用微電腦計算一些會溢位的數值以增加其有效位數的方法

劉祥通
省立嘉義師院

一、前　　言

自從有了電腦，對於既費時又費力的大數計算我們有了較好的工具，然而電腦也有技窮之時，我們還是遇到一些超大數的溢位（ overflow ）與有效位數不夠精確的情況，例如 2 的 1000 乘冪， 991 ，3／23等，在此我們將一些已學過的數學理論加以妥切的運用提供一點技巧以饗同好。

二、分項計算——進位儲存

㈠概念的分析

數系的乘法對加法分配率 $a*(b+c)=a*b+a*c$，在很多範疇裏都討論到，而此簡單的運算——

$$3\times 25=3\times(20+5)=60+15=75$$

我們並不覺得這個基本理論的重要，但是它有一個很強的概念却是大家所忽略的——分項計算：

1. 先將 3 乘以 5 得到15，將進位 1 另外儲存。
2. 再計算 3 乘以十位數 2 得到 6 。
3. 並把 $1+6=7$ 得到十位數 7 ，個位數是 5 ，結果是75。

運用這個方法在超大數的處理上，可將大數 a 分項得

$$a=a_0+a_1\times 10^1+a_2\times 10^2+a_3\times 10^3+\cdots\cdots$$

若 b 為個位數時，

$$\begin{aligned} a\times b&=(a_0+a_1\times 10^1+a_2\times 10^2+a_3\times 10^3+\cdots\cdots)\times b\\ &=a_0b+a_1\times b\times 10^1+a_2\times b\times 10^2+a_3\times b\times 10^3+\cdots\cdots \end{aligned}$$

在此每項乘法計算 $a_1 \times b$ 中將進位累進至下一項中，這裏我們是藉用電腦的陣列變數（DIMENSION ）來處理，至於 b 乘以 a_0 ，$a_1 \times 10^1$ ，$a_2 \times 10^2$ ，……等項，我們是藉用BASIC 語言的廻路（ Loop)來處理此重覆相同的運算。

如果 b 是二位數以上，則將 b 拆成 $b_0 + b_1 \times 10^1 + b_2 \times 10^2 + \cdots\cdots$，每項依次乘以 $a_0 + a_1 \times 10^1 + a_2 \times 10^2 + \cdots\cdots$ 然後再相加，在此我們是在程式中多加一層廻路，以內外雙廻路來計算。

㈡程式的作法

1.計算正整數M的 P 次方

(1)筆者首先讓M爲個位數開始著手。以計算 2 的10 次方爲例，2^{10} 是由 2^9（= 512）× 2 而得，吾人先將已做出之 2^9 = 512 存入NUM（ 3 ），NUM（ 2 ），NUM（ 1 ）內，依序乘以 2 加上進位變數（ carry ）所存之值，得1024 分別存回NUM（ 4 ），NUM（ 3 ），NUM（ 2 ），NUM（ 1 ）內，此時位數變數（ DIGIT ）已由 3 變成 4 了。

程式結構

```
FOR E=1 TO P
計算M的P次方此廻路繞P次
  FOR I=1 TO DIGIT
    內廻路處理（M的（E－1）次方）乘以M
    若M的（E－1）次方有DIGIT位則繞DIGIT次
  NEXT I
NEXT E
```

程式如下：（程式一）

```
10 REM computer m^p
20 INPUT "請輸入底數";M
30 INPUT "請輸入指數";P
40 N=INT(P*(LOG (M)))+1
50 DIM NUM(N)
60 NUM(1)=1
70 DIGIT=1
80 FOR E=1 TO P
90 CARRY=0
100 FOR I=1 TO DIGIT
110 PRODUCT=NUM(I)*M+CARRY
120 NUM(I)=PRODUCT MOD 10
```

```
130 CARRY=PRODUCT\10
140 NEXT I
150 IF CARRY >0 THEN DIGIT=DIGIT+1:NUM(DIGIT)=CARRY
160 NEXT E
170 PRINT M;"^";P;"=";
180 FOR I=DIGIT TO 1 STEP -1
190 PRINT USING"#";NUM(I);
200 NEXT I
```

```
RUN
請輸入底數? 2
請輸入指數? 1000
 2 ^ 1000 =107150860718626732094842504906000181056140481170553360744375038837035
1051124936122493198378815695858127594672917553146825187145285692314043598457757 4
6985748039345677748242309854210746050623711418779541821530464749835819412673987 6
7559165543946077062914571196477686542167660429831652624386837205668069376
```

(2)其次再考慮M為二位數的情況，以12的３次方為例，外廻路必須跑３次（變數E）

```
   144        FOR E=1 TO 3
 ×  12          FOR I=1 TO DIGIT
 ------           此廻路計算 144×2=288        並將 288 存入
   288            此廻路計算 144×10=1440      並將 1440 存入
  144             此廻路計算 288+1440=1728
 ------         NEXT I
  1728        NEXT E
```

程式如下：（程式二）

```
1000 REM computer m^p
1010 INPUT "請輸入底數 ";M
1020 INPUT "請輸入指數 ";P
1030 N=INT(P*(LOG (M)))+1
1040 DIM NUM(N):DIM NUM1(N):DIM NUM2(N):DIM NUM3(N)
1050 M1=M MOD 10:NUM(1)=1
1060 M2=M \10:NUM(2)=1
1070 DIGIT=1
1080 FOR E=1 TO P
1090 CARRY1=0
1100 DIGIT1=DIGIT
1110 FOR I=1 TO DIGIT1
1120 PRODUCT1=NUM(I)*M1+CARRY1
```

```
1130 NUM1(I)=PRODUCT1 MOD 10
1140 CARRY1=PRODUCT1\10
1150 NEXT I
1160 IF CARRY1>0 THEN DIGIT1=DIGIT1+1:NUM1(DIGIT1)=CARRY1
1170 CARRY2=0
1180 DIGIT2=DIGIT
1190 FOR I=1 TO DIGIT2
1200  PRODUCT2=NUM(I)*M2+CARRY2
1210 NUM2(I+1)=PRODUCT2 MOD 10
1220 CARRY2=PRODUCT2\10
1230 NEXT I
1240 IF CARRY2>0 THEN DIGIT2=DIGIT2+1:NUM2(DIGIT2+1)=CARRY2
1250 CARRY=0
1260 IF DIGIT1 >(DIGIT2+1) THEN DIGIT=DIGIT1 ELSE DIGIT=DIGIT2+1
1270 FOR I=1 TO DIGIT
1280 PRODUCT=NUM1(I)+NUM2(I)+CARRY
1290 NUM(I)=PRODUCT MOD 10
1300 CARRY=PRODUCT \10
1310 NEXT I
1320 IF CARRY>0 THEN DIGIT=DIGIT+1:NUM(DIGIT)=CARRY
1330 DIGIT1=DIGIT:DIGIT2=DIGIT
1340 NEXT E
1350 PRINT M;"^";P;"=";
1360 FOR I=DIGIT TO 1 STEP -1
1370 PRINT USING"#";NUM(I);
1380 NEXT I
```

```
RUN
請輸入底數 ? 99
請輸入指數 ? 99
 99 ^ 99 =369729637649726772657187905628805440595668764281741102430259972423552570455277523421410650010128232727940978889548326540119429996769494359451621570193644014418071060667659301384999779999159200499899
```

(3)若底數為三位數的次方問題，只要程式多設一個最內層廻路即可。以此類推，底數為四位數、五位數的次方問題，皆可用程式來處理。

2. 如何計算 n!

(1)首先從 n 是個位數著手，與計算 n^p相同，只是 PRODUCT＝NUM(I)×E 不同而已（E 為 1 至 n 之變數）。程式如下：（程式三）

```
10 REM computer 9 !
20 INPUT "請輸入一位數之值，以計算其階乘 ";N
30 DIM NUM(10)
40 NUM(1)=1 :DIGIT=1
50 FOR E=1 TO N
```

```
60 CARRY=0
70 FOR I=1 TO DIGIT
80 PRODUCT=NUM(I)*E+CARRY
90 NUM(I)=PRODUCT MOD 10
100 CARRY=PRODUCT\10
110 NEXT I
120 IF CARRY>0 THEN NUM(DIGIT+1)=CARRY:DIGIT=DIGIT+1
130 NEXT E
140 PRINT N;"!=";
150 FOR I=DIGIT TO 1 STEP -1
160 PRINT USING "#";NUM(I);
170 NEXT I
```

```
RUN
請輸入一位數之值,以計算其階乘 ? 9
 9 !=362880
Ok
```

(2)若 n 為二位數，程式吾人可用

```
FOR E=1 TO n
  處理E之個位數與(E-1)!相乘
  處理E之十位數與(E-1)!相乘
  處理上述二內迴路相加。
NEXT E
```

方法。

但若用以下分段方法來處理，亦即先算9!，然後再進行10!以後之計算。

```
FOR E=1 TO 9
  處理9!
NEXT E
FOR E=1 TO n
  處理E之位數與(E-1)!相乘
  處理E之十位數與(E-1)!相乘
  處理上述二內迴路相加。
NEXT E
```

如此之方法執行起來速度較快，程式如下：(程式四)

```
1000 REM computer   99!
1010 INPUT "請輸入二位數之值,以計算其階乘";N
1020 M=1000
1030 DIM NUM(M):DIM NUM1(M):DIM NUM2(M):DIM NUM3(M)
1040 NUM(1)=1
1050 DIGIT=1
```

```
1060 FOR E=1 TO 9
1070 CARRY=0
1080 FOR I=1 TO DIGIT
1090 PRODUCT=NUM(I)*E+CARRY
1100 NUM(I)=PRODUCT MOD 10
1110 CARRY=PRODUCT \10
1120 NEXT I
1130 IF CARRY>0 THEN NUM(DIGIT+1)=CARRY:DIGIT=DIGIT+1
1140 NEXT E
1150 FOR E=10 TO N
1160 R1=E MOD 10
1170 CARRY1=0
1180 DIGIT1=DIGIT
1190 FOR I=1 TO DIGIT1
1200 PRODUCT1=NUM(I)*R1+CARRY1
1210 NUM1(I)=PRODUCT1 MOD 10
1220 CARRY1=PRODUCT1\ 10
1230 NEXT I
1240 IF CARRY1>0 THEN NUM1(DIGIT1+1)=CARRY1:DIGIT1=DIGIT1+1
1250 R2=E \ 10
1260 CARRY2=0
1270 DIGIT2=DIGIT
1280 FOR I=1 TO DIGIT2
1290 PRODUCT2=NUM(I)*R2+CARRY2
1300 NUM2(I+1)=PRODUCT2 MOD 10
1310 CARRY2=PRODUCT2\10
1320 NEXT I
1330 IF CARRY2>0 THEN NUM2(DIGIT2+2)=CARRY2:DIGIT2=DIGIT2+1
1340 IF DIGIT1 > (DIGIT2+1) THEN DIGIT=DIGIT1 ELSE DIGIT=DIGIT2+1
1350 CARRY=0
1360 FOR I=1 TO DIGIT+1
1370 PRODUCT=NUM1(I)+NUM2(I)+CARRY
1380 NUM(I)=PRODUCT MOD 10
1390 CARRY=PRODUCT \10
1400 NEXT I
1410 IF CARRY>0 THEN NUM(DIGIT+1)=CARRY:DIGIT=DIGIT+1
1420 DIGIT1=DIGIT:DIGIT2=DIGIT
1430 NEXT E
1440 PRINT N;"!=";
1450 FOR I=DIGIT TO 1 STEP -1
1460 PRINT USING"#";NUM(I);
1470 NEXT I
```

```
Ok
RUN
請輸入二位數之值,以計算其階乘? 10
 10 !=3628800

RUN
請輸入二位數之值,以計算其階乘? 99
 99 !=30910495574036078847621540255898344159875697788332976262378430147315226745
61961594787510698224659750141906631182964009627372109168640000000000000000000000
Ok
```

(3)若利用此方法可拓展至能計算三位數的階乘。方法是將程式(五)後面再加如下的迴路，四位數階乘亦依此類推。

```
┌FOR  E = 100  TO  n
│  [處理E之個位數乘以（E－1）！
│  [處理E之十位數乘以（E－1）！
│  [處理E之百位數乘以（E－1）！
│  [將以上三內迴路相加
└NEXT  E
```

(三)驗算

甲．以2的1000 次方爲例

(1)由 2^{1000} 的位數來驗證

$\because \log 2^{1000} = 1000 \log 2 = 301.02 \cdots\cdots$

由此可知 2^{1000} 有 302位數

(2)由 2^{1000} 的末兩位來驗證（周雲雄，民72年）

吾人可用筆算出 2^1，2^2，2^3，2^4，……觀察其末兩位的分佈情形以尋找規律性，分別是02，04，08，16，32，64，28，56，12，24，48，96，92，84，68，36，72，44，88，76，52，04，08，……

其中 2^2，2^{22}，2^{42}之末兩位均出現04

由此可得 $2^2 \equiv 2^{22} \equiv 2^{42} \equiv \cdots\cdots \pmod{100}$

再推廣成 $2^n \equiv 2^{n+20k} \pmod{100}$，其中 $n \geq 2$，$k \geq 1$

令 $n = 20$，$k = 49$，吾人可得 $2^{1000} \equiv 2^{20} \equiv 76 \pmod{100}$

亦即 2^{1000} 的末兩位是76

(3)由 2^{1000} 的前二位數來驗證（周雲雄，民72年）

$\log 2 = 0.30102$（取前五位小數）

若考慮誤差 $0.30101 < \log 2 < 0.30103$

$$301.01 < 1000 \log 2 < 301.03$$

上式表示 2^{1000} 有 302位數……(1)

而 $\log \dfrac{2^{1000}}{10^{301}} = \log 2^{1000} - 301$

$\therefore \quad 0.01 < \log \dfrac{2^{1000}}{10^{301}} < 0.03$

又 $\log 1.1 = 0.04139 > 0.03$ （由計算機求得）

$\therefore \quad \log 1.0 < \log \dfrac{2^{1000}}{10^{301}} < \log 1.1$

$\Rightarrow \quad 1 < \dfrac{2^{1000}}{10^{301}} < 1.1$ ……(2)

由(1)，(2)二式可知 2^{1000} 的最高二位數是10。

以上三種驗算所得結果均與程式(一)之執行結果相符。

乙. 以99!爲例

(1)由99!末尾連續多少個0來驗證。

令 $99! = 2^{h_1} \cdot 5^{h_2} \cdot k$，其中k與2，5均互質。

$\because \left[\dfrac{99}{2}\right] = 49$，$\left[\dfrac{49}{2}\right] = 24$，$\left[\dfrac{24}{2}\right] = 12$，$\left[\dfrac{12}{2}\right] = 6$，

$\left[\dfrac{6}{2}\right] = 3$，$\left[\dfrac{3}{2}\right] = 1$

$\therefore h_1 = 49 + 24 + 12 + 6 + 3 + 1 = 95$

$\because \left[\dfrac{99}{5}\right] = 19$，$\left[\dfrac{19}{5}\right] = 3$，

$\therefore h_2 = 19 + 3 = 22$

$99! = 2^{95} \cdot 5^{22} \cdot k$，$2 \nmid k$，$5 \nmid k$。

$\therefore$ 99!尾部有22個“ 0 ”。

(2)由99!尾部最後一個不是“ 0 ”的數來驗證。

$\because 10! = 362880$

$20! = 362880 \times 11 \times 12 \times \cdots\cdots \times 20$

而 $11 \times 12 \times 13 \times \cdots\cdots \times 20$，其尾部最後一個不是零的數也會是8。

$\therefore$ 20!其尾部最後一個不是零的數是 8×8 的個位數4

由此推論100! 其尾部最後一個不是零的數是 8^{10} 的個位數4。

顯然地，99!其尾部最後一個不是零的數也是4。

以上二種驗算所得結果均與程式(四)之執行結果相符。

三、進一步探討——除法有效位數的需求

除法是乘法的逆運算，除法分段計算的技巧是將乘法對加法分配律的反向計算。

若$A = BQ_0 + R_1 \quad 0 < R_1 < B \quad Q_0 \in I$

令
$$\begin{cases} 10R_1 = BQ_1 + R_2 \\ 10R_2 = BQ_2 + R_3 \\ 10R_3 = BQ_3 + R_4 \end{cases}$$
而Q_1，Q_2，Q_3……$\in \{0, 1, 2, \cdots\cdots, 9\}$

R_1，R_2，R_3，……均小於B，

$$\Rightarrow \begin{cases} R_1 = B \times Q_1 \times 10^{-1} + R_2 \times 10^{-1} \\ R_2 = B \times Q_2 \times 10^{-1} + R_3 \times 10^{-1} \\ R_3 = B \times Q_3 \times 10^{-1} + R_4 \times 10^{-1} \end{cases}$$

則
$$\begin{aligned} A &= BQ_0 + R_1 \\ &= BQ_0 + B \times Q_1 \times 10^{-1} + R_2 \times 10^{-1} \\ &= B(Q_0 + Q_1 \times 10^{-1}) + R_2 \times 10^{-1} \\ &= B(Q_0 + Q_1 \times 10^{-1}) + B \times Q_2 \times 10^{-2} + R_3 \times 10^{-2} \\ &= B(Q_0 + Q_1 \times 10^{-1} + Q_2 \times 10^{-2}) + (B \times Q_3 \times 10^{-1} + R_4 \times 10^{-1}) \\ &\quad \times 10^{-2} \\ &= B(Q_0 + Q_1 \times 10^{-1} + Q_2 \times 10^{-2} + Q_3 \times 10^{-3}) + R_4 \times 10^{-3} \\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ &= B(Q_0 + Q_1 \times 10^{-1} + Q_2 \times 10^{-2} + Q_3 \times 10^{-3} + \cdots\cdots) \end{aligned}$$

$\because Q_0 \in I$，Q_1，Q_2，Q_3，……$\in \{0, 1, 2, \cdots\cdots, 9\}$

$\therefore A / B = Q_0 . Q_1 Q_2 Q_3 \cdots\cdots$

依上述方法每次將餘數乘以10再除以除數，每次所得之商（Q_0，Q_1，Q_2，Q_3，……）依次排出即可將有效位數無限制增加。

程式與例子：（程式五）

程式：

```
10 INPUT "請輸入分子";N
20 INPUT "請輸入分母";D
30 INPUT "請問計算至小數點以下第幾位";DIGIT
40 PRINT N;"/";D;"=";
50 PRINT USING "###";QUOTIENT;:PRINT".";
60 DIM NUM(DIGIT+1)
70 RESIDE=N MOD D
80 QUOTIENT=N \ D
90 FOR I=1 TO DIGIT
```

```
100 N=RESIDE*10
110 NUM(I)=N\D
120 RESIDE=N MOD D
130 NEXT I
150 FOR I=1 TO DIGIT
160 PRINT USING "#";NUM(I);
170 NEXT I
```

例子：

```
Ok
RUN
請輸入分子? 1
請輸入分母? 17
請問計算至小數點以下第幾位? 32
 1 / 17 =   0.05882352941176470588235294117647
Ok
RUN
請輸入分子? 3
請輸入分母? 23
請問計算至小數點以下第幾位? 22
 3 / 23 =   0.1304347826086956521739
Ok
```

四、結　　論

(一)以上幾種驗算方法所得結果均與程式執行結果相符，雖是如此，尚不足以充分說明程式一定萬無一失，但它却是檢驗程式正確的必要條件，本文提出此等數學方法驗算目的是讓讀者對於程式的正確性較爲放心。

(二)不可諱言，這些程式執行所花費的時間較多，對於大的程式與爭取時效的工作是有缺陷，但是對觀念的剖析與程式的處理，實爲野人獻曝，當然更希望能抛磚引玉，讓先進與朋友提供更好的作品與大家共享。

五、參考資料

1. 周雲雄（民72）；關於 2^{1000} 數學傳播第七卷第三期，中央研究院數學研究所發行。