

資料結構期末考(數位二，滿分 100 分)

一、(6%) 名詞解釋

1. Recursion

遞迴:程式本身可以呼叫自己，或者是先呼叫別的程式，再回來呼叫自己。

2. Self-referential structure (請舉例)

自我參考結構:一個結構中，含一個或多個指向自己的指標。例如: stack、queue、linked list、tree。

3. Stack

堆疊:一個有序序列，資料的插入、刪除都是在 top 端，有後進先出(LIFO)的特性。

二、(10%) 二維陣列 $A[m_1..m_2][n_1..n_2]$ 的記憶體起始位址為 α ，每筆資料為 S 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?

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

2. 若 $m_1=n_1=0$ ， $S=4$ ， $A[4][2]$ 的記憶體位址為 72， $A[6][4]$ 的記憶體位址為 112。試求 n_2 和 α ?

$$n_2 = 3 ; \alpha = 0$$

3. 承上題， $A[8][2]$ 的記憶體位址為何? 136

三、(8%) KMP 為 pattern matching (模式比對) 問題在最差狀況 (worst case) 的最佳演算法，其 failure function 定義與範例如下圖。請求出下列各模式的 failure 值: (1) abaabaab (2) abcababca

Definition: If $p = p_0p_1 \dots p_{n-1}$ is a pattern, then its *failure function*, f , is defined as:

$$f(j) = \begin{cases} \text{largest } i < j \text{ such that } p_0p_1 \dots p_i = p_{j-i}p_{j-i+1} \dots p_j & \text{if such an } i \geq 0 \text{ exists} \\ -1 & \text{otherwise} \end{cases} \quad \square$$

For the example pattern, $pat = abcabcacab$, we have:

j	0	1	2	3	4	5	6	7	8	9
pat	a	b	c	a	b	c	a	c	a	b
f	-1	-1	-1	0	1	2	3	-1	0	1

From the definition of the failure function, we arrive at the following rule for pattern matching: If a partial match is found such that $s_{i-j} \dots s_{i-1} = p_0p_1 \dots p_{j-1}$ and $s_i \neq p_j$ then matching may be resumed by comparing s_i and $p_{f(j-1)+1}$ if $j \neq 0$. If $j = 0$, then we may continue by comparing s_{i+1} and p_0 .

(1)

j	0	1	2	3	4	5	6	7
pat	a	b	a	a	b	a	a	b
f	-1	-1	0	0	1	2	3	4

(2)

j	0	1	2	3	4	5	6	7	8
pat	a	b	c	a	b	a	b	c	a
f	-1	-1	-1	0	1	0	1	2	3

四、(16%) 將中置式(infix)轉成前置式(prefix)和後置式(postfix)。

1. $A + B * D + C / D$

prefix: ++A * B D / C D ; postfix: A B D E * + C D / +

2. $A * (B + C) / D$

prefix: $/ * A + B C D$; postfix: $A B C + * D /$

3. $A * (B + C / D)$

prefix: $* A + B / C D$; postfix: $A B C D / + *$

4. $(A + B) * C - (D - E) * (F + G)$

prefix: $- * + A B C * - D E + F G$; postfix: $A B + C * D E - F G + * -$

五、(6%) 請證明一棵二元樹的樹葉節點等於分支度為 2 的節點總數加 1。(請寫出推論過程，全對才給分)

答案:

Let n 為二元樹的節點總數.

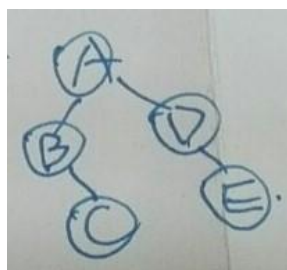
n_0	樹葉節點個數	$n = n_0 + n_1 + n_2$ — ①
n_1	分支度為 1 的節點個數	$\Rightarrow B = 0n_0 + 1n_1 + 2n_2$ — ②
n_2	分支度為 2 的節點個數	$B = n - 1$ (\because 根節點無分枝) — ③
B	的所有分支數	①、③合併 $\Rightarrow B = n_0 + n_1 + n_2 - 1$ — ④
		②、④合併 $\Rightarrow n_0 = n_2 + 1$ #

六、(10%)

1. 若前序追蹤的順序為 ABCDE，中序追蹤的順序為 BCADE，(1)請畫出唯一的二元樹，(2)請寫出後序追蹤的順序。

答案:

(1)

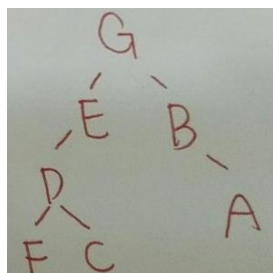


(2)後序追蹤:CBEDA

2. 若後序追蹤的順序為 FCDEABG，中序追蹤的順序為 FDCEGBA，(1)請畫出唯一的二元樹，(2)請寫出前序追蹤的順序。

答案:

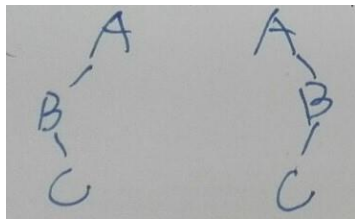
(1)



(2)後序追蹤:GEDFCBA

3. 請畫出至少兩顆二元樹，其前序追蹤的順序為 ABC，後序追蹤的順序為 CBA。

答案:



七、(10%) 請以數列{6, 4, 7, 11, 12, 8, 9, 10, 1, 2, 5, 3}建立一棵 AVL 樹 (An AVL tree is a height-balanced binary search tree.)。註:請畫出整個過程，每插入一個數字後，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:略

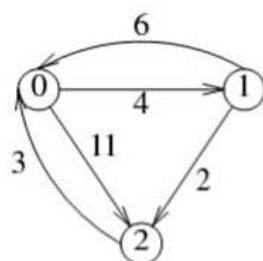
八、(5%) (承第七大題) 請進行由小而大之快速搜尋法(quick sort)。註:假設第一輪以 6 為 pivot，僅列出 6 就定位後的結果與結果即可。

答案:略

九、(5%) (承第七大題) 請先建立完整二元樹(complete binary tree)，然後調整成最大堆積樹(max heap tree)。

答案:略

十、(9%) 請完成下列矩陣值(A^0, A^1, A^2)。



(a) Example digraph

A^{-1}	0	1	2
0	0	4	11
1	6	0	2
2	3	∞	0

A^{-1}

A^0	0	1	2
0			
1			
2			

A^0

A^1	0	1	2
0			
1			
2			

A^1

A^2	0	1	2
0			
1			
2			

A^2

答案:

A^{-1}	0	1	2
0	0	4	11
1	6	0	2
2	3	∞	0

(b) A^{-1}

A^0	0	1	2
0	0	4	11
1	6	0	2
2	3	7	0

(c) A^0

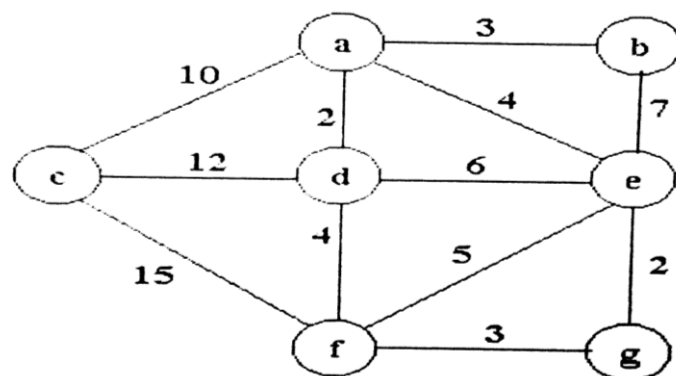
A^1	0	1	2
0	0	4	6
1	6	0	2
2	3	7	0

(d) A^1

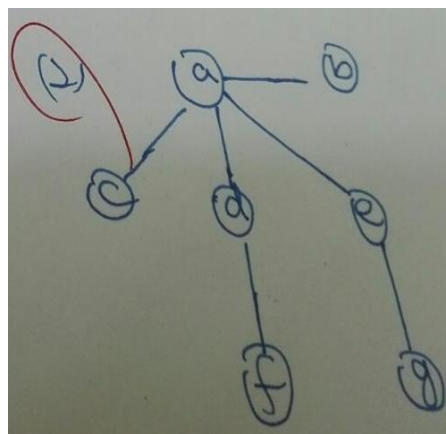
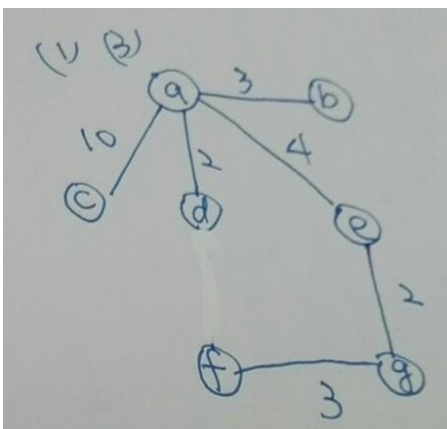
A^2	0	1	2
0	0	4	6
1	5	0	2
2	3	7	0

(e) A^2

十一、 (15%) 下圖為一無向網路圖(undirected network)，請從a點開始，分別以 (1) Kruskal (2) Prim (3) Sollin 等三種方法，畫出建構最小成本擴張樹(minimum cost spanning tree)的過程；(4)請問最小成本為何？



答案:



(4)24