

《資料結構》

試題評析	本次試題十分平易沒有刁鑽的題目，主要集中在樹、堆積、遞迴範圍。除了第二題可能會因為不熟悉插入排序遞迴程式，而沒取得理想分數外，其餘試題都算容易取分。大致而言一般考生的分數理應在75~85分之間，程度佳者可以達到90分以上。各題的特點如下： 第一題：霍夫曼編碼的基本問題，須說明霍夫曼演算法的建立、編碼與解碼過程。 第二題：遞迴版的插入排序法程式，計算時間複雜度等級。 第三題：堆積基本操作。 第四題：互斥集合的樹表示法。
高分命中	第一題：《高點資料結構講義》，王致強編撰，頁11-9。 第二題：《高點資料結構講義》，王致強編撰，頁9-3。 第三題：《高點資料結構講義》，王致強編撰，頁7-3~7-6。 第四題：《高點資料結構講義》，王致強編撰，頁6-64~6-68。

一、假設我們有一個由26個英文字母所構成的文字檔。

- (一)請說明如何建構一棵霍夫曼樹(Huffman tree)來壓縮該文字檔。(15分)
- (二)請說明如何利用你所述之方法建構的霍夫曼樹壓縮該文字檔。(5分)
- (三)請說明如何解壓縮利用你所述方法壓縮的文字檔。(5分)

答：

- (一)1.先統計每個英文字母的出現頻率。
 2.採用霍夫曼演算法(Huffman's Algorithm)先建立 Huffman Coding Tree，方法如下：
 - (1)合併頻率最低的兩個，其頻率相加並建立一個父節點。
 - (2)重覆步驟(1)，合併到只剩下一個節點為止。
- (二)根據合併之關係，對每一次合併的兩項分別配置一個bit，一個配置"0"；另一個配置"1"，即可得到每個字母的霍夫曼碼(Huffman codeword)。並將檔案中每個字母以對應的霍夫曼碼表示。
- (三)解碼時由霍夫曼樹根開始，根據"0"或"1"來決定往左子樹或右子樹搜尋，到達樹葉時即可解出該字母。然後再重新由樹根，重覆同樣程序解出次一字母，餘此類推。

二、下列的虛擬碼程式片段中，I和S均為遞迴函式(recursive function)，I和S的參數A是一個整數陣列；I和S的參數i為不為負的整數，主要是做為陣列A的索引(index)。假設陣列A的元素個數為n，且其索引值為0到n-1之間的數值。

虛擬碼swap x and y的意思是將變數x與變數y的儲存值互換；亦即執行之後變數x的儲存值為執行前變數y的儲存值，執行之後變數y的儲存值為執行前變數x的儲存值。令T(n)為呼叫函式I(A, n-1)的執行時間。T(n)會隨著陣列A所儲存的數值不同而有所不同。

```
S(A, i) {
  If i <= 0, then return;
  S(A, i - 1);
  I(A, i);
  Return; }
```

```
I(A, i) {
  If i <= 0, then return;
```

```
  If A[i] < A[i - 1] {
```

```
    swap A[i] and A[i - 1];
```

```
    I(A, i - 1); }
```

```
  }
```

【中壢】中壢市中山路100號14樓·03-4256899

【台北】台北市開封街一段2號8樓·02-23318268

【台中】台中市東區復興路四段231-3號1樓·04-22298699

【高雄】高雄市新興區中山一路308號8樓·07-2358996

【台南】台南市西區中山路14號3樓之1·06-2235868

【高雄】高雄市新興區中山一路308號8樓·07-2358996

【高雄】另有板橋·淡水·三峽·林口·羅東·逢甲·東海·中技·雲林·彰化·嘉義

Return; }

- (一)請用 O -notation 表示 $T(n)$ 的上界 (upper bound) ; 請用 Ω -notation 表示 $T(n)$ 的下界 (lower bound)。(5分)
- (二)請說明 $T(n)$ 最大時, 程式開始執行前陣列 A 所儲存的數值有何特性? 理由為何? (5分)
- (三)請問函式 $S(A, n-1)$ 的時間複雜度為何? 請說明理由。(5分)
- (四)請問執行函式 $S(A, n-1)$ 後, 陣列 A 儲存的內容有何特性? 請證明你的觀察。(10分)

答:

(一) $T(n)$ 的 upper bound 為 $O(n)$; $T(n)$ 的 lower bound 為 $\Omega(1)$ 。

(二) $A[n-1] < A[0] \sim A[n-2]$ 所有的元素。

(三) $S(A, n-1)$ 的時間為 $O(n^2)$, 因為, worst case 之下, 時間 = $\sum_{k=1}^n T(k) = O(n^2)$ 。

(四)執行之後, A 為一個由小而大遞增排序好的陣列。使用數學歸納法證明如下:

(1)base

$S(A, 0)$ 可以排序好 1 個元素的陣列 (即 $A[0]$)。

(2)hypothesis

$S(A, i-1)$ 可以排序好 $A[0] \sim A[i-1]$ 。

(3)induction

$S(A, i)$ 先進行 $S(A, i-1)$ 可以將 $A[0] \sim A[i-1]$ 排序好, 然後進行 $I(A, i)$ 可以將 $A[i]$ 插入到正確的位置。

三、堆積 (heap) 是一棵完整二元樹 (complete binary tree), 每個節點儲存一個鍵值 (key value), 且每一個內部節點 (internal node) 的鍵值都不比其子節點的鍵值小。

(一)請畫一棵七個節點的堆積, 其節點儲存的鍵值形成的集合為

$\{100, 10, 55, 69, 38, 27, 48\}$ 。(5分)

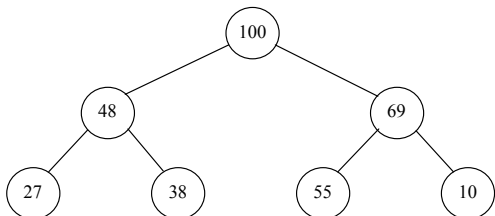
(二)請說明如何利用陣列 (array) 實做一棵 n 個節點的堆積。(5分)

(三)假設一棵 n 個節點的完整二元樹, 其每個節點儲存一個鍵值, 除了根節點 (root) 之外, 其他內部節點的鍵值均不比其子節點的鍵值小。請用虛擬碼描述將這樣的一棵二元樹調整成堆積的演算法。(10分)

(四)請說明如何利用上述演算法將一棵 n 個節點之堆積的根節點儲存的鍵值刪除, 得到一棵儲存其餘 $n-1$ 個鍵值的堆積。(5分)

答:

(一)



(二)Heap 通常使用 binary tree 的 1-D Array representation, 因為 heap 是完全二元樹, 故可以有效降低空間的浪費率。其表示法如下, 於實作時可以根據下面關係式來進行處理:

(a)root 可以置於 $a[1]$ 。

(b) $a[i]$ 的 parent 為 $a[\lfloor \frac{i}{2} \rfloor]$ 。

高點·高上高普特考 goldensun.get.com.tw 台北市開封街一段2號8樓 02-23318268

【台中】台中市東區復興路四段231-3號1樓·04-22298699

【台南】台南市中西區中山路147號3樓之1·06-2235868 【高雄】高雄市新興區中山一路308號8樓·07-2358996

【另有板橋·淡水·三峽·林口·羅東·逢甲·東海·中技·雲林·彰化·嘉義】

(c) $a[i]$ 的 left child 為 $a[2i]$ 。

(d) $a[i]$ 的 right child 為 $a[2i+1]$ 。

(e) $a[i]$ 的 sibling 為 $\begin{cases} a[i+1] & , i \text{ 為偶數} \\ a[i-1] & , i \text{ 為奇數} \end{cases}$ 。

(三) 使用下面遞迴程序

```
Adjust(root)
  if (root is not a leaf)
  {
    找出 root 最大的child x;
    if (x > root) swap x and root;
    Adjust(x);
  }
```

(四) delete root;

將最後一個節點移到 root;

Adjust(root);

四、我們想設計一個動態資料結構儲存數字集合 $S = \{0, 1, 2, \dots, n-1\}$ 的兩兩沒有交集，而且聯集等於 S 的子集合。初始時有 n 個元素，個數為 1 的子集合，分別為 $\{0\}, \{1\}, \dots, \{n-1\}$ 。我們希望這個資料結構可以支援以下兩個功能：

1. $\text{union}(x, y)$: $x, y \in S$ 。 $\text{union}(x, y)$ 將包含 x 的子集合與包含 y 的子集合聯集得到一個新的子集合，原來的子集合不再存在。
 2. $\text{equivalence}(x, y)$: $x, y \in S$ 。 $\text{equivalence}(x, y)$ 判斷 x 與 y 是否屬於同一個子集合，若屬於同一個子集合，則回傳“TRUE”，否則回傳“FALSE”。
- 上述兩個函式必須能夠依任何順序交替執行。

- (一) 請描述一個可以達成上述需求而且 $\text{union}(x, y)$ 與 $\text{equivalence}(x, y)$ 的時間複雜度均為 $O(\log n)$ 的資料結構。(15分)
- (二) 請用虛擬碼描述可以在上述資料結構運作的 $\text{union}(x, y)$ 函式。(5分)
- (三) 請用虛擬碼描述可以在上述資料結構運作的 $\text{equivalence}(x, y)$ 函式。(5分)

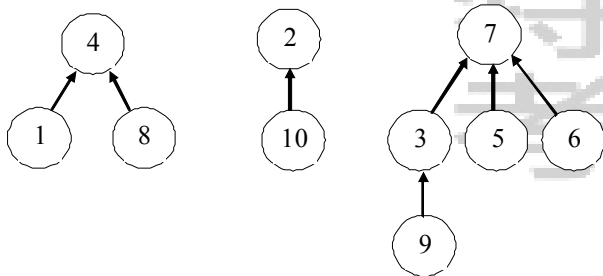
答：

(一) 本題為一群集合的交集，若皆為空集合，則這群集合稱為互斥集合。

例： $A = \{1, 4, 8\}$ $B = \{2, 10\}$ $C = \{3, 5, 6, 7, 9\}$ 為三個互斥集合

互斥集合可以用一般樹來表示，每一個互斥集合用一棵樹來代表。

例：



資料結構

(1) $\text{parent}[i] \geq 0$ ：代表指向父親的指標。

(2) $\text{parent}[i] < 0$ ：表示 node i 為 root，且整棵樹共有一 $|\text{parent}[i]|$ 個 nodes。

做法：Weighting rule 是指在進行 union 運算時，根據兩個 sets 的 elements 個數大小，來選擇那一棵 tree 的樹根來做為整體的樹根，並將節點數較少的樹根的 parent 指標指向較多節點的樹根。

```

(二)
union(x, y)
{
  if (parent[x] <= parent[y])
  {
    parent[x] ← parent[x] + parent[y];
    parent[y] ← x;
  }
  else
  {
    parent[y] ← parent[y] + parent[x];
    parent[x] ← y;
  }
}

```

```

(三) 利用 Find(x) 可以實作 equivalence(x, y)
equivalence(x, y)
{
  return Find(x) == Find(y);
}

```

Find 程序如下：

```

find(x)
{
  i ← x;
  while (parent[i] >= 0)
    i ← parent[i];
  // Collapsing
  j ← x;
  while (parent[j] >= 0)
  {
    k ← parent[j];
    parent[j] ← i;
    j ← k;
  }
  return i;
}

```



高點
·
高上高普特考

高點·高上高普特考 goldensun.get.com.tw 台北市開封街一段2號8樓 02-23318268

【中壢】中壢市中山路100號14樓·03-4256899

【台中】台中市東區復興路四段231-3號1樓·04-22298699

【台南】台南市中西區中山路147號3樓之1·06-2235868

【高雄】高雄市新興區中山一路308號8樓·07-2358996

【另有板橋·淡水·三峽·林口·羅東·逢甲·東海·中技·雲林·彰化·嘉義】