

臺北市立第一女子高級中學 98 學年度  
資訊學科能力競賽初賽

Reading & Question



June 6, 2009

測驗時間：2009 年 6 月 6 日上午 08:10—09:00

## PART I 數字系統及資料表示法

---

### Reading

在電腦的世界裡，所有的資料都是由 0 和 1 組成，平日我們可以看得到的畫面、使用的工具程式、瀏覽的網頁等等等的，背後都是有著複雜而大量的運算，才得以呈現如此的面貌。電腦裡最基礎的一種資料，就是數值資料。有數值資料，就有處理數值資料表示的數字系統。我們生活中最常見的一種數字系統是 10 進位，但對電腦來說，是由 0 和 1 表示的 2 進位法。在一般的文件中，如果我們特別標明一個數字是  $n$  進位制，我們會用如下的方法表示：

$$(\text{數字})_n$$

舉例來說， $(11)_{10}$  代表的是十進位制的 11， $(11)_2$  則代表二進位制的 11，換算成 10 進位數字就是 3。

如果我們要把 10 進位轉成 2 進位，例如把  $(14)_{10}$  換成二進位表示，我們可以使用短除法取餘數來得到答案：

$$\begin{array}{r} 2 \overline{) 14} \quad \dots 0 \\ 2 \overline{) 7} \quad \dots 1 \\ 2 \overline{) 3} \quad \dots 1 \\ 1 \end{array}$$

把餘數從下往上寫，得到的就是 2 進位表示法，以這個例子來說， $(14)_{10}$  用 2 進位表示就是  $(1110)_2$ 。若要將 10 進位制轉其他進位制，也可以使用此方法。

如果要把各個進位制數字轉回 10 進位，可以使用以下的方法(以剛剛的例子作示範)：

$$\begin{array}{cccc} \underline{1} & \underline{1} & \underline{1} & \underline{0} \\ 2^3 & 2^2 & 2^1 & 2^0 \end{array} \quad \rightarrow \quad 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 14$$

(備註：仔細觀察，不難發現可以如此轉換的原因。)

因為電腦裡面只有 0 跟 1，那如果我們想要表示負數該怎麼辦？為了解決這個問題，「補數法」誕生了。對於一個採用  $n$  進位制的系統來說，它可以使用的補數法有  $n$  的補數法以及  $n-1$  的補數法，前者是後者再加上 1 的結果。現行的電腦表示負數採用 2 的補數法，好處是可以去除 +0 和 -0 的差別。

一個十進位負數以 2 的補數法來表示，要經過以下過程：

1. 將其絕對值換為 2 進位
2. 得知此系統以幾個 bytes 來儲存此數字
3. 取其 1 的補數(即 0 與 1 互換)
4. 將取 1 補數後的值+1

這樣得到的數值就是該負數的 2 的補數表示法，一般我們會用第一個 bit 來作為正負數的判定依據，因為負數的第一個 bit 一定為 1。

若我們要將 -14 以 1 個 byte 的 2 補數法來表示，則過程如下：

$$14 \rightarrow \underline{00001110} \rightarrow \underline{11110001} \rightarrow 11110010$$

2 進位表示      1 的補數      2 的補數

採用 2 的補數法有一個特點：對已經表示成 2 的補數法之數字再取一次 2 的補數，則可得到其絕對值的 2 進位數字。

## Questions

1. 請將  $(3824)_{10}$  換成 7 進位表示。A: (            )<sub>7</sub>
2. 請將  $(1021)_3$  換回 10 進位表示。A: (            )<sub>10</sub>
3. 請將  $(-748)$  以 2 個 bytes 的 2 補數法來表示。A: \_\_\_\_\_
4. 請將 2 的補數法表示的  $(10111011)_2$  換回 10 進位數字。A: \_\_\_\_\_

## PART II 布林代數運算

---

### Reading

布林代數運算是邏輯中常用的一種運算。我們定義了如下的規則：

- 對於一個變數  $x$ ，其值只可能為 1(True)，或者是 0(False)。
- 我們用  $x'$  來表示  $x$  的相反值。

布林代數的基本運算如下：

- 兩個布林代數做「+」運算：兩個布林代數做 OR 的運算，兩數只要有其中一個為 True，此布林代數式就為 True。
- 兩個布林代數做「·」運算：兩個布林代數做 AND 運算，兩數只要有其中一個為 False，此布林代數式就為 False。

在這樣的基本運算之下，我們有了一些基本的定理：

- $X \cdot 0 = 0$
- $X \cdot 1 = X$
- $X \cdot X = X$
- $X \cdot X' = 0$
- $X + 0 = X$
- $X + 1 = 1$
- $X + X = X$
- $X + X' = 1$

在這樣的基本定理之下，我們有了一些基本的定律：

|   |   |
|---|---|
| <b>(1) 交換律：</b><br>a. $X+Y = Y+X$<br>b. $X \cdot Y = Y \cdot X$   | <b>(2) 結合律：</b><br>a. $(X+Y)+Z = X+(Y+Z)$<br>b. $(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$ |
| <b>(3) 分配律：</b><br>a. $X \cdot (Y+Z) = XY+XZ$<br>b. $X+(Y \cdot Z) = (X+Y) \cdot (X+Z)$<br>c. $(W+X) \cdot (Y+Z) = WY+WZ+XY+XZ$ | <b>(4) 吸收律：</b><br>a. $X+XY = X$<br>b. $X+X'Y = X+Y$  |

因為有這些定理和定律，所以我們常常可以化簡布林代數運算式，舉例來說：

$$\text{化簡 } Y = A + A' + B$$

$$\begin{aligned} Y &= A + A' + B \\ &= (A + A') + B && \text{(運用結合律)} \\ &= 1 + B && (\because A + A' = 1) \\ &= 1 \end{aligned}$$

## Questions

1. 請化簡  $Y = AB'C + A'B'C + BC$
2. 請化簡  $Y = ABC + AB'C + ABC' + AB'C'$
3. 請化簡  $Y = A(B+C)A' + D$
4. 請化簡  $Y = AB + A'B$
5. 真值表(Truth table)是最常見的邏輯運算結果表示形式，以下提供 AND( $\wedge$ ), OR( $\vee$ ), XOR( $\oplus$ )與 NOT( $\sim$ )四種布林運算，其中 P、Q 為布林變數，T 表示 True，而 F 表示 False。

| P | Q | $\sim P$ | $P \wedge Q$ | $P \vee Q$ | $P \oplus Q$ |
|---|---|----------|--------------|------------|--------------|
| T | T | F        | T            | T          | F            |
| T | F | F        | F            | T          | T            |
| F | T | T        | F            | T          | T            |
| F | F | T        | F            | F          | F            |

假設 X、Y、Z 為三個布林變數，下面 A、B、C、D 四個布林運算式中，哪一項與其它項不等值（具有不同的真值結果）？A: \_\_\_\_\_

- (A)  $Y \vee (X \oplus Y \oplus Z)$
- (B)  $Y \vee (X \wedge \sim Z) \vee (\sim X \wedge Z)$
- (C)  $(X \oplus Z) \vee (Y \wedge \sim Z) \vee (Y \wedge Z)$
- (D)  $(X \oplus \sim Y) \vee (Y \oplus \sim Z)$

## PART III 網路

---

### Reading

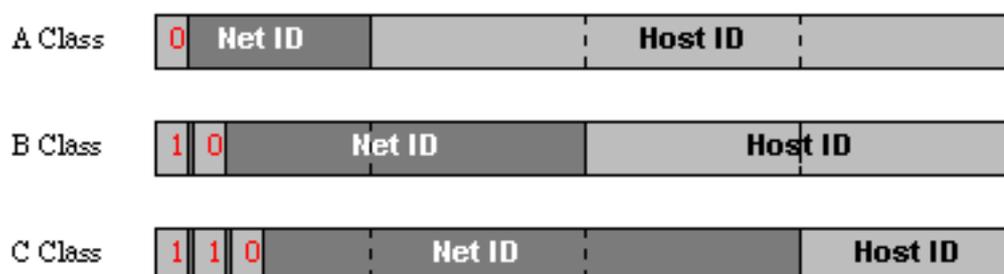
所謂的網路(Network)是指兩台以上的電腦透過傳輸媒介(有線或無線皆可)串接，可以和彼此分享資源的架構。而網際網路(Internet)，就是串接全世界網路的架構。

我們若想要將自己的電腦連上網際網路，就得透過網際網路服務提供商(Internet Service Provider, ISP)。現今一般住家最普遍的連線方式就是透過 ISP 提供的 ADSL 上網服務。ADSL 的全名為 Asymmetric Digital Subscriber Line，其特徵為上傳和下載的頻寬並不相同，目前市面上一般用戶可申裝的 ADSL 速度最高可達 12M/1M (下載/上傳)。

一般我們要描述網路的速度，會使用頻寬來計算。所謂的頻寬是指在單位時間內，傳輸媒介所能傳輸的資料量。常用的單位為 bps，也就是 bit per second。12M/1M 換算回一般常用的 byte 來計算，實際上只有 1.5MB/128KB。

在網路的世界裡，每台電腦都必須要有一個完全獨立的代號，我們將之稱為 IP 位址。IP 位址是由 32 位元(bits)所組成，通常以 8 個位元為一個單位，將 IP 位址區分成四個部份，以十進位來表示，並以「.」區隔，如 203.64.52.1。因為是 8 個位元為一個單位，所以每個部份的十進位數字都不會超過 255。

IP 的位置依照網路的規模大小可區分成 A、B、C、D、E 五類，以 32 個位元之中最左邊的幾個位元來決定。我們生活中一般會看到的 IP 多為 A、B、C 三類，通常如下區分：



1. A 級網路：大型網路，多為國際公司。(0~127.x.x.x)  
最左邊的位元必須為(0)<sub>2</sub>，有效網路位址只剩 7 個位元(第一段)，共可決定出  $2^7=128$  個 A 級網路(Net ID)，每個網路有  $2^{24}$  個不同的 Host ID。
2. B 級網路：中型網路，多為大型企業、政府機構、大學等。(128~191.x.x.x)  
最左邊兩個位元必須為(10)<sub>2</sub>，所以剩下的 14 個位元(第一段+第二段)共可決定出  $2^{14}=16384$  個 B 級網路(Net ID)，每個網路有  $2^{16}$  個不同的 Host ID。
3. C 級網路：小型網路，多為中小學、小型公司。(192~223.x.x.x)  
最左邊三個位元必須為(110)<sub>2</sub>，所以剩下的 21 個位元可決定出  $2^{21}=2097152$  個 C 級網路(Net ID)，每個網路有  $2^8$  個不同的 Host ID

## Questions

1. 請問下列何者不可能為的ADSL網路的傳輸速率？ A: \_\_\_\_\_  
(A) 8M/640K (B)256K/256K (C)256K/64K (D)3M/768K
2. 某部遠方的電腦中存有 3.6 GB 的資料，如果用目前下載速度8Mbps 的 ADSL 網路下載這些資料，至少需要多少時間？(A) 約一小時 (B)約半小時 (C)約7.5 分鐘 (D)約一分鐘
3. 若使用 8M/640K 的 ADSL 網路下載一個 120MB 的檔案需要多少分鐘？  
A: \_\_\_\_\_ mins
4. 下列何者為 B 級網路？ A: \_\_\_\_\_  
(A) 120.9.2.1 (B)128.256.2.1 (C)192.168.0.1 (D)191.255.220.9

## PART IV 資料結構

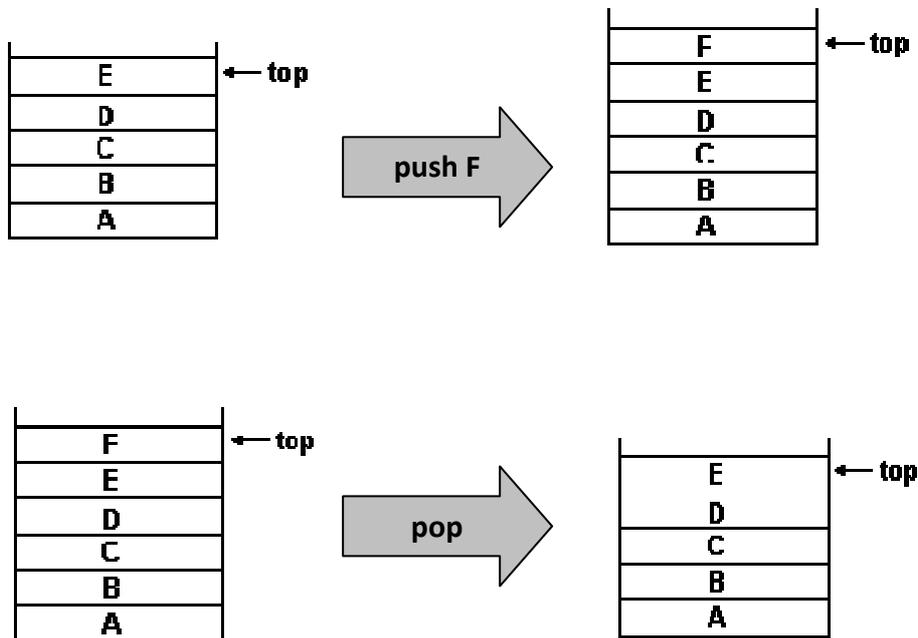
---

### Reading

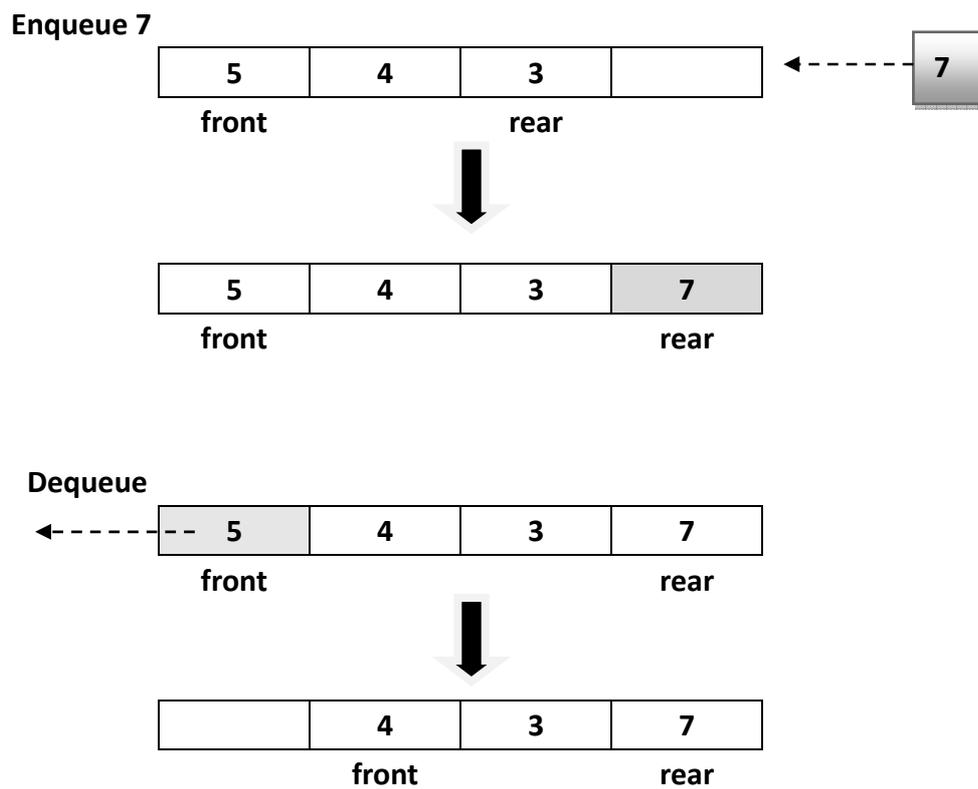
資料結構(Data Structure)主要是在研究如何把原始的「資料」(data)·組織、安排、存放到電腦中的一門學問。如果我們學好資料結構將對於節省儲存空間、增加資料的安全性與處理速度將有很大幫助·因此對於資料結構的了解與否·對日後在軟、硬體發展上有彌足深遠的影響。

堆疊(Stack)和佇列(Queue)是資訊領域相當重要且基礎的兩種資料結構·應用層面非常廣泛·常用在許多經典演算法(Algorithm)之中。

堆疊就像是個鳳梨罐頭·當鳳梨片放入罐頭時·最先放進去的會最後被拿出來·這個特性我們稱為先進後出(First In Last Out, FILO)。我們可以對堆疊進行的操作有兩種：push 和 pop。push 就是將一個元素放入堆疊·pop 則是將最上面的元素拿出堆疊。



佇列就像是羽球筒，從一邊放入從另外一邊拿出，最先放進去的會最先被拿出來，這個特性我們稱為先進先出(First In First Out, FIFO)。我們可以對佇列進行的操作有兩種：enqueue 和 dequeue。Enqueue 是將一個元素放入佇列，dequeue 則是將最前面的元素拿出佇列。



## Questions

1. 對堆疊執行了以下操作後，請將 Pop 出的資料依序寫出來。  
Push A  
Push B  
Push O  
Pop  
Pop  
Push AB  
Push Q  
Pop  
Pop  
Pop
2. 對佇列執行了以下操作後，請將 Dequeue 出的資料依序寫出來。  
Enqueue A  
Enqueue B  
Enqueue C  
Enqueue D  
Dequeue  
Dequeue  
Dequeue  
Enqueue E  
Dequeue  
Dequeue
3. 將一個序列 3 1 2 5 4 6 依序 Push 進堆疊，妳可以在堆疊中有元素的任意時刻執行 Pop 指令。請寫下 Push 和 Pop 的順序，以得到序列 1 2 3 4 5 6。

## PART V 程式

---

### Questions

1. 請寫出以下片段程式碼輸出結果：

```
int i, n = 0;
char str[] = "6061";
for(i=0; '0'<=str[i]&& str[i]<='9'; i++)
{
    n = n * 10 + (str[i] - '0');
}
printf("%d\n", n);
```

2. 請寫出下方程式碼中 if 內的條件式，以印出 1 到 100 之間為 2 的倍數或 3 的倍數但不為 6 的倍數的數。

```
for(i=1; i<=100; i++)
{
    if(
        )
    {
        printf("%d\n", i);
    }
}
```

3. 請寫出下列程式碼執行的輸出結果。

```
#include <stdio.h>
#define SQUARE(X) X * X
int main(void)
{
    int n = 5;
    printf("%d %d\n", SQUARE (n), SQUARE (n + 1));
    return 0;
}
```

4. 下列為一副程式的虛擬碼，其中 **B** 為函數名稱，而 **K**、**n**、**m** 為變數。若輸入之 **n** 為 5、**m** 為 3。(1) 請寫出下列程式呼叫 B(5,3)的執行結果 \_\_\_\_\_。(2) 此一副程式將會被呼叫幾次? \_\_\_\_\_。

```
Function B(n,m)
begin
    if (n = 0 || m = n) then K = 1;
    else K = B(n-1,m) + B(n-1,m-1);
    return (K);
end
```

