

# 《資料結構》

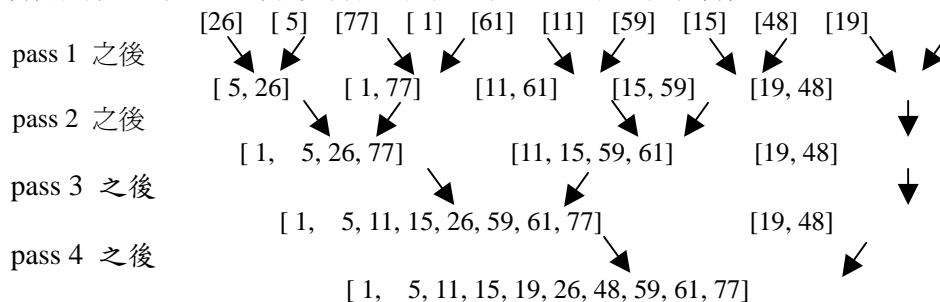
試題評析	<p>今年檢事官資料結構的命題分佈正常，五題分別測驗排序、堆積、圖形、二元樹與鏈結串列，大概比較主要的部份都有所涵蓋。不過題目難度稍微簡單了一些，因此考生要拿到高分並不難。</p> <p>第一題測驗是否知道合併排序的兩種方法，反覆合併是基本的方法，應當要得到分數；遞迴法則看考生是否有準備到。</p> <p>第二題是堆積的處理，大致上應該要拿到分數。</p> <p>第三題是圖形追蹤問題，也測驗了 arcs 的分類，只要準備過不難取分。</p> <p>第四題是二元樹追蹤問題，也十分簡單。</p> <p>第五題鏈結串列的基本操作，只要能習慣虛擬碼的表示法，應該也不難取分。</p> <p>預計今年考生拿到 70 分是基本底限，程度較好的考生應可拿到 85 以上分數。</p>
------	---

一、分別用反覆合併排序法 (iterative merge sort) 和遞迴合併排序法 (recursive merge sort) 將下列數字由小至大排序，必須列出整個排序過程。(20 分)

26, 5, 77, 1, 61, 11, 59, 15, 48, 19

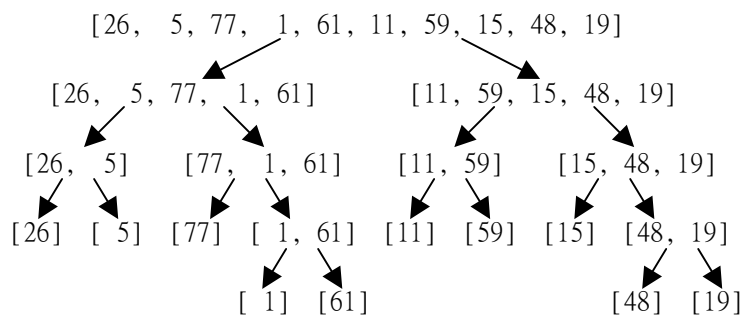
## 【擬答】

(一)反覆合併排序法，是先將每項資料各別分成一組，再將相鄰兩組合併。

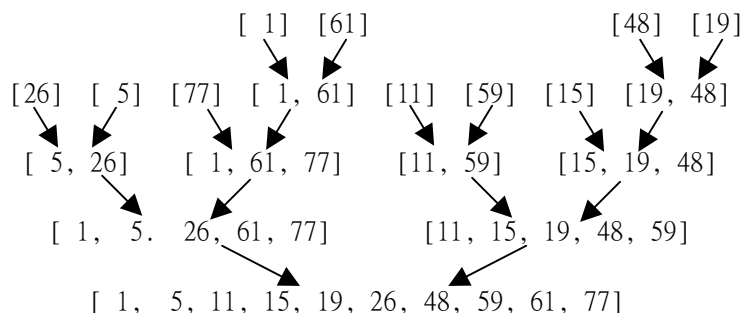


(二)遞迴合併排序法

divide phase



merge phase



二、對陣列  $A=(5, 3, 17, 10, 84, 19, 6, 22, 9)$ ，

(一)列出堆積化 (heapify) 後的陣列。(10 分)

(二)接著列出加入 20 後的陣列。(5 分)

(三)再接著列出刪除最大值後的陣列。(5 分)

【擬答】

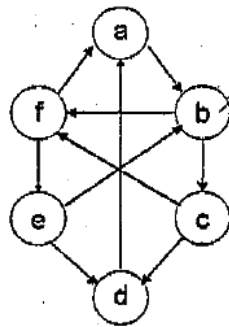
(一)堆積化之後的陣列內容為  $A=(84, 22, 19, 10, 3, 17, 6, 5, 9)$

(二)加入 20 之後的陣列內容為  $A=(84, 22, 19, 10, 20, 17, 6, 5, 9, 3)$

(三)刪除最大值之後的陣列內容為  $A=(22, 20, 19, 10, 3, 17, 6, 5, 9)$

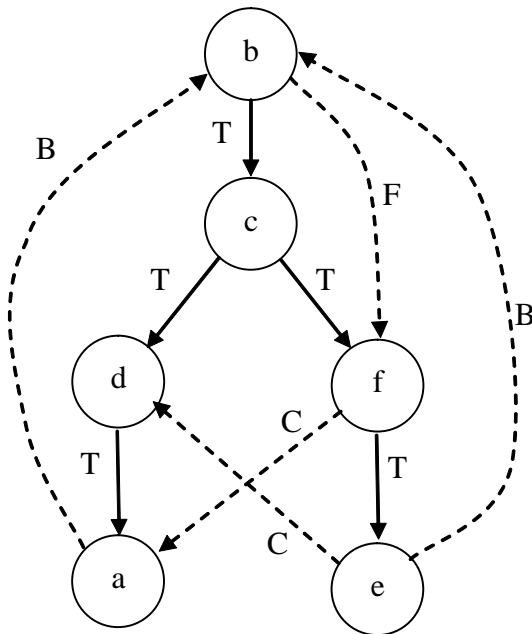
三、(一)依下圖建構以節點 (node) b 為根 (root) 的深度搜尋樹 (depth-first search tree)，掃描邊 (arc) 的順序是根據字母順序。(10 分)

(二)根據建構的深度搜尋樹，列出下圖的邊那些是樹邊 (tree arc)、前向邊 (forward arc)、後向邊 (backward arc)、交叉邊 (cross arc)。(10 分)



【擬答】

(一)



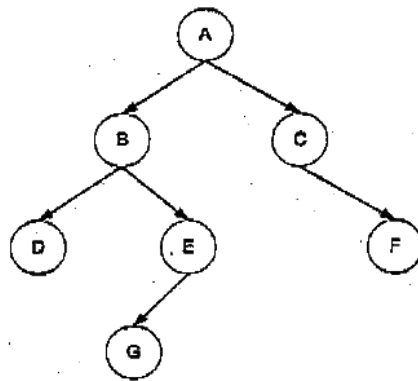
(二)tree arcs(T):  $\langle b, c \rangle, \langle c, d \rangle, \langle d, a \rangle, \langle c, f \rangle, \langle f, e \rangle$

forward arcs(F):  $\langle b, f \rangle$

backward arcs(B):  $\langle a, b \rangle, \langle e, b \rangle$

cross arcs(C):  $\langle f, a \rangle, \langle e, d \rangle$

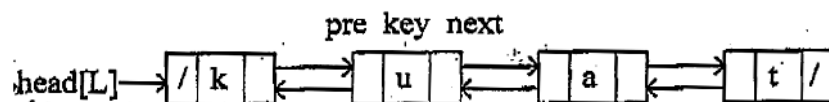
四、請將下列二元樹 (binary tree) 以左前序 (left preorder)、右前序 (right preorder)、左後序 (left postorder)、右後序 (right postorder) 方式表示。(20 分)



【擬答】

- (一)左前序(NLR)：A,B,D,E,G,C,F
- (二)右前序(NRL)：A,C,F,B,E,G,D
- (三)左後序(LRN)：D,G,E,B,F,C,A
- (四)右後序(RLN)：F,C,G,E,D,B,A

五、在下列之雙向鏈結串列(doubly linked list)的插入及刪除副程式中，依序填入空白位置，下圖為串列之示意圖。(20 分)



```

list-insert(L, x)
{
    next[x] ← head[L]
    if head[L] ≠ NIL
        then _____ (1) _____;
    head[L] ← x
    _____ (2) _____;
}

list-delete(L, x)
{
    if prev[x] ≠ NIL
        then _____ (3) _____;
    else head[L] ← next[x];
    if next[x] ≠ NIL
        then _____ (4) _____;
}
    
```

【擬答】

- (1) prev[head[L]] ← x;
- (2) prev[x] ← NIL;
- (3) next[prev[x]] ← next[x];
- (4) prev[next[x]] ← prev[x];