

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。

(三)本科目得以本國文字或英文作答。

一、計算正整數 a 和 b 的最大公因數 $\gcd(a, b)$ 的演算法，以類似 C 語言表示如下：

```
1  integer gcd(a, b) {  
2      x = a; y = b;  
3      while (y > 0) {r = x % y; x = y; y = r;}  
4      return x;  
5  }
```

其中資料型態 integer 表示整數， $x \% y$ 表示 x 除以 y 的餘數。請回答下列問題：（每小題 10 分，共 20 分）

- (一)請證明：輸入任意兩個正整數，此程式執行一定時間後就會停止，不會造成無窮迴圈。
- (二)假設 $a > b$ ，請證明此程式之 **while** 迴圈（第 3 行）至多只會被執行 $2 \log_2 b + 1$ 次。

二、給定一個權重圖（weighted graph）， $G=(V, E, w)$ ，假設 $V = \{1, 2, \dots, n\}$ ，且每個邊（edge） e 的權重 $w(e)$ 都是正整數。令 $l(v)$ 為以 v 為端點的所有邊中權重最小的邊。將這些邊集合起來稱作 L ，也就是 $L = \bigcup_{v \in V} l(v)$ 。

（每小題 5 分，共 20 分）

- (一)假設每個邊的權重都不相同。請證明由 L 中這些邊所構成的子圖（edge induced subgraph） $G[L]$ 沒有迴圈。
- (二) $G[L]$ 是否一定是 G 的擴張樹（spanning tree）？若是請證明之，若不是請給一個反例。
- (三)用以上之結論，設計一個計算 G 的最小權重擴張樹（minimum spanning tree）的演算法。
- (四)在一般的應用中，邊的權重可能會相同，請修正上述之演算法，使修正後之演算法可以正確找出答案。

三、假設陣列 $A[1..n]$ 儲存 n 個正整數 $x_1, x_2, \dots, x_n$ 。(每小題 10 分，共 20 分)

(一) 已知所有的正整數 $x_i \leq M$ 。請設計一個 $O(n + M)$ 時間的演算法將這些整數由小到大排列。

(二) 已知所有的正整數 $x_i \leq n^2$ 。請設計一個 $O(n)$ 時間的演算法將這些整數由小到大排列，或證明這是不可行的。

四、假設有個陣列 $A[1..n]$ 儲存著 n 個整數。可將 $A[1..n]$ 看成二元樹，其中 $A[1]$ 是樹根。 $A[i]$ 的左右子節點分別為 $A[2i]$ 和 $A[2i + 1]$, $i = 1, 2, \dots, n/2$ 。若 $2i > n$ 或 $2i + 1 > n$ ，則這些子節點是不存在的。若 A 滿足 $A[i] \geq \max\{A[2i], A[2i + 1]\}$, $1 \leq i \leq n/2$ ，則稱陣列 $A[1..n]$ 是一個堆疊 (heap)。假設有個副程式 $\text{sift}(A, r, n)$ 其輸入參數 A 是一個陣列， n 是 A 的大小， $r \leq n$ 是一個指標，指向此子樹的樹根。副程式 $\text{sift}(A, r, n)$ 的功能是將 $A[r]$ 為樹根的子樹變成 heap。在呼叫 $\text{sift}(A, r, n)$ 之前，它的左右子樹都已經是 heap。副程式 $\text{sift}(A, r, n)$ 所需的計算時間是 $O(h(r))$ ，其中 $h(r)$ 是以 $A[r]$ 為樹根的子樹的高度，也就是從樹根到任一樹葉的最長距離。

(每小題 10 分，共 20 分)

(一) 用 $\text{sift}(A, r, n)$ 設計一個線性時間的演算法，將陣列 $A[1..n]$ 變成 heap。

(二) 分析以上所設計演算法的計算複雜度為 $O(n)$ 。

五、斐波納契數 (Fibonacci number) F_n 的定義是 $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n > 1$ 。

計算 Fibonacci number F_n 的演算法，以類似 C 語言表示如下：

```

1  integer f[N]; // array of N integers
2  integer F(n) {
3      if (f[n] < 0)
4          f[n] = F(n-1) + F(n-2);
5      return f[n];
6  }
7  integer Fib(n) {
8      f[0] = 0; f[1] = 1;
9      for (i = 2; i ≤ n; i = i + 1)
10         f[i] = -1;
11     return F(n);
12 }
```

其中資料型態 integer 表示整數。假設輸入的整數 $n > 1$ 。主程式執行 $\text{Fib}(n)$ ，則副程式 $F(n)$ 第 4 行之指令：

$f[n] = F(n-1) + F(n-2)$ 會被執行幾次？請說明理由。(20 分)