

等 別：三等考試

類 科：資訊處理

科 目：資料結構

考試時間：2 小時

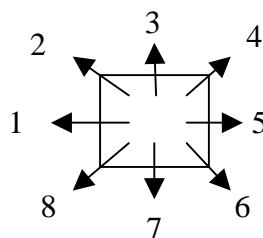
座號：_____

※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。

- 一、左下圖為一 6×6 迷宮，其中灰色區域表示不可通行，其餘可通行區域則有編號，入口與出口分別位於左上角（編號 0）與右下角（編號 35）。假定每個區域有八個可能的行動方向，找尋出口路徑時，會依序嘗試此八個方向（次序請參考右下圖之箭頭編號）。利用深度優先搜尋（depth-first search），在迷宮中探索所有可行路徑（路徑中不可包含重複的區域）。

0	1	2	3	4	
6	7			10	
	13	14	15		17
18		20		22	
	25	26			
30		32	33	34	35



- (一)利用區域編號，寫出找到的第一條路徑（編號間用逗號隔開）。（4分）
- (二)利用區域編號，寫出找到的倒數第二條路徑（編號間用逗號隔開）。（4分）
- (三)找到第一條路徑前，曾經拜訪過但最後未出現在該路徑上之區域有那些？（利用區域編號作答）（5分）
- (四)若迷宮大小改為 2×2 ，且所有區域均可通行，仍以左上角與右下角區域做為入口與出口，則共存在多少種不同路徑？（5分）
- 二、線性探測（linear probing）、平方探測（quadratic probing）與雙雜湊（double hashing）可用來解決雜湊（hashing）時發生的碰撞（collision）問題，它們使用不同的 $g(\text{key}, i)$ 函數來決定發生第 i 次（ $i \geq 0$ ）碰撞時，鍵值 key 在雜湊表中的探測位置。
- (一)線性探測為何會產生群集（clustering）問題？假設雜湊函數與表格容量分別為 $h(\text{key})$ 與 T ，先寫出其 $g(\text{key}, i)$ 函數後，再說明之。（4分）
- (二)問題同(一)，但將線性探測改為平方探測。（4分）
- (三)設計雙雜湊函數時，有何基本原則？先寫出其 $g(\text{key}, i)$ 函數，再說明之。（4分）
- (四)在何種狀況下，使用雙雜湊才能探測到雜湊表中所有可用的位置？（3分）
- (五)某空雜湊表共有 7 個位置，使用線性探測來排解碰撞問題。假設鍵值 k_1, k_2, k_3 均對應至相同的雜湊值 4，先依序將 k_1, k_2 與 k_3 加入雜湊表後，再刪除 k_2 。請問此時查詢表中是否含有 k_3 ，其結果為成功或失敗？請配合圖形說明之。（5分）

(請接背面)

等 別：三等考試
 類 科：資訊處理
 科 目：資料結構

三、二元搜尋樹 (binary search tree)：

- (一)二元搜尋樹與堆積 (heap) 有何主要相異點？寫出二項。(6分)
- (二)將一含有 n 個節點 (n>1) 之二元搜尋樹以堆積來表示，並以一陣列來儲存此堆積，請問此陣列容量可能之最小值與最大值分別為何？請說明原因。(6分)
- (三)已知二元樹節點含有二個指標欄位以指向其左子與右子，請問一棵具有 n 個節點 (n>=1) 的二元樹，存在多少個空的指標欄位 (也就是其值為 NULL)？請說明原因。(5分)
- (四)依序將整數鍵值 45, 33, 17, 65, 54, 70, 88, 25 加入一棵空的二元搜尋樹，再繪製此二元樹對應之引線樹 (threaded tree)。(8分)

四、排序 (Sorting)：參考下表之程式一與程式二回答問題。

- (一)程式一是一那一種排序演算法的實作？又其中的 s 變數有何功用？(6分)
- (二)當程式一結束執行後，第 9 行的 swap() 函數共被呼叫幾次？又此時變數 i 的值為何？(6分)
- (三)程式二是那一種排序演算法的實作？當其中的 while 迴圈第一輪執行完畢後，陣列 a 的內容為何？(6分)
- (四)參考程式二，假設陣列 a 的元素個數為 N (N>1)，若要成功完成排序，整數變數 p 的值需有那些限制？請說明原因。(7分)

	程式一	程式二
1	int a[] = {17, 88, 19, 7, 43, 23, 56, 35};	int a[] = {17, 88, 19, 109, 7, 43, 1, 23, 99, 56};
2	int n = 8;	int n = 10;
3	int i, j, s=0;	int i, j, g = n, p=2;
4	for (i = 0; i<n-1 && s==0; i++){	while ((g=g/p)>0) {
5	s = 1;	for (i = g; i<n; i++) {
6	for (j=0; j<n-i-1; j++) {	int temp = a[i];
7	if (a[j]>a[j+1]) {	for (j = i; j>=g && a[j-g]>temp; j-=g) {
8	/*將 a[j]與 a[j+1]互換*/	a[j] = a[j-g];
9	swap(a, j, j+1);	}
10	s = 0;	a[j] = temp;
11	}	}
12	}	}
13	}	

五、參考下右圖形 (graph) 回答問題，頂點 (vertices) 中的數字為頂點編號，邊 (edge) 上的數值代表成本 (cost)。

- (一)分別使用相鄰矩陣 (adjacency matrix) 與相鄰串列 (adjacency list) 來儲存此圖時，何者所需之記憶體空間較小？假設節點編號與邊值均不大於 255，且指標欄位需占用 4 個位元組 (byte)。(5分)
- (二)利用 Sollin 演算法 (Sollin's Algorithm) 找出此圖的最小成本生成樹 (minimum cost spanning tree)，須按步驟寫出此樹的成長過程。(7分)

