

# 《資料結構》

試題評析	<p>第一題：堆疊用於處理運算式，包括中序式轉換後序式，以及後序式求值計算，屬於基本問題。</p> <p>第二題：各種排序法操作問題，基本上拿分也不難。</p> <p>第三題：二元樹基本特性與中序追蹤，亦屬基本題型。</p> <p>第四題：Kruskal algorithm 以及實作所需之資料結構，對演算法有較深入認識，可以得到分數。</p> <p>綜合來說今年題目平易近人，涵蓋度亦適中，一般應試者可以拿 75~80 分，詳細準備者可拿 90 分以上的分數。</p>
高分閱讀	<p>第一題：王致強，《資料結構》，高點出版，pp.4-55~63。</p> <p>第二題：王致強，《資料結構》，高點出版，pp.9-63~66、pp.9-51~55、pp.9-24~27。</p> <p>第三題：王致強，《資料結構》，高點出版，pp.6-12~13(精選範例 9)、p.6-16。</p> <p>第四題：王致強，《資料結構》，高點出版，pp.8-39~41。</p>

- 一、(一)請將中序運算式  $(8 \times 3 - 6 / 2) + 5 / (1 + 4)$  轉換成後序運算式 (postfix expression)。(10 分)
- (二)請使用堆疊 (stack) 說明算出後序運算式  $1, 2, 3, *, 4, 6, +, 5, /, /, +$  的過程與結果。(10 分)

## 【擬答】

(一)後序運算式為  $8\ 3 \times 6\ 2\ /\ -\ 5\ 1\ 4\ +\ /\ +$

(二)1,2,3 :存入堆疊。

3
2
1

\* : 取出兩個運算元做乘法  $2 * 3 = 6$  存入堆疊。

6
1

4, 6 :存入堆疊。

6
4
6
1



+: 取出兩個運算元做加法  $4 + 6 = 10$  存入堆疊。

10
6
1

5: 存入堆疊。

5
10
6
1

/: 取出兩個運算元做除法  $10 / 5 = 2$  存入堆疊。

2
6
1

/: 取出兩個運算元做除法  $6 / 2 = 3$  存入堆疊。

3
1

+: 取出兩個運算元做加法  $1 + 3 = 4$  存入堆疊。

4
---

最後，由堆疊取出計算結果 4。



二、有一陣列  $A=(163, 231, 356, 93, 869, 987, 58, 349, 271, 33)$  要由小排到大。

- (一)使用基數排序法 (radix sort) 需要三個回合 (pass) 排序 A 陣列，請寫出前兩個回合結束時 A 陣列的內容。(10 分)
- (二)使用堆積排序法 (heap sort) 需要先將 A 陣列整理成 maxheap，然後再經過九個回合 (pass) 的 reheap 才能將資料由小排到大，請寫出整理成 maxheap 後與第一個回合 reheap 結束時 A 陣列的內容。(10 分)
- (三)使用快速排序法 (quick sort) 將 A 陣列排序，每一回合 (pass) 選擇待排序子陣列 (sub-array) 最左邊那筆資料做為比較基準，且左邊子陣列會比右半子陣列先處理，請寫出前兩個回合結束時 A 陣列的內容。(10 分)

【擬答】

(一)基數排序法 (radix sort)

第一回合

$A=(231, 271, 163, 93, 33, 356, 987, 58, 869, 349)$

第二回合

$A=(231, 33, 349, 356, 58, 163, 869, 271, 987, 93)$

(二)堆積排序法 (heap sort)

整理成 maxheap 後

$A=(987, 869, 356, 349, 231, 163, 58, 93, 271, 33)$

第一個回合 reheap 結束時

$A=(869, 349, 356, 271, 231, 163, 58, 93, 33, 987)$

(三)快速排序法 (quick sort)

第一個回合，以 163 為基準

$A=(33, 58, 93, 163, 869, 987, 356, 349, 271, 231)$

第二回合，以 33 為基準

$A=(33, 58, 93, 163, 869, 987, 356, 349, 271, 231)$

三、(一)有一  $N$  個節點 (node) 的二元樹 (binary tree)，令  $N_0$  代表沒有子節點的樹葉 (leaf node) 個數， $N_1$  代表只有一個子節點的節點個數， $N_2$  代表有兩個子節點的節點個數，請證明  $N_0=N_2+1$ 。(10 分)

(二)請填入下面 C 程式中三個空格以完成 ptr 指向樹根的二元樹中序追蹤 (inorder traversal) 程式並將追蹤結果顯示在螢幕上。(15 分)



```

struct node {
    struct node *left;
    int      data;
    struct node *right;};
void inorder(struct node *ptr)
{
    if(ptr !=NULL){
        _____(1)_____ ;
        _____(2)_____ ;
        _____(3)_____ ;
    }
}

```

## 【擬答】

(一)證明如下

$$b = n - 1 \quad \dots\dots(1)$$

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

(2) 代入 (1) 得

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

又分支度亦有下面關係式

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

(3) 與 (4) 應相等

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

化簡即得

$$n_0 = n_2 + 1。$$

(二)

(1) inorder(ptr→left)

(2) printf("%d",ptr→data)

(3) inorder(ptr→right)

四、(一)請寫出在無向圖中找出 Minimum Cost Spanning Tree 的 Kruskal 演算法。(15 分)

(二)請說明 heap (除了 heap sort 外) 與 disjoint set 這兩種資料結構在這個演算法中有何作用？(10 分)

## 【擬答】

(一)Kruskal 演算法

 $T \leftarrow \Phi$ ;while T 中少於  $n-1$  個 edges 且 E is not empty do

begin



(v,w) ← E 中 weight 最低的 edge;  
delete (v,w) from E;  
if (v,w) 不會造成 T 中的迴路 then add (v,w) to T  
else discard (v,w)

end;

if T 中的 edges 數少於 n-1 then not\_found;

(二) heap 用來選取 E 中 weight 最小的邊。

disjoint sets 用來判斷是否有產生迴路。

