《資料結構》

試題評析

本次命題大致分爲四大主題：包括「陣列與指標」、「二元搜尋樹」、「圖形基本觀念」以及「遞迴」。由於考題難度不高，得分容易，反而難以測出考生程度。

第一題：以「指標與多維陣列」的關聯性命題，主要在測驗考生是否熟悉「C程式語言」的特性。不過試題中的程式，有些敘述可能有筆誤，造成題意模糊，因此閱卷與評分上存在些許不確定因素。

第二題：以「二元搜尋樹」的搜尋特性命題，計算搜尋時的比較次數，除了成功搜尋之外，也要注意搜尋失敗的狀況。

第三題：測驗考生有關圖形的基本定義與特性，對圖形熟稔的考生，要得分並不困難。

第四題：以常見的「遞迴範例Ackermann's Function」命題，了解此函數定義的考生，作答上應該不會有困難。

綜觀本次高考資料結構試題，除了第一題偏向程式語言範疇，考題方向較爲意外，其餘應當可輕易得分。預估一般考生應當可以拿到70分以上，程度佳者應可拿到85分以上。

一、現有變數宣告如下：（每小題5分，共25分）

```
int intArray[3][2] = {{10, 20}, {15, 25}, {50, 40}};
int ** intPtrl = intArray;
int * intPtr2 = &intArray[1][1];
int * intPtr3[2] = &intArray[2];
```

intArray的記憶體位址是0x0008600；int為sizeof(int) = 4；

試回答下列問題（如果是正確的敘述請寫出左邊變數的數值，錯誤請說明原因，但每題題目是有關連性的）：

(一)*intPtr2 = intArray[1][1];

(二) intPtr1 + 1 = intArray[0] ;

(三)++intPtr = &intArray[1];

(四)*(*intPtr + 1) = intArray[1][0];

(五)*(*intPtr3 + 1) = intArray[2][1];

答：

(一)正確，將數值 25 存入位址 0x0008600+ 4 * 3 = 0x000860C 的記憶體中。

(二)錯誤，運算式不可置於 = 的左側(因爲 運算式沒有 L-value) 。

(三)錯誤，intPtr 未宣告。

註：題目可能筆誤，原來的題意可能是

```
++intPtr1=&intArray[1];
```

則答案應爲 0x0008600 + 4 * 2 = 0x0008608 存入 intPtr1。

(四)錯誤，intPtr 未宣告。

註：題目可能筆誤，原來的題意可能是

```
(*intPtr1 + 1) = intArray[1][0];
```

相當於

```
intPtr1[0][1]=intArray[1][0];
```

由於前一小題已將 intPtr1指向 intArray[1]

則答案應爲 將15存入位址 0x0008608+ 4 * 1 = 0x000860C，相當於將 15 存入 intArray[1][1] 中。

(五)錯誤，宣告 int * intPtr3[2]=&intArray[2]; 有compile-time error。

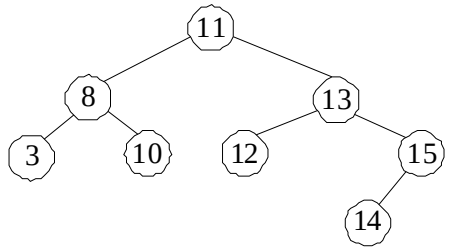
註：題目可能筆誤，若將宣告改成如下即可執行

```
int (* intPtr3)[2]=&intArray[2];
```

則 *(*intPtr3 + 1) = intArray[2][1];

會將值 40 存入 *(*intPtr3 + 1)。

二、有一個二元搜尋樹（Binary Search Tree） T如下：



(一)若欲搜尋的鍵值（Key）平均分布在1到100之間，請算出該值於搜尋樹中平均要比較幾次。（5分）

(二)設鍵值K = 2時，其機率為0.5，K = 5時其機率為0.3，K = 9時其機率為0.103，其餘97個數機率均為0.001，請算出該值於搜尋樹中要比較幾次。（10分）

(三)設各鍵值的機率如上述第（二）小題，是否能將此搜尋樹重新安排以獲得較佳的平均比較次數？請說明原因或理由。（10分）

答：

(一)

key	比較次數	機率
1~2(失敗)	3	0.02
3	3	0.01
4~7(失敗)	3	0.04
8	2	0.01
9(失敗)	3	0.01
10	3	0.01
11	1	0.01
12	3	0.01
13	2	0.01
14	4	0.01
15	3	0.01
16~100(失敗)	3	0.85

$$1\times 0.01+2\times(0.01+0.01)+3\times(0.02+0.01+0.04+0.01+0.01+0.01+0.01+0.85)+4\times 0.01$$
$$=1\times 0.01+2\times 0.02+3\times 0.96+4\times 0.01=2.97$$

(二)

key	比較次數	機率
1(失敗)	3	0.001
2(失敗)	3	0.5
3	3	0.001
4(失敗)	3	0.001
5(失敗)	3	0.3
6~7(失敗)	3	0.002
8	2	0.001
9(失敗)	3	0.103
10	3	0.001
11	1	0.001
12	3	0.001
13	2	0.001
14	4	0.001
15	3	0.001
16~100(失敗)	3	0.085

$$1\times 0.001+2\times 0.002+3\times 0.996+4\times 0.001=2.997$$

(三)重新安排搜尋樹，可以得到較佳的平均比較次數。例如：以 3 做爲樹根，可以讓機率較高的失敗節點所需的比較次數變少，而整體的平均比較次數也會變少。

三、有關圖形與樹的名詞：（每小題5分，共25分）

（一）請說明何謂擴張樹（Spanning Tree）。

（二）請說明何謂雙連通圖（Biconnected Graph）。

（三）請說明何謂二分圖（Bipartite Graph）。

（四）請說明每個樹是否均屬於二分圖。

（五）請說明每一個高度平衡二元樹（AVL）是否均屬於完滿二元樹（Fully Binary Tree）。

答：

(一)於圖形中選取 lvl-1 個邊(edges)以構成一個非循環的連通圖形(acyclic connected graph)，即稱爲擴張樹。

(二)不包含關節點(articulation node)的圖形，即稱爲雙連通圖形。

(三)圖形的頂點 V 可以分爲兩個互斥的集合 V_1 與 V_2，即 $V_1 \cap V_2 = \phi$ 且 $V_1 \cup V_2 = V$，而圖形中的每一個邊(edge) (u,v)，必定（$u \in V_1$且$v \in V_2$）或（$u \in V_2$且$v \in V_1$）。

(四)每個樹都是二分圖，因爲可以將奇數層(odd levels)的節點視爲 V_1 的節點；而偶數層(even levels)的節點視爲 V_2 的節點。如此可以看出，每個邊所連接的兩個節點，都分別屬於不同集合(V_1 與 V_2)。

(五)高度平衡二元樹不一定是滿二元樹，因爲高度平衡二元樹中的節點，只要兩個子樹的高度相差不超過1即可；而完滿二元樹，則兩個子樹的高度皆須相同。

四、Ackermann’s Function A(m, n)的定義如下：

$$A(m,n)=\begin{cases} n+1 & ,\text{if } m=0 \\ A(m-1,1) & ,\text{if } n=0 \\ A(m-1,A(m,n-1)) & ,\text{otherwise} \end{cases}$$

此函數的成長速度相當快，對於m和n是很小時亦然

(一)試寫一遞迴演算法（Recursive Algorithm）來計算此函數值。（15分）

(二)試求算出A(2, 2)的值。（需列出求算過程）（10分）

答：

(一)

```
int A(int m, int n)
{
    if (m==0) return n+1;
    else if (n==0) return A(m-1,1);
    else return A(m-1,A(m,n-1));
}
```

(二)

Ack(2,2)=Ack(1,Ack(2,1))=Ack(1,5)=Ack(0,Ack(1,4))=Ack(0,6)=7
Ack(2,1)=Ack(1,Ack(2,0))=Ack(1,3)=Ack(0,Ack(1,2))=Ack(0,4)=5
Ack(2,0)=Ack(1,1)=Ack(0,Ack(1,0))=Ack(0,2)=3
Ack(1,0)=Ack(0,1)=2
Ack(1,2)=Ack(0,Ack(1,1))=Ack(0,3)=4
Ack(1,4)=Ack(0,Ack(1,3))=Ack(0,5)=6

【參考書目】王致強「資料結構」講義第四章第11頁及上課筆記。