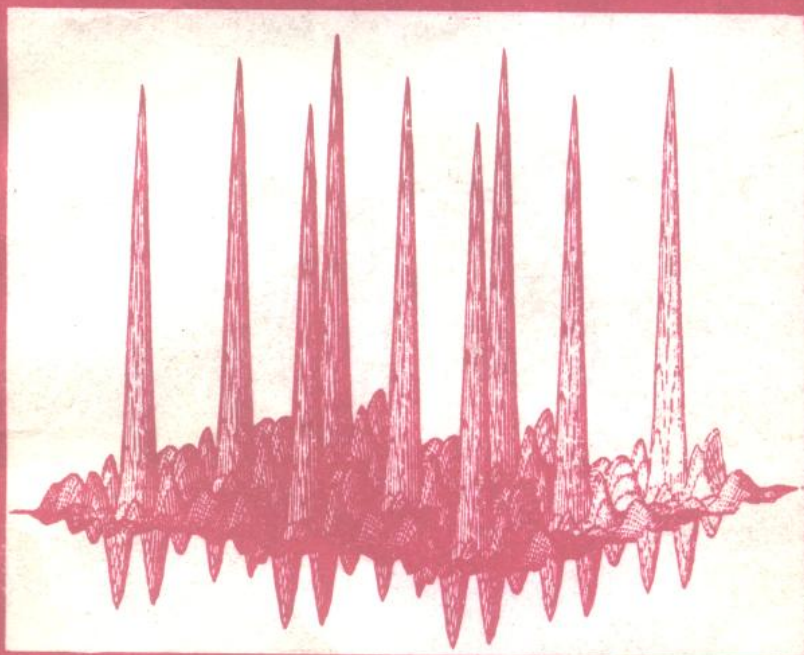


電腦資訊結構

林烟桂 編著



松崗電腦圖書資料有限公司

電腦資訊結構

林烟桂 編著

松崗電腦圖書資料有限公司 印行

電腦資訊結構

所有權



定價新台幣 160 元 整

書號：510106

編著者：林 炯 桂

發行人：吳 守 信

發行所：道明出版社

台北市仁愛路二段一一〇號三樓

總經銷：松崗電腦圖書資料有限公司

台北市仁愛路二段一一〇號三樓

電話：3930255 · 3930249

郵政劃撥：109030號

承印者：泉崗印刷設計股份有限公司

台北市仁愛路二段一一〇號三樓

電話：3930255 · 3930249

中華民國六十六年一月 初版

中華民國六十七年十月 再版

中華民國七十年十月 三版

中華民國七十二年六月 四版

本出版社經行政院新聞局核准登記，

登記證為局版台業字第一七二九號。

自序

這幾年來，政府大力提倡工業升級，特別注重資訊的研究。由此而帶動個人用電腦的發展，各界對於電腦軟體的需求，更上層樓。電腦資訊結構的研究也就應運而生。

本書所使用的程式，曾經用過 PL/1，及 COBOL。而如今改用 BASIC，主要是配合科技的發展。TI 990 BASIC 是美國德州儀器公司所發展出來的產品，稍許具備結構化的味道，例如 IF... THEN... ELSE... 敘述，FOR... NEXT 敘述，GO-SUB 敘述，CALL 敘述等，多部終端機同時執行程式時，反應時間相當快，程式書寫簡單容易，適合教學之用。且個人用電腦主要也採用 BASIC，具有普遍化的潛力，因此本書改採用 BASIC。

第一章說明 BASIC 之文法及資料結構之觀念，讀者若希望進一步了解 BASIC，請查閱 TI 990 BASIC REFERENCE MANUAL。

第二章說明疊列，佇列，雙佇列，等的用法。第三章介紹樹的結構，語律分析，陳式樹等。第四章搜尋。第五章分類。第六章計劃評核術。第七章檔。其中 BASIC 文法，語律分析，陳式樹，檔等均是本版所加入。例題和習題儘量增多，程式儘可能寫成結構化，因此不必要的流程圖均捨去。

本書之能夠出版，要感謝私立正修工專提供 TI 990 電腦及各種方便，龔瑞璋先生提供部份資料。本書編印匆促，錯誤在所難免，希高明指正。不勝感激。

林炳桂

民國七十二年五月

高雄・正修工專

電腦資訊結構

目 錄

第一章 簡 介

1.1	資訊結構簡介	1
1.2	程式語言 BASIC	5
1.2.1	輸入／輸出敘述	6
1.2.2	指定敘述	9
1.2.3	控制敘述	9
1.2.4	函數敘述	13

第二章 線性序列

2.1	疊序列、佇列、雙佇列	17
2.2	順序排列	20
2.3	鏈排列	30
2.3.1	疊序列	33
2.3.2	佇列及雙佇列	35
2.3.3	環鏈序列	38
2.3.4	雙鏈序列	41
2.4	陣 列	60
2.4.1	順序排列	60
2.4.2	鏈 排 列	62

第三章 樹

3.1	簡 介	79
3.1.1	二進樹	81
3.2	經歷一棵樹	81
3.2.1	經歷二進樹	84
3.3	樹的轉換	86
3.3.1	一般樹轉換成 Knuth 二進樹	86
3.3.2	一般樹轉換成純二進樹	87
3.4	樹的電腦表示法	89
3.4.1	雙鏈法表示二進樹	89
3.4.2	雙鏈標記法表示一般樹	101
3.5	路逕長度	104
3.5.1	二進樹的路逕長度	106
3.5.2	二進樹加權路逕長度	109
3.6	語律分析	112
3.7	陳式樹	118

第四章 搜 尋

4.1	簡 介	131
4.2	掃描法	131
4.2.1	直接掃描	132
4.2.2	控制掃描	132
4.3	鍵轉換法	142
4.3.1	重轉換法	143
4.3.2	鏈法	151

第五章 分 類

5.1	簡 介	159
5.2	內分類	159
5.2.1	計數分類	159
5.2.2	加入分類	163
5.2.2.1	直接加入法	163
5.2.2.2	薛爾法	164
5.2.2.3	序列法	166
5.2.3	互換分類	168
5.2.3.1	泡昇法	168
5.2.3.2	分隔法	171
5.2.4	選擇分類	175
5.2.4.1	直接選擇	175
5.2.4.2	樹選擇法	176
5.2.5	合併分類	185
5.3	外分類	188
5.3.1	多元合併與取代選擇	189
5.3.2	多相合併分類	200
5.4	分類法之比較	210

第六章 計劃評核術

6.1	前言	213
6.2	計算方法	214
6.3	資訊結構	216
6.3.1	節編號轉換為節指標	217
6.3.2	前進順序佇列	219
6.3.3	後退順序佇列	221

6.3.4	作業資訊結構.....	223
6.3.5	節的資訊結構.....	224
6.3.6	在前節佇列前面加入一個節.....	226
6.3.7	在後節佇列前面加入一個節.....	227
6.4	應用.....	229
6.5	程式設計步驟.....	231

第七章 檔

7.1	檔	241
7.2	檔之儲存及取還	242
7.3	隨機檔	244
7.4	索引順序檔	248
7.5	倒裝檔	250

附 錄

A.	懷柏南西係數表.....	253
B.	「 $\log_2 N$ 」及 N 之對照表.....	253
C.	習題之題示.....	254
D.	名詞對照表.....	276
E.	參考資料一覽表	280

第一章 簡 介

1.1 資訊結構簡介

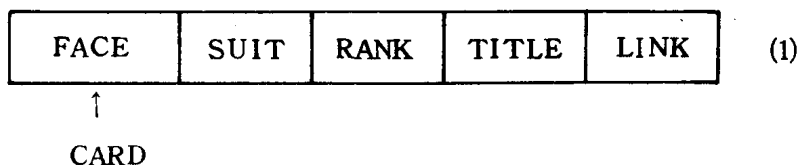
電腦程式通常處理資訊。絕大多數的資訊 (information) 不僅僅包含一些數值，也包含一些文字及各數據元素間的結構關係。最簡單的資訊，各元素之間呈線性序列 (linear list)，這種簡單的結構說明了：在序列 (list) 中那個元素最先、那個最後、那個在那個之前、那個在那個之後。複雜一點的資訊，像二元陣列 (two-dimensional array)， n 元陣列 (n -dimensional array)，樹 (tree) 等說明了各代或各分支的關係。更複雜的資訊像多鏈結構 (multilinked structure)，有着錯綜複雜的關係。

爲了正確使用電腦，了解資訊的結構關係是很重要的。不僅要了解資訊的結構關係，也要了解資訊表示的技巧及其運用的方法。本書所要探討的是關於資訊結構：各種不同的結構的靜態及動態特性，以及結構的建立，資訊的取得、更改、消除等各種有效的演算法。

我們要討論的大部份資訊常常被稱爲「序列處理」，因爲很多的程式設計系統是針對着序列而設計的。雖然序列處理很有用，但對程式設計師而言，却是一種不必要的束縛。他可以運用本書的方法於其程式中，而得到完美的結果。不幸的是人們現在乃覺得序列處理的技巧太困難了，而寧願使用別人已經設計完成的

次常式 (subroutine) 。我們將會看到對於複雜結構的處理一點也不神秘、一點也不困難。這種技巧對程式設計師特別重要，因為他可以使用這種技巧於組合語言 (ASSEMBLER)，高級程式語言像式譯語言 (FORTRAN)，商業通用語言 (COBOL)，或程式語言 PASCAL 。

這裡要說明往後用到的名詞及符號。資訊或稱為檔 (file)，包含許多組的記錄 (record)。每一組記錄是由許多欄 (field) 組成的。例如玩撲克牌，假定每一組記錄記載每一張牌的資料，如牌面朝上或朝下，什麼牌組，幾點的牌，下一張牌是那一張，說明等。

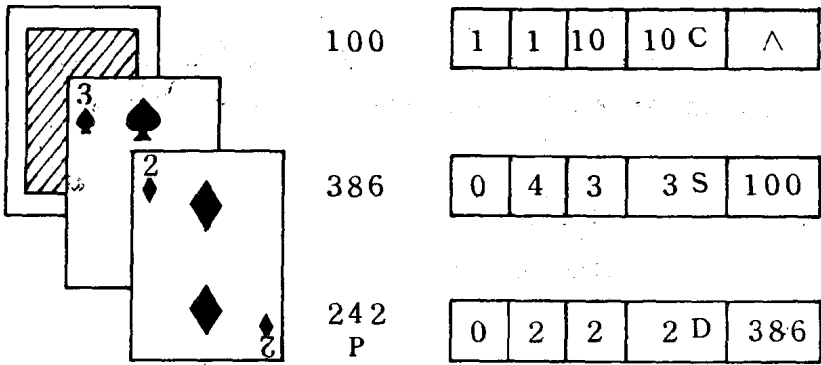


該記錄 CARD 包含五欄。各自為 FACE、SUIT、RANK、TITLE、LINK。記錄的地址稱為鏈 (link) 是該記錄開始第一個字的位置。地址常常從某一個基點算起，稱為相對地址。也可以從主儲存體第一個位置算起，稱為絕對地址。

每一欄的內含可能是數字、文字、鏈或其他的資訊。拿撲克牌的例子來講，假定 FACE = 1 表示牌面朝下，FACE = 0 表示牌面朝上。SUIT = 1, 2, 3 或 4 各自代表梅花、方塊、紅心或黑桃。RANK = 1, 2, …… , 13 代表 A, 2, …… , K。LINK 是表示鏈。指出下一張牌的位置。TITLE 是該牌的說明。

一組牌可能是這樣：

地 址 記 錄



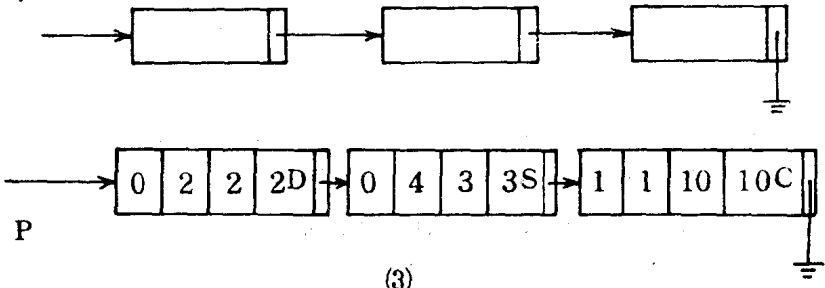
實際牌組

電 腦 表 示 法

(2)

這三張牌在電腦主儲存體的地址各為 100、386，及 242。這些地址可以是其他的數字，只要牌的前後順序關係不變就行。注意在地址 100 的記錄裏有一特殊的鏈 ^，表示沒有鏈，也就是說該地址 100 的記錄已經是最後一個。沒有鏈在電腦裏常以 0 表示。

鏈至資訊的其他元素，這種觀念在電腦的程式設計裏非常地重要，它是複雜結構表示法的關鍵。記錄的前後關係常以箭頭表示。



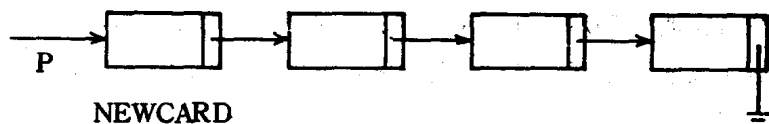
(3)

實際的地址 242, 386, 及 100 不再出現了。沒有鏈以電路中接地的符號表示。(3)中加入一 P 表示鏈變數或指標變數 (pointer variable)。

另一項重要的表示方法是關於記錄裡的欄，如何指定該欄。可以在變數後頭的括弧內填入鏈變數即可。例如：

```
TITLE$(P) = "2D"
SUIT(P)   = 2
LINK(P)   = P
RANK(P)   = 3
```

讀者應該仔細地觀察上述幾個例子，因為在以後的演算中會利用到這些表示法。為了使這些觀念更清楚起見，我們拿一個簡單的例子來說明，置一張新牌於這組牌上如何？假定 NEWCARD 是鏈變數，表示這張新牌的地址。則演算如下：



```
10 LINK(NEWCARD) = P
20 P = NEWCARD
30 FACE(P) = 0
```

10 說明新牌的 LINK 欄置入鏈變數 P，使得這張新牌可以鏈至原來的第一張牌 ♦ 2，20 說明新的頂張為 NEW - CARD，30 說明這張新牌的內含（牌面朝上）。

下面的演算法計算出這一組牌的張數：

```

10 N=0
20 IF P<>C THEN GOSUB 50 :: GOTO 20
   ...
50 N=N+1
60 P=LINK(P)
70 RETURN

```

注意上述演算中使用兩種名稱。一種是變數的名稱如 P, NEWCARD, N。另一種是欄的名稱如 FACE, LINK。這兩種性質的名稱千萬混淆不得。若 F 是欄的名稱, L 是鏈的名稱, 則 F (L) 是一個變數, 但 F 本身並不是一個變數, 因為它並不表示某一個值。若與非 0 的鏈連起來使用, 它就代表該地址的內容, 也就構成一個變數。

1.2 程式語言 BASIC

BASIC 是由 Beginner's All-purpose Symbolic Instruction Code 的第一個字母組成。顧名思義, 它是為初學程式語言的人所設計的, 是學習其他程式語言的基礎。

每一行表示一個敘述, 每一行包含一個敘述號碼, 一至五位, 以及敘述本體。例如

```
10 READ A
```

10 為敘述號碼, READ A 為敘述本體。

常數有整數常數, 實數常數及文字常數。例如

12, +130, 0, -123 俱為整數

13.56, -2.38, 1.2E+2, 1.5E-3 俱為實數

"X=", "FINAL MATRIX" 俱為文字常數

變數分為數值變數及文字變數兩種。數值變數的第一個字為英文字, 其後為英文字或數字, 文字變數最後一個字必須為 \$。例如

X, MASS, Q123 俱為數值變數

X\$, NAME\$, P5\$ 俱為文字變數

一般而言，BASIC的敘述可分為幾個大類：(1)輸入／輸出敘述，(2)指定敘述，(3)控制敘述，(4)函數敘述。今就最常用的敘述加於說明。

1.2.1 輸入/輸出敘述

```
10 DATA 3, -8
```

```
20 READ A, B
```

READ敘述由DATA取得資料，A的內含為3，B的內含為-8。另外一種敘述INPUT可由控制台或終端機鍵入資料。如

```
40 INPUT X, Y, Z, CH$
```

若從終端機鍵入13, 25, -4, ABCDE 那麼X的內含為13，Y的內含為25，Z的內含為-4，CH\$的內含為ABCDE。變數若要當為陣列，若不宜告，則BASIC把它看做11個元素的變數。例如A不宜告，則認為A(10)，共11個變數，各為A(0)，A(1)，A(2)，…，A(10)。

```
70 PRINT "X=" ; X, "Y+Z" ; Y+Z
```

印出的結果為X = 13，Y + Z = 21。變數或常數之間的分隔符號若用逗號，則其印出間隔為每一變數或常數各佔16個位置。若以分號；隔開，每個變數或常數緊接着印下去。若要依某一特定格式印在某一特定的位置，則要用USING說明了。

```
80 VALUE = 1234.569
```

```
90 FORMAT$ = "####.##"
```

```
100 PRINT USING "####.##" : VALUE
```

```
110 PRINT USING FORMAT$ : VALUE
```

敘述 100 及 110 印出的結果相同 1234.57 小數第三位，自動四捨五入，# 稱為格式控制字，為實際的數字所取代。

若不從鍵盤，而從其他的設備或檔輸入資料，那麼該設備或該檔必先給予編號，OPEN 以後才能 INPUT，處理完最後必須 CLOSE。例如資料從讀卡機輸入，讀卡機之系統代號為 CR01，而檔號可任意設定，例如 #15，即第 15 號檔的意思，#15 稱為邏輯檔代號，CR01 稱為系統設備代號，使用 OPEN 把它們兩個互相連貫起來。如下例。其中 #15 只在本程式有定義，在其他程式中可能沒有定義，或其他的定義，而 CR01 在任何程式中均是同一定義。

```
110 OPEN #15: "CR01", INPUT
    ...
210 INPUT #15: A, B, C
    ...
910 CLOSE #15
```

同理，資料要輸出的設備不是螢幕時，要定義一個檔號給它，使用它以前要先 OPEN，當 PRINT 完後，最後要加於 CLOSE。列表機在系統中代號為 LP01，至於檔號可任意指定。例如

```
120 OPEN #3: "LP01"
    ...
220 PRINT #3: A, B, C
    ...
920 CLOSE #3
```

同理，磁碟檔寫出檔名稱即可。例如

```
130 OPEN #8: "D5.LP100", APPEND
    :
230 PRINT #8: A, B, C
    :
930 CLOSE #8
```

在OPEN之後，通常用來說明該檔性質的文字有 INPUT, OUTPUT, UPDATE 及 APPEND。INPUT 表示該檔使用為輸入，OUTPUT 使用為輸出，UPDATE 使用為更新，即可同時為輸入及輸出，APPEND 表示輸出至該檔最後面，即加入原檔之後。

輸出至某檔，而要依照某種格式時，檔號必須置於 USING 之前，並且檔號後之冒號省略。例如

```
320 IMAGE ##. # ### ##. ##
330 PRINT #3 USING 320:A,B,C
340 PRINT #3 USING "X=##. #":X
```

要控制列表機跳頁時可使用 CHR\$ 函數，該函數專門用來表示不能用文字表示的 ASCII 字碼，例如跳頁的符號為 FF (即 Form Feed 的意思)，ASCII 碼的十進位值為 12，那麼 CHR\$(12) 即表示 FF。例如

```
350 PRINT #3:CHR$(12)
```

若要空一行，則下列兩種表示法均可。

```
360 PRINT #3:" "
370 PRINT #3
```

常用的設備名稱，系統代號，以及磁碟檔如下表

設備名稱或檔	系統代號	說 明
讀卡機	CR 01	
磁帶機	MT 01	
列表機	LP 01	
磁碟機	DS 01	
磁碟檔	.d.f.	d 表 DIRECTORY 名稱。 f 表檔名。由設計人員命名。

1.2.2 指定敘述

指定敘述有兩種，一種是算術指定敘述，一種是文字指定敘述。指定敘述的格式是等號左邊一個變數，等號右邊一個式子，變數左邊可以加上LET這個字，也可以不加。例如

```

100      Y = X * X * X - 3 * X + 1
110 LET XL = 0
120      A$ = " STRING A "
130 LET B$ = " STRING B "
140 LET C$ = A$ & B$

```

等號右邊的式子是由常數，變數及運算符號所組成，運算符號有加減乘除及乘方。文字的運算符號有聯合（&）。

加	+
減	-
乘	*
除	/
乘方	^
聯合	&

其運算的順序，有括號先算括號內的式子，否則先算乘方，再算乘除，後算加減。與算術四則計算的規則相同。

1.2.3 控制敘述

控制電算機執行敘述的順序稱為控制敘述，最簡單的一種稱為無條件移轉。

```

200 GOTO 360

```