位数 ln , 于是 c_1 从 c_0 ~ sqrt(10^{ln}) 循环取值, 其中 c_0 为 $b_1 + 1$ 与 sqrt($10^{(ln-1)}$) 的较大者。

这样设置的三重循环, 确保 a, b, c ($a < b < c$) 都为平方数, 且其数字共 9 个。

判定 a, b, c 中是否有重复数字同上。输出三组平方时增加了避免同一种分法重复打印的条件。

(2) 一般的三组数字平方程序设计

```

* 一般的三组数字平方 f262
set talk off
n=0
x=0
z=0

```

```

? "三组平方的所有分法为:"
? "      no      a      b      c"
for al=1 to sqrt(798)
  for bl=al+1 to sqrt(8976)
    a=al*al
    ad=ltrim(str(a))
    b=bl*bl
    bd=ltrim(str(b))
    ln=9-len(ad)-len(bd)      && ln 为第三个平方数的位数
    c0=max(bl+1,int(sqrt(10^(ln-1))+0.01))
    for cl=c0 to int(sqrt(10^ln))
      c=cl*cl
      hd=str(c)+ad+bd      && 三个平方数连成字符串
      m=0
      for i=1 to 9
        if !(str(i,1) $ hd)      && 测试是否有遗漏数字
          m=1
          exit
        endif
      endfor
      if m=0 and (x#a or z#c)
        n=n+1
        ? str(n)+": "+ltrim(str(a))+ "("+ltrim(str(al))+ "^2)  "
        ?? ltrim(str(b))+ "("+ltrim(str(bl))+ "^2)  "
        ?? ltrim(str(c))+ "("+ltrim(str(cl))+ "^2) "
        x=a
        z=c
        cl=10000
      endif
    endfor
  endfor
endfor
? "共"+ltrim(str(n))+ "种分法."
return

```

(3) 程序运行结果

三组平方的所有分法为:

	no	a	b	c
1 :	1	(1 ^2)	4 (2 ^2)	3297856 (1816 ^2)
2 :	1	(1 ^2)	49 (7 ^2)	872356 (934 ^2)
3 :	1	(1 ^2)	64 (8 ^2)	537289 (733 ^2)
4 :	1	(1 ^2)	256 (16 ^2)	73984 (272 ^2)



5 :	4 (2 ^2)	16 (4 ^2)	537289 (733 ^2)
6 :	4 (2 ^2)	25 (5 ^2)	139876 (374 ^2)
7 :	4 (2 ^2)	289 (17 ^2)	15376 (124 ^2)
8 :	9 (3 ^2)	324 (18 ^2)	15876 (126 ^2)
9 :	16 (4 ^2)	25 (5 ^2)	73984 (272 ^2)
10 :	16 (4 ^2)	784 (28 ^2)	5329 (73 ^2)
11 :	25 (5 ^2)	784 (28 ^2)	1369 (37 ^2)
12 :	25 (5 ^2)	841 (29 ^2)	7396 (86 ^2)
13 :	36 (6 ^2)	81 (9 ^2)	74529 (273 ^2)
14 :	36 (6 ^2)	729 (27 ^2)	5184 (72 ^2)
15 :	81 (9 ^2)	324 (18 ^2)	7569 (87 ^2)
16 :	81 (9 ^2)	576 (24 ^2)	3249 (57 ^2)
17 :	81 (9 ^2)	729 (27 ^2)	4356 (66 ^2)
18 :	361 (19 ^2)	529 (23 ^2)	784 (28 ^2)

共 18 种分法。

显然，最后一组解即为三组三位平方的解。

27 完美和式

1. 问题提出

把数字 1, 2, ..., 9 这九个数字分别填入下列两个和式的九个□中，数字 1~9 这 9 个数字在各式中出现且只出现一次（体现完美），使得各式成立。

$$\square\square\square + \square\square\square = \square\square\square \quad ①$$

$$\frac{\square}{\square + \square} + \frac{\square}{\square + \square} = \frac{\square}{\square + \square} \quad ②$$

为避免重复，和式①中右边三位数相同的为同一和式，即要求①所有和式的右边三位数不同。要求分数和式②中三个分数均为小于 1 的真分数；且左边两分数，分子小的在前；三个分母中的两个整数，较小的在前。

2. 完美整数和式①求解

(1) 设计要点

这是一道容易发生增解与遗解的填数趣题，要求填入每个数式的 9 个数字为 1, 2, ..., 9，既不遗漏，也不重复。约定以上和式中右边三位数相同的为同一和式，即要求所有和式的右边三位数不同。

计算并输出所有不同的和式。

设式左的前一个三位数为 a ，后一个三位数为 b ，为不至重复，设 $a < b$ 。

设置 c ， a 循环， c 为外循环，每打印输出一个解即跳出，保证一个 c 至多输出一个解。

每一对 c ， a ，有 $b=c-a$ 。把 a ， b ， c 转化为字符串相连为 d 。若 d 的长度不等于 9 或者 $b \leq a$ ，返回。

然后应用字符串 \$ 运算进行重复数字筛选。

筛选出的和式即为完美和式，输出并用 n 统计解的个数。

(2) 程序设计

* 十进制完美和式 f271

```
set talk off
n = 0
for c=312 to 987
for a=123 to 498
    b=c-a
    t=0
    d=ltrim(str(a))+ltrim(str(b))+ltrim(str(c))
    if len(d)#9 or b<=a      && 确保 a,b,c 为 9 个数字
        loop
    endif
    for i=1 to 9            && 测试有没有重复数字
        if !(str(i,1) $ d)
            t=1
            exit
        endif
    endfor
    if t=0                  && 打印所有和值 c 不同的和式
        ?? str(a,5)+" "+str(b,3)+"="+str(c,3)+" "
        n=n+1
        if mod(n,4)=0
            ? ""
        endif
        exit
    endif
endfor
endfor
? "共以上"+str(n,3)+"个."
return
```

(3) 程序运行结果

173+286=459	173+295=468	127+359=486	127+368=495
162+387=549	128+439=567	182+394=576	216+378=594



152+487=639	251+397=648	218+439=657	182+493=675
215+478=693	143+586=729	142+596=738	124+659=783
134+658=792	243+576=819	142+695=837	317+529=846
125+739=864	214+659=873	234+657=891	243+675=918
341+586=927	152+784=936	162+783=945	216+738=954
215+748=963	314+658=972	235+746=981	

共以上 31 个。

2. 完美分数和式②求解

(1) 设计要点

设置一维 a 数组，式②为

$$a(1)/(a(2)+a(3))+a(4)/(a(5)+a(6))=a(7)/(a(8)+a(9))$$

a(i) 在 1~9 中取值，出现数字相同 (a(i)=a(k)) 时返回。

当 i<9 时，还未取 9 个数，i 增 1 后 a(i)=1 继续；

当 i=9 时，且三个分数的分子分母符合要求时输出一个解。其中要求三个分数均小于 1，只要右边的分数小于 1 即可。

当 a(i)<9 时 a(i) 增 1 继续。

当 a(i)=9 时回溯或调整。直到 i=1 且 a(1)=9 时结束。

(2) 程序设计

* 完美分数和式 f272

```
set talk off
dime a(9)
a=0
i=1
a(1)=1
n=0
do while .t.
    g=1
    for k=1 to i-1
        if a(i)=a(k)
            g=0      && 两数相同, 标记 g=0 返回
            exit
        endif
    endfor
    if i=9 and g=1 and a(1)<a(4) and a(2)<a(3) and a(5)<a(6) and a(8)<a(9)
        m1=a(2)+a(3)
        m2=a(5)+a(6)
        m3=a(8)+a(9)
        if a(1)*m2*m3+a(4)*m1*m3=a(7)*m1*m2 and a(7)/m3<1
            n=n+1      && 判断是否满足等式
            ?? "("+str(n,2)+"): "
```

```

?? str(a(1),1)+"/("+str(a(2),1)+"+"+str(a(3),1)+")+"
?? str(a(4),1)+"/("+str(a(5),1)+"+"+str(a(6),1)+")="
?? str(a(7),1)+"/("+str(a(8),1)+"+"+str(a(9),1)+")  "
if n%2=0
    ? ""
endif
endif
endif
if i<9 and g=1
    i=i+1
    a(i)=1                && 不到9个数,往后继续
    loop
endif
do while a(i)=9 and i>1
    i=i-1                && 往前回溯
enddo
if i=1 and a(i)=9
    exit                && 至第1个数为9时结束
else
    a(i)=a(i)+1
endif
enddo
return

```

(3) 程序运行示例

运行程序, 得

(1): $1/(2+3)+4/(5+7)=8/(6+9)$ (2): $1/(2+4)+5/(7+8)=6/(3+9)$
 (3): $1/(2+4)+6/(3+9)=8/(5+7)$ (4): $1/(2+4)+6/(5+7)=8/(3+9)$
 (5): $1/(2+6)+9/(4+8)=7/(3+5)$ (6): $1/(2+9)+5/(3+8)=6/(4+7)$
 (7): $1/(2+9)+5/(4+7)=6/(3+8)$ (8): $1/(3+5)+9/(4+8)=7/(2+6)$
 (9): $1/(3+8)+5/(2+9)=6/(4+7)$ (10): $1/(3+8)+5/(4+7)=6/(2+9)$
 (11): $1/(4+7)+5/(2+9)=6/(3+8)$ (12): $1/(4+7)+5/(3+8)=6/(2+9)$
 (13): $1/(4+9)+2/(5+8)=3/(6+7)$ (14): $1/(4+9)+2/(6+7)=3/(5+8)$
 (15): $1/(5+8)+2/(4+9)=3/(6+7)$ (16): $1/(5+8)+2/(6+7)=3/(4+9)$
 (17): $1/(6+7)+2/(4+9)=3/(5+8)$ (18): $1/(6+7)+2/(5+8)=3/(4+9)$
 (19): $2/(3+9)+7/(6+8)=4/(1+5)$ (20): $2/(6+9)+4/(1+5)=8/(3+7)$

28 完美乘积式

1. 问题提出

把数字 1, 2, ..., 9 这 9 个数字分别填入下列各式的 9 个口中, 数字 1~9 这 9 个数字在各式中出现且只出现一次, 使得各式成立。



$$\square\square\times\square\square\square=\square\square\square\square$$

①

$$\square\square\square^2=\square\square\square\square\square\square$$

②

2. 完美乘积式求解

(1) 设计要点

要求填入每个数式的 9 个数字为 1, 2, ..., 9, 既不遗漏, 也不重复。

设二位数为 a, 三位数为 b。数 a 取 12~98, 数 b 取 123~987, 然后算出 $c=b*a$ 。

显然, 乘积 c 至少有 4 位。若 c 超过 4 位返回。

检查组成 a, b, c 的共 9 个数字是否为 1, 2, ..., 9, 既无遗漏也无重复。若满足这一要求, 打印相应的解, 并用变量 n 统计解的个数。

(2) 程序设计

* 完美乘积式 f281

set talk off

? " $\square\square*\square\square\square=\square\square\square\square$ "

n = 0

for a=12 to 98

for b=123 to 987

c=a*b

if c>9999

exit

endif

t=0

d=str(a,2)+str(b,3)+str(c,4) && 把 a, b, c 连成字符串

for i=1 to 9

if at(str(i,1),d)=0 && 应用 at 函数测试重复数字

t=1

exit

endif

endfor

if t=0

? str(a,2)+"*"+str(b,3)+"="+str(c,4)

n=n+1

endif

endfor

endfor

? "共以上"+str(n,2)+"个解."

return

(3) 程序运行结果

$$\square\square*\square\square\square=\square\square\square\square$$

12*483=5796
 18*297=5346
 27*198=5346
 28*157=4396
 39*186=7254
 42*138=5796
 48*159=7632
 共以上 7 个解。

3. 完美幂式求解

把幂式②适当拓展为： $x^n = z$ ， n 为大于 1 的整数，数字 1~9 在 x ， z 中出现且只出现一次。

(1) 设计要点

首先估计 x 的位数不会超过 3 位，因为 4 位数的平方至少 7 位，显然不符合题意要求。

设置 x 从 2~987 循环，对每一个 x ，设置 n 从 2~30 循环。

在 n 循环外， z 赋初值 $z=x$ 。在 n 循环中，赋值语句 $z=z*x$ 为 x 的 n 次方。把 x 与 z 连成字符串 st 。

若 st 的长度不为 9，则返回；若 st 中缺少数字 1~9 这 9 个数字中的某一个（即满足 $at(str(k, 1), st)=0$ ），标注 $g=1$ 返回。

若 st 的长度为 9 且 $g=0$ （即没有重复数字），输出完美幂式。

(2) 程序实现

```
* 完美幂式  $x^n = z$   f282
set talk off
clear
for x=2 to 987
    z=x
    for n=2 to 30
        z=z*x
        st=ltrim(str(x))+ltrim(str(z))
        if len(st)#9      && 不为 9 位则返回
            loop
        endif
        g=0
        for k=1 to 9      && 检查是否存在重复数字
            if at(str(k, 1), st)=0
                g=1
                exit
            endif
        endfor
    endfor
endfor
```



```

endfor
if g=0          && 满足条件输出结果
    ? ltrim(str(x))+"^"+str(n,2)+"="+ltrim(str(z))
endif
endfor
endfor
return

```

(3) 程序运行结果

$567^2=321489$

$854^2=729316$

共有两个完美幂式。

29 完美综合运算式

1. 问题提出

把数字 1, 2, ..., 9 这 9 个数字填入以下两个含加、减、乘、乘方(^)、除综合运算式中的 9 个□中, 使得下式成立

$$\square\square\times\square+\square\square\square\div\square-\square\square=0 \quad \textcircled{1}$$

$$\square^\square+\square\square\div\square\square-\square\square\times\square=0 \quad \textcircled{2}$$

要求数字 1, 2, ..., 9 这 9 个数字在各式中都出现且只出现一次, 且约定数字“1”不出现在乘、除、乘方的某位数中(即排除各式中的某位数为 1 这一平凡情形)。

2. 完美综合运算式①求解

(1) 求解要点

设式右的 5 个整数从左至右分别为 a, b, c, d, e, 其中 a, e 为二位整数, b, d 为大于 1 的一位整数, c 为三位整数。设置 a, b, c, d 循环, 对每一组 a, b, c, d, 计算 $e=a*b+c/d$ 。若其中的 c/d 非整数, 或所得 e 非二位数, 则返回。

然后将 5 个整数转换为字符串并相连为字符串 f, 应用 VFP 的 at() 函数检验数字 1~9 在字符串 f 是否各出现一次。若所有数字 1~9 在字符串 f 中各出现一次, 满足完美要求, 保持标记 t=0, 则输出所得的完美四则运算式。

设置 n 统计解的个数。

(2) 程序实现

* 完美四则运算式 f291

set talk off

? " □□*□+□□□/□-□□=0"

n = 0

```

for a=12 to 49
for b=2 to 8
for c=123 to 897
for d=2 to 9
    e=a*b+c/d
    if e#int(e) or e>99
        loop
    endif
    f=str(a,2)+str(b,1)+str(c,3)+str(d,1)+str(e,2)
    t=0                                && 把 a, b, c, d, e 连成字符串
    for i=1 to 9
        if at(str(i,1),f)=0          && 应用 at 函数测试重复数字
            t=1
            exit
        endif
    endfor
    if t=0
        n=n+1
        ? str(n,2)+": "+str(a,2)+"*"+str(b,1)+"+"
        ?? str(c,3)+"/"+str(d,1)+"-"+str(e,2)+"=0 "
    endif
endfor
endfor
endfor
endfor
? " 共以上"+str(n,2)+"个解."
return

```

(3) 程序运行结果与变通

$\square\square*\square+\square\square\square/\square-\square\square=0$

1: $12*4+376/8-95=0$

2: $17*3+258/6-94=0$

3: $35*2+168/7-94=0$

共以上 3 个解.

3. 完美综合运算式②求解

(1) 求解要点

式②含有加减乘除与乘方 5 种运算, 求解难度更大些。

设式右的 6 个整数从左至右分别为 a, b, z, c, d, e, 即要求的综合运算式为

$$a^b+z/c-d*e=0$$

其中 z, c, d 为 2 位整数, a, b, e 为大于 1 的一位整数。



设置 a, b, c, d, e 循环, 对每一组 a, b, c, d, e 计算 $z=(d*e-a^b)*c$ 。若计算所得 z 非二位数, 则返回。

然后将 6 个整数转换为字符串并相连为字符串 h, 应用 VFP 的 at() 函数检验数字 1~9 在字符串 h 中是否各出现一次。若所有数字 1~9 在字符串 h 中各出现一次, 满足完美要求, 保持标记 t=0, 则输出所得的完美综合运算式。

设置 n 统计解的个数。

(2) 程序实现

* 完美综合运算式 f292

set talk off

? " $\square^{\square}+\square\square/\square\square-\square\square*\square=0$ "

n = 0

for a=2 to 9

for b=2 to 9

for c=12 to 98

for d=12 to 98

for e=2 to 9

z=int($d*e-a^b$)*c

if c>99 or c<10

loop

endif

h=str(a,1)+str(b,1)+str(z,2)

h=h+str(c,2)+str(d,2)+str(e,2)

t=0

&& 把 a, b, c, d, e 连成字符串

for i=1 to 9

if at(str(i,1),h)=0

&& 应用 at 函数测试重复数字

t=1

exit

endif

endfor

if t=0

n=n+1

&& 输出四则运算式

? str(n,3)+" : "+str(a,1)+"^"+str(b,1)+"+"

?? str(z,2)+" / "+str(c,2)+" - "

?? str(d,2)+" * "+str(e,1)+" = "+str(z,2) " "

endif

endfor

endfor

endfor

endfor

endfor

? " 共以上"+str(n,2)+"个解."

return

(3) 程序运行结果

$$\square^{\square} \square + \square \square / \square \square - \square \square * \square = 0$$

$$1: 3^5 + 87 / 29 - 41 * 6 = 0$$

$$2: 7^3 + 28 / 14 - 69 * 5 = 0$$

$$3: 7^3 + 82 / 41 - 69 * 5 = 0$$

共以上 3 个解。

4. 程序变通

请修改上述程序，求把数字 1, 2, ..., 9 这 9 个数字不重复填入以下各综合运算式中的 9 个 \square 中，使得以下 3 个

$$\square \square \times \square + \square \square \div \square - \square \square \square = 0$$

$$\square^{\square} \square + \square \square \div \square - \square \times \square = \square \square$$

$$\square^{\square} \square - \square \times \square + \square \div \square = \square \square \square$$

含加减乘除与乘方的综合运算式成立。

30 桥本分数式

1. 问题提出

日本数学家桥本吉彦教授于 1993 年 10 月在我国山东举行的中日美三国数学教育研讨会上向与会者提出以下填数趣题：把 1, 2, ..., 9 这 9 个数字填入下式的九个 \square 中（不得重复），使下面的分数式成立

$$\frac{\square}{\square \square} + \frac{\square}{\square \square} = \frac{\square}{\square \square}$$

桥本教授当即给出了一个解答。这一分数等式填数趣题究竟共有多少个解答？试求出所有解答（等式左边两个分数交换次序只算一个解答）。

2. 穷举求解

(1) 求解要点

设分数式为 $b1/b2+c1/c2=d1/d2$ ，数字 1~9 在这 6 个变量中出现且只出现一次，即这 9 个数字组成的 9 位数能被 9 整除，最小为 123456789，最大为 987654321。以此设置穷举 a 循环，循环步长设置为 9 可缩减无效探索。

把 a 转换为字符串 d，应用 VFP 的 at(k, d) 函数检验数字 1~9 在 d 中是否有重复或



遗漏。当出现重复或遗漏 ($t=1$) 或 $b1>c1$ 时返回。由于约定 $b1<c1$, 所以穷举的最大数可以从 987654321 缩减为 897654321。

在没有重复或遗漏的情形下, 按顺序形成 $b1, b2, c1, c2, d1, d2$, 检验是否满足分数式, 转化为整数比较形式即是否满足条件 $b1*c2*d2+c1*b2*d2=d1*b2*c2$ 。若满足则打印输出。

(2) 程序实现

```
* 桥本分数式穷举求解 f301
set talk off
clear
s=0
? [ 桥本分数式有: ]
? []
for a=123456789 to 897654321 step 9    && 穷举循环设置
    d=str(a, 9)
    t=0
    for k=1 to 9                        && 检验数字 1~9 是否有遗漏
        if at(str(k, 1), d)=0
            t=1
            exit
        endif
    endfor
    if t=1
        loop                            && 出现重复数字
    endif
    b1=val(substr(d, 1, 1))             && 形成分数式各项的分母与分子
    b2=val(substr(d, 2, 2))
    c1=val(substr(d, 4, 1))
    c2=val(substr(d, 5, 2))
    d1=val(substr(d, 7, 1))
    d2=val(substr(d, 8, 2))
    if b1>c1
        loop                            && 约定左边第一项分母小于第二项分母
    endif
    if b1*c2*d2+c1*b2*d2=d1*b2*c2      && 满足分数式则输出解
        s=s+1
        ?? str(s, 2)+[: ]+str(b1, 1)+[/]+str(b2, 2)+[+] +str(c1, 1)
        ?? [/]+str(c2, 2)+[=]+str(d1, 1)+[/]+str(d2, 2)+[ ]
        if s%2=0
            ? []
        endif
    endif
endfor
```

```
endif
endfor
return
```

(3) 程序运行结果

```
1: 1/26+5/78=4/39      2: 1/32+5/96=7/84
3: 1/32+7/96=5/48      4: 1/78+4/39=6/52
5: 1/96+7/48=5/32      6: 2/68+9/34=5/17
7: 2/68+9/51=7/34      8: 4/56+7/98=3/21
9: 5/26+9/78=4/13      10: 6/34+8/51=9/27
```

3. 另一穷举求解

(1) 求解要点

设分数式为 $b_1/b_2+c_1/c_2=d_1/d_2$ ，注意等式左边两个分数交换次序只算一个解答，约定 $b_2 < c_2$ 。

对 3 个分数所涉及的 6 个设置循环穷举。

若分数式不成立，即 $b_1*c_2*d_2+c_1*b_2*d_2 \neq d_1*b_2*c_2$ ，则返回继续。

数字 1~9 在这 6 个变量中出现且只出现一次，分离出 9 个数字后用 f 数组统计各个数字的个数。因 VFP 数组下标不为 0，用数组元素 $f(x+1)$ 统计数字 x 的个数（如 $f[4]=2$ ，即数字“3”有 2 次）。当没有重复数字时打印输出解。

(2) 程序实现

```
* 桥本分数式另穷举实现 f302
set talk off
dime m(6), f(10)
f=0
n=0
for b2=12 to 97          && 设分数式为 b1/b2+c1/c2=d1/d2
for c2=b2+1 to 98
for d2=12 to 98          && 对 3 个分数的分子分母实施穷举
for b1=1 to 9
for c1=1 to 9
for d1=1 to 9
if b1*c2*d2+c1*b2*d2#d1*b2*c2
loop          && 若分数式不成立则返回
endif
m(1)=b1
m(2)=b2
m(3)=c1
m(4)=c2
m(5)=d1
```



```

m(6)=d2
for x=1 to 10
    f(x)=0
endfor
for k=1 to 6
    y=m(k)          && 分离数字并用 f 数组统计
    do while y>0
        x=y%10
        f(x+1)=f(x+1)+1  && 因下标不为 0, 故下标为 x+1
        y=int(y/10)
    enddo
endfor
t=0
for x=2 to 10
    if f(x) != 1
        t=1          && 检验数字 1~9 是否有重复
        exit
    endif
endfor
if t=0              && 输出一个解
    n=n+1
    ?? str(n,2)+": "+str(b1,1)+"/"+str(b2,2)+"+"
    ?? str(c1,1)+"/"+str(c2,2)+"="
    ?? str(d1,1)+"/"+str(d2,2)+" "
    if n%2=0
        ? ""
    endif
endif
endfor
endfor
endfor
endfor
endfor
endfor
endfor
return

```

(3) 程序运行结果与说明

1: 5/26+9/78=4/13	2: 1/26+5/78=4/39
3: 1/32+7/96=5/48	4: 1/32+5/96=7/84
5: 6/34+8/51=9/27	6: 9/34+2/68=5/17
7: 4/39+1/78=6/52	8: 7/48+1/96=5/32
9: 9/51+2/68=7/34	10: 4/56+7/98=3/21

以上两个穷举求解程序设计都比较简单，打印出的 10 个解相同，只是顺序不同。因为穷举的范围较大，致使程序运行时间较长。为了提高程序的求解效率，缩短程序求解时间，可改进应用回溯法求解。

关于桥本分数式求解，已有应用程序设计得到 9 个解的报导，遗失了一个解。可见在程序设计求解时，如果程序中结构欠妥或参量设置不当，或判定等式未作前述必要的整数等式转化，都可能造成增解或遗解。不要认为计算机求解就万无一失了，程序设计掉以轻心照样会造成失误。只要我们分析时稍缜密些，对运行结果作必要的检验，遗解增解是可以避免的。

31 埃及分数式

1. 问题提出

金字塔的故乡埃及，也是数学的发源地之一。古埃及数系中，记数常采用分子为 1 的分数，称为“埃及分数”。

人们研究较多且颇感兴趣的问题是：把一个给定的整数或分数转化为若干个不相同的埃及分数之和。当然，转化的方法可能有很多种。常把分解式中埃及分数的个数最少，或在个数相同时埃及分数中最大分母为最小的分解式称为最优分解式。把给定整数或分数分解为埃及分数之和，分解的优化往往是一个繁琐艰辛的过程。

例如，对 $5/121$ ，可分解为：

$$5/121=1/61+1/62+1/121+1/3782+1/7381+1/7382+1/54486542$$

为尽可能减少分解项数，数学家布累策在《数学游览》中给出了以下优化的三项分解式：

$$5/121=1/25+1/759+1/208725$$

同时布累策证明了： $5/121$ 不可能分解为两个埃及分数之和。

从项数来说，上述三项分解式不可能再优化了。但对最大分母来说，布累策的分解式不是最优的。我国两位青年数学爱好者于 1983 年发现：

$$5/121=1/27+1/297+1/1089 \quad \textcircled{1}$$

$$5/121=1/33+1/99+1/1089 \quad \textcircled{2}$$

$$5/121=1/33+1/121+1/363 \quad \textcircled{3}$$

$$5/121=1/33+1/91+1/33033 \quad \textcircled{4}$$

这 4 个分解式都比布累策的结论要优。人们通常约定分解式中不得包含与待分解分数同分母的埃及分数。从这个意义上，显然应把分解式③排除在外。因此，现在所知把 $5/121$ 分解为三个埃及分数的最小分母为 1089，即上述埃及分数分解式①，②。

那么，分解 $5/121$ 为三个埃及分数之和，其最大分母能否小于 1089 呢？我们可通过程序设计来探索。



2. 构建 3 个埃及分数分解式

(1) 设计要点

设指定的分数 m/d 的三个埃及分数的分母为 a, b, c ($a < b < c$)，最大分母不超过 z ，通过三重循环实施穷举。

确定 a 循环的起始值 $a1$ 与终止值 $a2$ 为：

$$\frac{1}{a1} = \frac{m}{d} - \frac{2}{z} \Leftrightarrow a1 = \frac{dz}{mz - 2d} \quad (\text{即把 } b, c \text{ 全放大为 } z)$$

$$a2 = \frac{3d}{m} + 1 \quad (\text{即把 } b, c \text{ 全缩减为 } a)$$

b 循环起始取 $a+1$ ，终止取 $z-1$ 。

c 循环起始取 $b+1$ ，终止取 z 。

对于三重循环的每一组 a, b, c ，计算 $x=mabc$ ， $y=d(ab+bc+ca)$ 。

如果 $x=y$ 且 b, c 不等于 d ，即满足分解为三个埃及分数的条件，打印输出一个分解式。然后退出内循环，继续寻求。

(2) 程序实现

* 构建指定数的 3 个埃及分数之和 f311

set talk off

input " 请确定指定数的分子: " to m

input " 请确定指定数的分母: " to d

input " 请确定分母的上界值: " to z

? " 把分数"+str(m,2)+"-"+str(d,3)+"分解为 3 个埃及分数之和:"

? " (分母不得为"+str(d,3)+"，最大分母不超过"+str(z,4)+")"

n=0

a1=int(d*z/(m*z-2*d))

a2=int(3*d/m)+1

for a=a1 to a2

for b=a+1 to z-1

for c=b+1 to z

x=m*a*b*c && 计算 x, y 值

y=d*(a*b+b*c+c*a)

if x=y and b#d and c#d && 输出分解式

n=n+1

? " NO"+str(n,1)+" : "+str(m,2)+"-"+str(d,3)+"=1/"

?? str(a,3)+"1/"+str(b,3)+"1/"+str(c,4)

exit

endif

endfor

endfor

endfor

? " 共上述"+str(n, 1)+"个分解式."

return

(3) 程序运行示例与讨论

请确定指定数的分子: 5

请确定指定数的分母: 121

请确定分母的上界值: 1100

把分数 5/121 分解为三个埃及分数之和:

(分母不得为 121, 最大分母不超过 1100)

N01: 5/121=1/ 27+1/297+1/1089

N02: 5/121=1/ 33+1/ 99+1/1089

N03: 5/121=1/ 45+1/ 55+1/1089

共上述 3 个分解式.

这样, 我们通程序设计得到: 分解 5/121 为三个埃及分数之和, 最大分母最小为 1089, 即不可能有比上述三个分解式更优的分解。

结果中的最后一个分解式是程序设计得到的新的最优分解式。

运行程序可为指定分数构建三个埃及分数的分解式。

请确定指定数的分子: 7

请确定指定数的分母: 169

请确定分母的上界值: 1000

把分数 7/169 分解为三个埃及分数之和:

(分母不得为 169, 最大分母不超过 1000)

N01: 7/169=1/ 27+1/338+1/ 702

N02: 7/169=1/ 28+1/338+1/ 364

N03: 7/169=1/ 30+1/195+1/ 338

N04: 7/169=1/ 39+1/ 78+1/ 338

共上述 4 个分解式.

3. 构建 4 个埃及分数分解式

有资料指出, 如果把 5/121 分解为 4 个埃及分数之和, 其分母能否小于 1089 呢? 对于这一还没有定论的分解问题, 我们继续应用程序设计进行探索。

(1) 设计要点

设指定的分数 m/d 的 4 个埃及分数的分母为 a, b, c, e ($a < b < c < e$), 最大分母不超过 z , 通过 4 重循环实施穷举。

确定 a 循环的起始值 a_1 与终止值 a_2 为:

$$\frac{1}{a_1} = \frac{m}{d} - \frac{3}{z} \Leftrightarrow a_1 = \frac{dz}{mz - 3d} \quad (\text{即把 } b, c, e \text{ 全放大为 } z)$$

$$a_2 = \frac{4d}{m} + 1 \quad (\text{即把 } b, c, e \text{ 全缩减为 } a)$$

b 循环起始取 $a+1$, 终止取 $z-2$ 。



c 循环起始取 b1, 终止取 z-1。

e 循环起始取 c+1, 终止取 z。

其中 b1 为 $\text{int}(1/(m/d-1/a-2/z))$, 当 $b1 > z-1$ or $b1 \leq a$ 时, 不予判别, 返回 a 循环使 a 增值。

对于 4 重循环的每一组 a, b, c, e, 计算 $x=mabce$, $y=d(abc+bce+cea+eab)$ 。

如果 $x=y$, 且 b, c, e 不等于 d, 即满足分解为 4 个埃及分数的条件, 打印输出一个分解式。然后退出内循环, 继续寻求。

(2) 构建 4 个埃及分数分解式程序设计

* 构建指定数的 4 个埃及分数之和 f312

input " 请确定指定数的分子: " to m

input " 请确定指定数的分母: " to d

input " 请确定分母的上界值: " to z

? " 把分数"+str(m,2)+"/" +str(d,3)+"分解为 4 个埃及分数之和:"

? " (分母不得为"+str(d,3)+" , 最大分母不超过"+str(z,4)+")"

n=0

a1=int(d*z/(m*z-3*d))

a2=int(4*d/m)+1

for a=a1 to a2

 b1=int(1/(m/d-1/a-2/z))

 if b1>z-1 or b1<=a

 loop

 endif

 for b=b1 to z-2

 for c=b+1 to z-1

 for e=c+1 to z

 x=m*a*b*c*e

 && 计算 x, y 值

 y=d*(a*b*c+b*c*e+c*e*a+e*a*b)

 if x/y>1+1/z

 exit

 endif

 if x=y and b#d and c#d and e#d && 输出分解式

 n=n+1

 ? " NO"+str(n,2)+" : " +str(m,2)+"/" +str(d,3)+"=1/"

 ?? str(a,3)+"1/" +str(b,3)+"1/" +str(c,3)+"1/" +str(e,3)

 exit

 endif

 endfor

 endfor

endfor

endfor

? " 共上述"+str(n, 1)+"个分解式."

return

(3) 程序运行结果与说明

请确定指定数的分子: 5

请确定指定数的分母: 121

请确定分母的上界值: 1000

把分数 $5/121$ 分解为 4 个埃及分数之和:

(分母不得为 121, 最大分母不超过 1000)

NO 1: $5/121=1/28+1/462+1/484+1/726$

NO 2: $5/121=1/30+1/220+1/484+1/726$

NO 3: $5/121=1/33+1/120+1/605+1/968$

NO 4: $5/121=1/33+1/132+1/484+1/726$

NO 5: $5/121=1/36+1/99+1/484+1/726$

NO 6: $5/121=1/44+1/66+1/484+1/726$

共上述 6 个分解式.

从以上 6 个分解式可知, 把 $5/121$ 分解为 4 个埃及分数之和, 最优分母为 726。

再次运行程序,

请确定指定数的分子: 3

请确定指定数的分母: 23

请确定分母的上界值: 100

把分数 $3/23$ 分解为 4 个埃及分数之和:

(分母不得为 23, 最大分母不超过 100)

NO 1: $3/23=1/12+1/46+1/69+1/92$

共上述 1 个分解式.