

## 資料結構期末考(數位二，滿分 130 分)

一、(8%)將中置式(infix)轉成前置式(prefix)和後置式(postfix)。

1.  $X = A * B / C + D / E - F * G$

2.  $A * B / (C + D) \geq E - F * G$

答案:

1. prefix:  $= X - + / A B C / D E * F G$  ; postfix:  $X A B * C / D E / + F G * - =$

2. prefix:  $\geq * A / B + C D - E * F G$  ; postfix:  $A B C D + * + E F G * - \geq$

二、(8%)將前(後)置式轉成中置式及後(前)置式。

1.  $- + A * B + C D / E F$

2.  $A B + C * D E / -$

答案:

1. infix:  $A + B * (C + D) - E / F$  ; prefix:  $A B C D + * + E F / -$

2. infix:  $(A + B) * C - D / E$  ; prefix:  $- * + A B C / D E$

三、(8%)二維陣列  $A[m_1..m_2][n_1..n_2]$  的記憶體起始位址為  $\alpha$ ，每筆資料為 4 Bytes。請回答下列問題:

1. 若資料儲存方式是 row-major，則  $A[i][j]$  的記憶體公式為何?

2. 若  $m_1=n_1=1$ ， $A[5][3]$  的記憶體位址為 5186， $A[8][5]$  的記憶體位址為 5302，請問  $A[6][3]$  的記憶體位址為何?

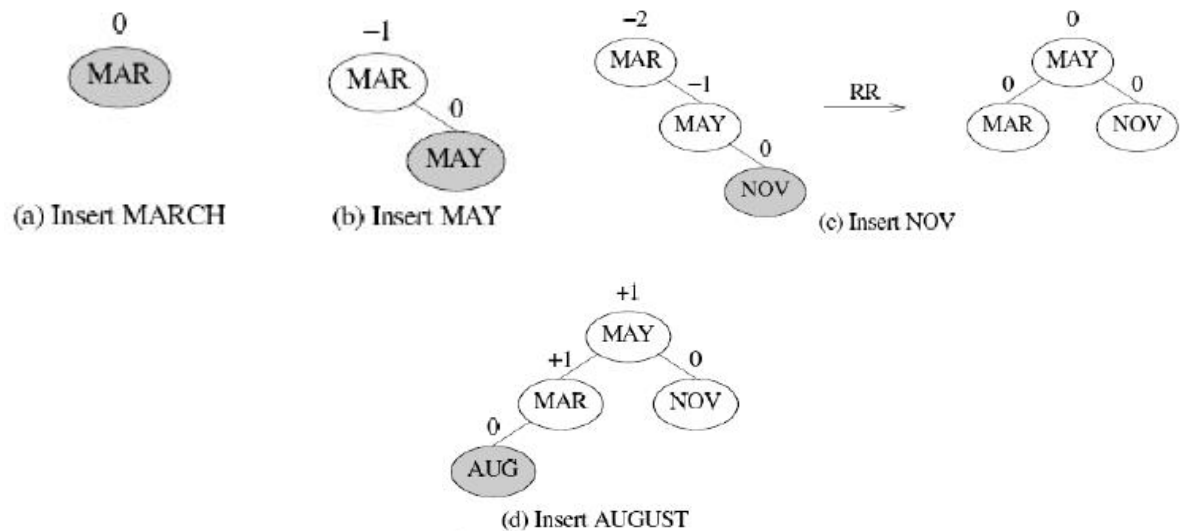
答案: (1)  $A[i][j] = \alpha + [(i - m_1) * (n_2 - n_1 + 1) + (j - m_1)] * S$  (2) 5222

四、(6%) 請導出  $n$  個碟子的河內塔的搬動次數，即為  $f(n) = 2^n - 1$ 。(推導全對才給分)

答案:

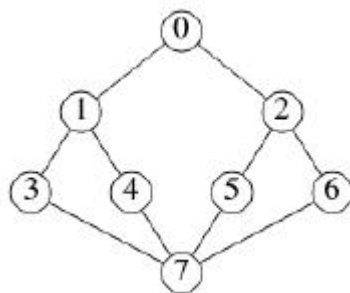
$$\begin{aligned} f(n) &= f(n-1) + 1 + f(n-1) = 2f(n-1) + 1 \\ &= 2(2f(n-2) + 1) + 1 \\ &= 2^2 f(n-2) + 2 + 1 \\ &\vdots \\ &= 2^{n-1} f(n-(n-1)) + 2^{n-2} + 2^{n-1} + \dots + 2^2 + 2 + 1 \\ &= 2^n - 1 \end{aligned}$$

五、(15%)假如我們要以 {MAR, MAY, NOV, AUG, APR, JAN, DEC, JUL, FEB, JUN, OCT, SEP} 建立一棵 AVL 樹 (An AVL tree is a height-balanced binary search tree.)，依序插入 MAR, MAY, NOV, AUG, APR 的過程如下圖(a)~(d)所示。請你從 insert APR 開始，完成所有的建構過程。註:在每插入一個之後，需算出平衡因子(balance factor)，如有導致不平衡的節點，並說明要使用哪種調整方法 (LR, RL, LL, RR)，以及調整後的結果。



答案:在第五張講義的 62~65 頁

六、(10%)下圖為一無向網路圖(undirected network)，請分別以深度優先擴張樹(Depth-first search, DFS)和廣度優先搜尋法(Breadth-first search, BFS)拜訪所有的節點(要列出次序)，並畫出最後的擴張樹(spanning tree)。

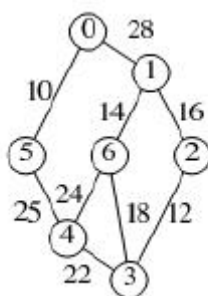


答案:

擴張樹的答案在講義第六章的 32 頁

節點順序: (1) DFS: 0 → 1 → 3 → 7 → 6 → 2 → 5 → 4 (2) BFS: 0 → 1 → 2 → 3 → 4 → 7 → 5 → 6

七、(10%)下圖為一無向網路圖(undirected network)，請從①點開始，分別以下列方法建構最小成本擴張樹(Minimum cost spanning tree)，並請依次列出加入擴張樹的邊(edge，例如：⑤⑥表示一個邊) (1) Kruskal's Algorithm (2) Prim's Algorithm (3) Sollin's Algorithm



答案:

最小成本擴張樹的答案在講義第六章的 43.45.47 頁

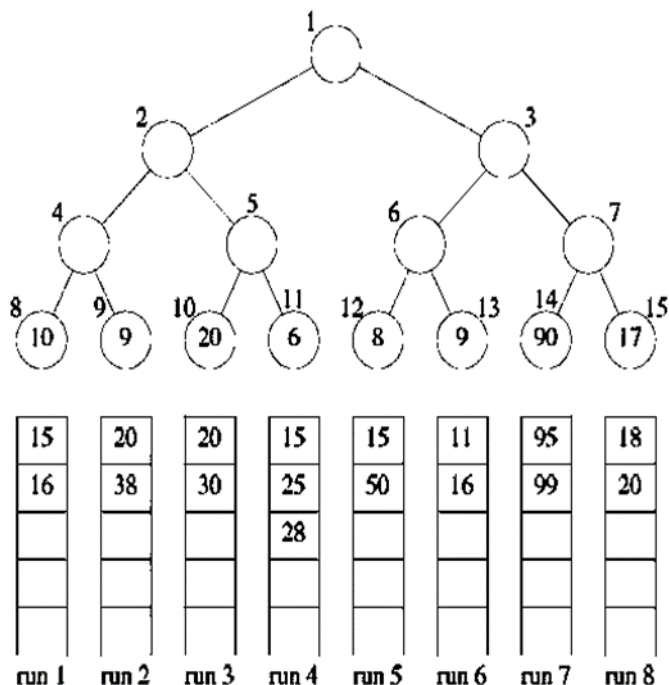
增加順序: (1) ①⑤ → ②③ → ①⑥ → ①② → ③④ → ④⑤

(2) ①⑤→④⑤→③④→②③→①②→①⑥ (3) ①⑤→①⑥→②③→③④→①②→④⑤

八、(10%)假如我們要合併已經排序好的數列，如下圖，採用 winner tree 是一個有效率的方法。

(1)請根據 winner tree 的定義，請將下圖的空白節點填滿。

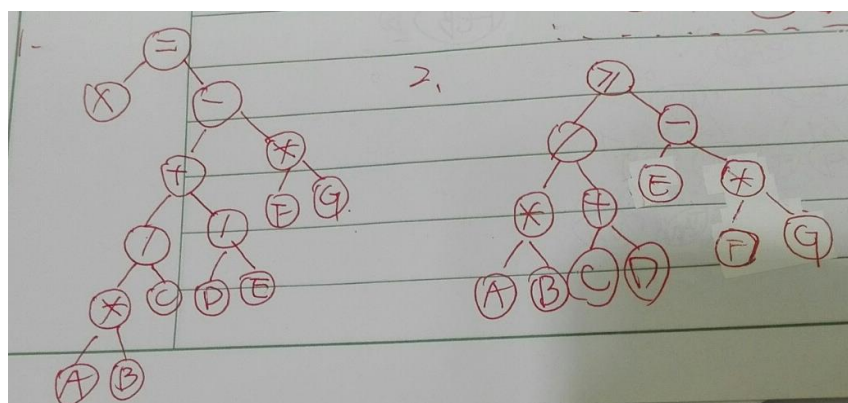
(2)承上題，如果我們從 winner tree 輸出第一個最小的樹，請畫出新的 winner tree。(註:只需畫出 tree 即可)



答案:在第五張講義的 70~71 頁

九、(10%)承第一大題，請劃出二元樹。

答案:



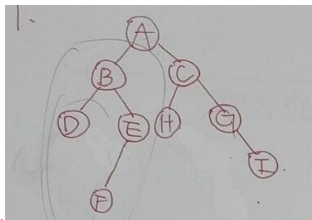
十、(15%)

1.若前序追蹤的順序為 ABDEFCHGI，中序追蹤的順序為 DBFEAHCGI，請畫出二元樹。

2.若後序追蹤的順序為 FECHGDBA，中序追蹤的順序為 FCEABHGDG，請畫出二元樹。

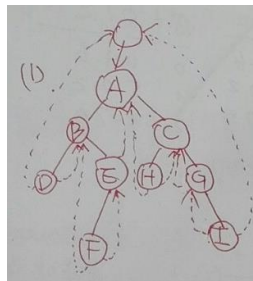
3.畫出前兩小題的引線二元樹。

答案:



1.

2. 不會 3. (1)



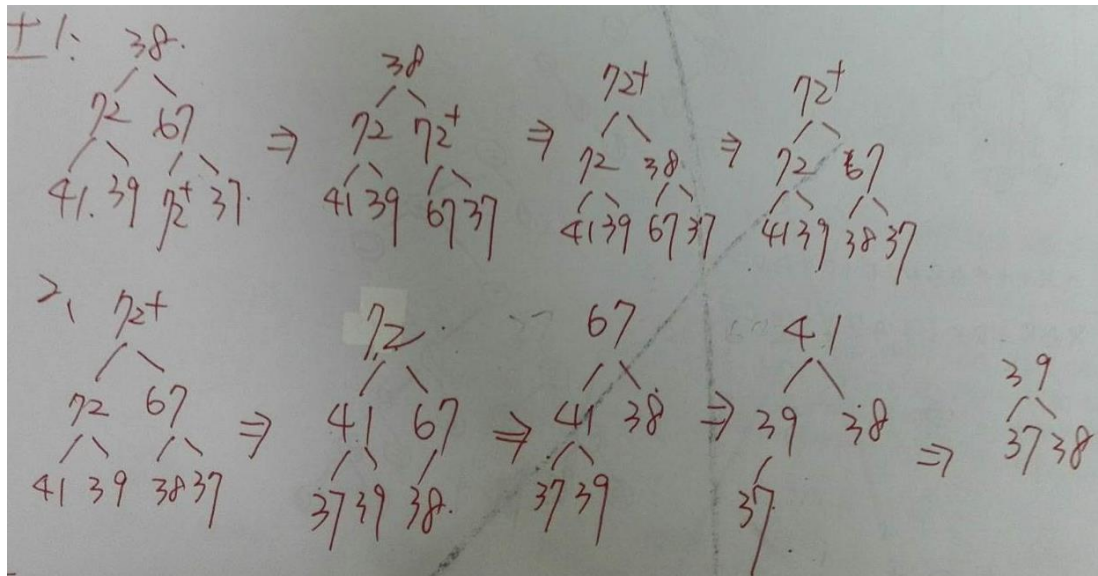
(2) 不會

### 十一、 (15%)

(1) 請先以數列  $\{38, 72, 67, 41, 39, 72^+, 37\}$  建立最大堆積樹(max heap tree)。

(2) 請畫出堆積排序法(heap sort)的完整過程。

答案:



### 十二、 (10%) (請寫出推論過程，全對才給分)

1. 若以鏈結串列表示一棵  $n$  個節點的  $k$  元樹 ( $k$ -way tree)，請問有幾個指向 null 的指標?

2. 請證明一棵二元樹的樹葉節點( $n_0$ )等於分支度為 2 的節點總數( $n_2$ )加 1。

答案:

1. (1)	二元樹應有 $2n$ 個 pointers, 但實際指向兒子的有 $(n-1)$ pointers, $\Rightarrow$ 共有 $2n - (n-1) = n+1$ 個 null pointers #.
(2)	$k$ 元樹應有 $nk$ 個 pointers, $(n-1)$ $nk - (n-1) = n(k-1) + 1$ 個 all pointers. #
2. Let $n$ 為二元樹的節點總數.	
$n_0$	樹葉節點個數 $n = n_0 + n_1 + n_2$ — ①
$n_1$	分支度為 1 的節點個數 $\Rightarrow B = 0n_0 + 1n_1 + 2n_2$ — ②
$n_2$	分支度為 2 的節點個數 $B = n - 1$ ( $\because$ 根節點無分支) — ③
$B$	的所有分支數 ①、②合併 $\Rightarrow B = n_0 + n_1 + n_2 - 1$ — ④
	③、④合併 $\Rightarrow n_0 = n_2 + 1$ #.