

# 《資料結構》

## 試題評析

本次高考資料結構試題十分淺顯，並沒有特別艱澀或深入的問題，各題說明如下。

第一題：是簡單的迴圈次數計算，只有最後第三小題考生較可能觀察不出來複雜度。

第二題：為字串比對的問題，但使用的是普通的做法，所以小心計算次數便可取得分數。

第三題：是樹的基本特性與公式推導，屬不困難的計算與證明，分數亦可輕鬆取得。

第四題：是圖形的拓樸排序，屬基礎的題型，亦應拿分。

第五題：考二分搜尋法，也是簡單的問題。

綜觀今年試題，預估考生可以拿到不錯的成績，一般考生應可拿到75分以上的分數，準備充分者應可拿到90分以上。

一、本題是關於演算法效率分析 (Algorithm and performance analysis)

(一)請分別寫出下列程式第一行 (line 1) 到第五行 (line 5) 的執行次數 (frequency count)，於試卷上請標明是幾行，次數是多少。(10分)

```
void mult (int a[][n], int b[][n], int c[][n])
{
    int i, j, k;
    for(i=0; i<n; i++)..... line 1
        for(j=0; j<n; j++)..... line 2
        {
            c[i][j]=0;..... line 3
            for(k=0; k<n; k++)..... line 4
                c[i][j]=a[i][k]*b[k][j];.... line 5
        }
}
```

(二)於下列程式，請計算指令x++；一共會執行多少次？(5分)

```
for(i=0; i<n; i++)
    for(j=i+1; j<n; j++)
        x++;
```

(三)請根據下列表格的數據，size是問題量 (或問題大小)，count是程序指令的總執行次數，來推測程式執行的時間複雜度 (time complexity)，請以Big-Theta  $\Theta$ 表示之 (例如： $\Theta(3^n)$ )。(5分)

size	1,000	2,000	3,000	4,000	5,000	6,000	7,000	8,000
count	11,863	24,227	40,003	53,217	67,393	78,961	91,985	113,997

**答：**

(一)line 1: n+1 次  
 line 2: n(n+1)次  
 line 3: n<sup>2</sup> 次  
 line 4: n<sup>2</sup>(n+1) 次  
 line 5: n<sup>3</sup> 次

(二) $(n-1) + (n-2) + \dots + 3 + 2 + 1 + 0 = \frac{n(n-1)}{2}$  次

(三)以 $n=3000, 4000, 5000, 6000, 7000$  這個部分觀察， $\frac{\text{count}}{n}$  的比例大約都在13.28左右，應為線性時間，為  $\theta(n)$ 。

n	3000	4000	5000	6000	7000
count	40,003	53,217	67,393	78,961	91,985
$\frac{\text{count}}{n}$	$\frac{40,003}{3,000} = 13.33$	$\frac{53,217}{4,000} = 13.30$	$\frac{67,393}{5,000} = 13.48$	$\frac{78,961}{6,000} = 13.16$	$\frac{91,985}{7,000} = 13.14$

二、關於字串樣式比對 (string pattern matching)，最簡單的方法是使用窮舉樣式比對法 (exhaustive pattern matching)，此即將樣式 (pattern) 的字元逐一比較本文 (text) 的字元，若不對則移下一字元繼續比對，直到比對成功或本文剩下的字元數目少於樣式長度。

(一) 假設本文是：THERE\_IS\_MORE\_TO\_LIFE\_THAN\_INCREASING\_ITS\_SPEED，欲找尋的樣式 (pattern) 為GENTLE，問：

1. 總共比較多少次？(5分)
2. 一共比較多少個字元？(5分)

(二) 假設本文是一千個"0"，欲找尋的樣式 (Pattern) 為01010，請問：

1. 總共比較多少次？(5分)
2. 一共比較多少個字元？(5分)

**答：**

- (一) 1. pattern 總共比對42次。  
2. 共比較43個字元。  
(二) 1. pattern 總共比對996次。  
2. 共比較1992個字元。

三、(一) 說明樹 (tree) 與二元樹 (binary tree) 有那三項主要的不同？(5分)

(二) 已知某一樹其分支度 (degree) 為1的節點 (node) 有5個，分支度為2的節點有4個，分支度為3的節點有3個，分支度為4的節點有2個，分支度為5的節點有1個，請問此樹一共有幾個節點？(5分)

(三) 證明：於任意一個二元樹中，若  $n_0$  代表分支度為0的節點數目， $n_1$  代表分支度為1的節點數目， $n_2$  代表分支度為2的節點數目，則  $n_0 = n_2 + 1$ 。(10分)

**答：**

- (一) 1. 樹不能是空的；二元樹可以是空的。  
2. 二元樹的degree為2；樹的degree可以大於2。  
3. 即使樹的degree等於2時，亦不見得是二元樹，因為二元樹有定義左、右子樹；而樹沒有這些定義。

(二)  $B = n - 1 = n_0 + n_1 + n_2 + n_3 + n_4 + n_5 - 1 = n_0 + 5 + 4 + 3 + 2 + 1 - 1 = n_0 + 14$

又另一式

$$B = 1 \times n_1 + 2 \times n_2 + 3 \times n_3 + 4 \times n_4 + 5 \times n_5 = 1 \times 5 + 2 \times 4 + 3 \times 3 + 4 \times 2 + 5 \times 1 = 35$$

$$\text{故 } n_0 + 14 = 35$$

$$\text{即 } n_0 = 21$$

$$\text{節點總數 } n = n_0 + n_1 + n_2 + n_3 + n_4 + n_5 = 21 + 5 + 4 + 3 + 2 + 1 = 36。$$

(三)  $b = n - 1$  .....(1)

$$n = n_0 + n_1 + n_2 \quad \dots\dots(2)$$

(2) 代入 (1) 得

$$b = n_0 + n_1 + n_2 - 1 \quad \dots\dots(3)$$

又分支亦有下面關係式

$$b = 0 \times n_0 + 1 \times n_1 + 2 \times n_2 = n_1 + 2n_2 \dots(4)$$

(3) 與 (4) 應相等

$$n_1 + 2n_2 = n_0 + n_1 + n_2 - 1$$

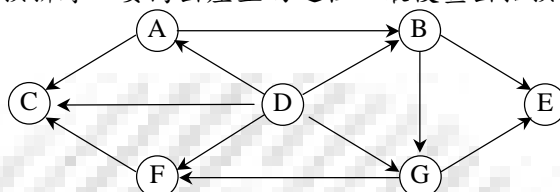
化簡即得  $n_0 = n_2 + 1$ 。

四、一個有向圖形 (directed graph)，若圖形的任何路徑 (path) 沒有環路 (cycle)，則此圖形可找到拓撲排序 (topological sorting)，問：

(一) 說明什麼是拓撲排序？(5分)

(二) 舉出一種拓撲排序的應用。(3分)

(三) 於下圖中找出一種拓撲排序，要寫出產生的過程，最後畫出拓撲排序圖。(12分)

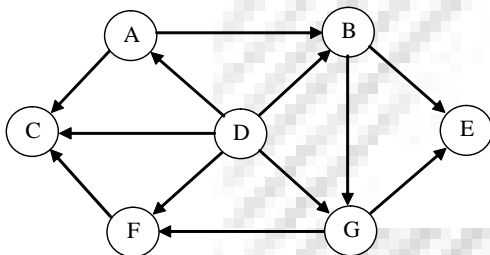


**答：**

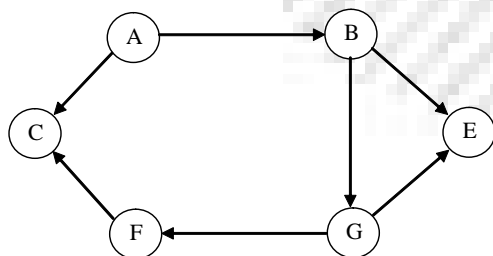
(一) 拓撲排序是將有向圖的節點排成一組線性順序，且若其中  $u$  為  $v$  的 predecessor，則  $u$  必須排在  $v$  之前。

(二) 一群有部分先後次序關係的工作之排程。

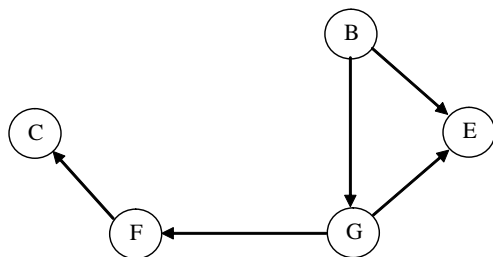
(三) 原始圖形



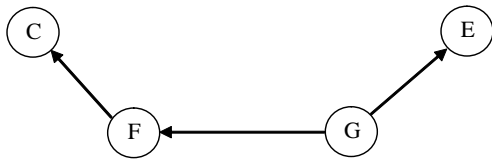
輸出：D



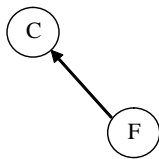
輸出：A



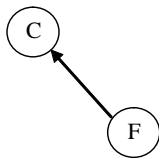
輸出：B



輸出：G



輸出：E



輸出：F



輸出：C

拓樸順序：DABGEFC

- 五、假設有一個陣列 $A[0..12]$ ，儲存13個數字：4，14，25，31，37，42，56，70，73，83，86，90，94。今使用二元搜尋 (binary search)，問：
- (一) 寫出找尋70的比較過程 (沒寫過程不予計分)。(8分)
  - (二) 列出比較次數最多的所有數字。(6分)
  - (三) 假設現有100,000個數字已經依由小而大的次序排列好，請分別使用二元搜尋 (binary search) 與循序搜尋 (sequential search)，計算兩者成功找尋 (successful search) 的平均比較次數，並說明兩者大概相差多少倍？(6分)

**答：**

(一)A[0..12]中間項 $56 < 70$ 搜尋A[7..12]

A[7..12]中間項 $83 > 70$ 搜尋A[7..8]

A[7..8] 中間項  $70 = 70$  搜尋成功。

(二)搜尋次數最的有：14,31,42,73,86,94

(三)二分搜尋法平均時間為 $\log_2 n = \log_2 100000 \approx 16.6$ 次

循序搜法平均時間為 $\frac{n+1}{2} = \frac{100001}{2} \approx 50001$ 次

相差大概3012倍。

