

類科：資訊處理

科目：資料結構

考試時間：2小時

座號：_____

※注意：(一)禁止使用電子計算器。

(二)不必抄題，作答時請將試題題號及答案依照順序寫在試卷上，於本試題上作答者，不予計分。

(三)本科目除專門名詞或數理公式外，應使用本國文字作答。

一、給定二元樹 (binary tree) 如右圖，樹高為 4 且共有 7 個節點。

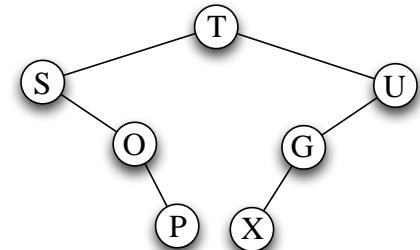
(一)請寫出該樹之後序遍歷 (postorder traversal) 結果。(5 分)

(二)若以陣列 A[1..15] 實作該二元樹，請列舉陣列 A[1..15] 的內容。

(5 分)

(三)若要將數值 x 設為或取代 A[i] (任一 $1 \leq i \leq 7$) 所代表的節點之右子節點 (right child node) 的內容，令 x 會被放入陣列中 A[j] 的位置。請以 j、i 表示，寫出 j 位置之公式。(5 分)

(四)若要在原始的二元樹中加入一些節點使其成為完整二元樹 (complete binary tree) 及完滿二元樹 (full binary tree)，請問最少各需加入幾個新節點？(5 分)

**二、遊樂園設計公司正在設計新的遊樂園，遊樂園將有 9 個遊樂設施，設施名稱暫定為 A, B, C, D, E, F, G, H, I。遊樂設施之間將透過不盡相同距離但極具特色的商店街相連。給定遊樂園的初步規劃如表一，表內數字為兩遊樂設施之間商店街道之預計長度 (每條街道長度皆不同)。若無數字則代表兩遊樂設施之間沒有商店街道之規劃。設計公司將依不同考量來決定實際建置那些商店街道。**

表一

	A	B	C	D	E	F	G	H	I
A		18		5	13	1			
B	18		9		10	16			8
C		9			11		7	12	3
D	5				17		19		
E	13	10	11	17			2		
F	1	16						14	15
G			7	19	2			4	
H				12		14	4		6
I		8	3			15		6	

(一)若要節省開發預算，在可到達所有遊樂設施的前提下，所建置的商店街道總長度需越短越好，請問可以用那一個演算法來選擇應建置的街道？請給演算法名稱並簡單說明該演算法特性。(5 分)

(二)請計算符合上述(一)小題條件下，所應建置的商店街道總長度，並由小到大列舉所有應該建置街道的長度。(10 分)

(三)但若要規劃一條路徑，使得遊客可以從任一遊樂設施開始玩，且只要依照該路徑行走，就可以玩遍 9 項遊樂設施並回到起始的遊樂設施，遊客所需走過的商店街道總長度需越短越好且每項遊樂設施只能到達一次。請問此問題最適合用下列那一種演算法來幫忙找到所應開發的街道：尤拉迴路 (Euler Cycle)，漢密爾頓迴路 (Hamiltonian Cycle)，旅行商人問題 (Traveling Sales Man Problem)，最短路徑 (例如 Dijkstra 演算法)，任兩點最短距離 (例如弗洛伊德(Floyd-Warshall)演算法)？(5 分)

(請接背面)

類科：資訊處理
科目：資料結構

三、表二列出五種常見的排序演算法，請填滿該表以顯示各排序法在最佳情況、一般情況、最壞情況下的時間複雜度、所需額外記憶體空間及是否為穩定排序法。快速排序法的各項資料已事先填入作為範例。((a),(b),(c),(d)各5分，共20分)

表二

	最佳情況 (best case)	一般情況 (average case)	最壞情況 (worst case)	所需額外空間	是或不是 穩定排序法 (stable sort)
快速排序法 (quicksort)	$O(n \log n)$	$O(n \log n)$	$O(n^2)$	$O(n)$	不是
(a)泡沫排序法 bubble sort					
(b)插入排序法 insertion sort					
(c)合併排序法 merge sort					
(d)堆積排序法 heap sort					

四、矩陣相乘是問題解決中常見的計算，但相乘順序對於計算效能有極大的影響。給定 n 個矩陣， A_1, A_2, \dots, A_n ，且任一矩陣 A_i 大小為 $p_{i-1} \times p_i$, p_0, \dots, p_n 皆為正整數。 $A_1 \times A_2 \times \dots \times A_n$ 實際計算過程可以是 $\dots((A_1 \times A_2) \times A_3) \times \dots \times A_n$ 、 $(A_1 \times (A_2 \times (\dots \times (A_{n-2} \times (A_{n-1} \times A_n))\dots)))$ 、或其他合理的順序，而因矩陣相乘順序不同，所需要的乘法運算次數可能也會不同。透過動態規劃 (dynamic programming)、二維陣列的應用及遞迴程式，可以找到最少乘法運算次數的計算順序。方法如下：令 $m[i, j]$ 為計算 $A_i \times A_{i+1} \times \dots \times A_j$ 時所需最少乘法運算次數， $m[i, j]$ 可以下列遞迴公式表示之：

$$m[i, j] = \begin{cases} \min_{i \leq k < j} \{m[i, k] + m[k+1, j] + p_{i-1} p_k p_j\}, & \text{if } i < j \\ 0, & \text{if } i \geq j \end{cases}$$

- (一) 請說明 $A_1 \times A_2 \times \dots \times A_n$ 相乘過後的矩陣大小為何？(3分)
- (二) 透過上述方法所找到的最少乘法運算次數，應為二維陣列 $m[i, j]$ 中的那個元素，亦即 i, j 應分別為何？(3分)
- (三) 若 $n = 4$ 且 p_0, p_1, p_2, p_3, p_4 分別為 3, 4, 5, 4, 2，請計算並填寫出二維陣列 $m[i, j]$ 。(11分)
- (四) 承上小題(三)，請說明該四矩陣相乘， $A_1 \times A_2 \times A_3 \times A_4$ ，最少共需有幾次乘法運算。(3分)

五、請依序將 17, 23, 36, 13, 38, 11, 52, 44, 25, 35, 2, 18, 21 儲存至下列 13 桶 (buckets) $\times 1$ 槽 (slots) 的雜湊表 (hashing table)。請以各小題所設定的雜湊函式 (hashing function) 將資料依序存入並顯示最後的雜湊表。

雜湊表	0	1	2	3	4	5	6	7	8	9	10	11	12

- (一) 雜湊函式 $F(x) = x \bmod 13$ ，碰撞時，採取「線性探測法」(open addressing with linear probing)來放入資料。請顯示最後的雜湊表。(5分)
- (二) 雜湊函式 $F(x) = x \bmod 13$ ，碰撞時，採取「二次方探測法」(open addressing with quadratic probing)來放入資料。請顯示最後的雜湊表。(5分)
- (三) 雜湊函式 $F_1(x) = x \bmod 13$ ，碰撞時，採取「雙探測法」(open addressing with double hashing)來放入資料，第二雜湊函式為 $F_2(x) = 7 - (x \bmod 7)$ 。請顯示最後的雜湊表。(5分)
- (四) 若雜湊表夠大(例如 slots = 2 或更大)但資料量多時，針對三種碰撞時所採取的處理方式，請說明那一種方式較能有效率的儲存或搜尋資料？請說明那一種處理方式效率最差？(5分)