

# 國立中央大學八十六學年度碩士班研究生入學試題卷

所別: 數學研究所 不分組 科目: 資料結構 共 2 頁 第 / 頁

- 1 (10 points) 考慮以下的虛擬程式 (pseudo code), 請問, 若輸入一正整數  $N$ , 輸出是什麼 (和  $N$  的關係).

```
input integer N
integer x, count
x = 0
count = 0
while x < N do
  x = 2*x
  count = count + 1
end while
write count
return
```

- 2 (10 points) 若  $f(n)$  和  $g(n)$  分別是定義在正整數集合上的函數. 定義大  $O$  符號 (Big-O) 如下

$$f(n) = O(g(n)) \iff \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = c \text{ 且 } 0 < c < \infty.$$

例如  $5n^2 + 6n = O(n^2)$ . 針對下列 (a)-(e) 五小題, 分別在 (A)-(D) 之中選一個相等的量.

(A)  $O(1)$  (B)  $O(n)$  (C)  $O(n^2)$  (D)  $O(n^3)$

(a)  $1 + 2 + \dots + n$

(b)  $\max\{n^3, 1000n\}$

(c)  $\sqrt{n^2 + \frac{1}{2}n}$

(d) 21

(e)  $\frac{1}{2}n + \log n$

- 3 (20 points) 考慮一個矩陣與向量的乘法. 如果我們在 FORTRAN 語言中, 宣告一個  $N$  維的方陣  $A$ , 兩個  $N$  維的序列  $x$  和  $b$ . 事實上, FORTRAN 處理雙參數序列 (double-array) 的方法是所謂的 column-oriented. 也就是說,  $A$  的元素在記憶體中的排列順序如下:

$$A_{11}A_{21}A_{31} \dots A_{N1}A_{12}A_{22} \dots A_{N2}A_{13}A_{23} \dots A_{N3} \dots A_{1N} \dots A_{NN}$$

現在, 針對以下兩個程式片斷 (R) 與 (C). 請

(a) 指出它們有什麼不同?

(b) 哪個比較好? 為什麼?

```
(R)          DO 10 I = 1, N
              b(I) = 0.
              DO 20 J = 1, N
                b(I) = b(I) + A(I,J) * x(J)
              20 CONTINUE
            10 CONTINUE
```

```
(C)          DO 10 J = 1, N
              b(J) = 0.
            10 CONTINUE
              DO 20 J = 1, N
                DO 30 I = 1, N
                  b(I) = b(I) + A(I,J) * x(J)
                30 CONTINUE
              20 CONTINUE
```

# 國立中央大學八十六學年度碩士班研究生入學試題卷

所別: 數學研究所 不分组 科目: 資料結構 共 2 頁 第 2 頁

4 (20 points) 以下是有關 stack 和 queue 的問題.

- (a) 比較 stack 和 queue 在觀念上的不同.
- (b) 分別就以下動作, 敘述 stack 和 queue 在實作 (implementation) 上有何不同.

(b-1) 添加一個元素

(b-2) 取出一個元素

(c) 在以下的應用問題中, 指出應該選用 stack 還是 queue 作為基本的 資料結構.

(c-1) 作業系統中的 printer buffer

(c-2) 在迷宮中尋找出路

(c-3) 執行一個後運算式 (postfix expression)

(c-4) 航空站的起降排班

5 (10 points) 請將下列的普通運算式 (infix expression) 改成

(a) 前運算式 (prefix expression).

(b) 後運算式 (postfix expression).

$$((a + b) + c \times (d + e) + f) \times (g + h)$$

6 (10 points) 依序讀入以下整數

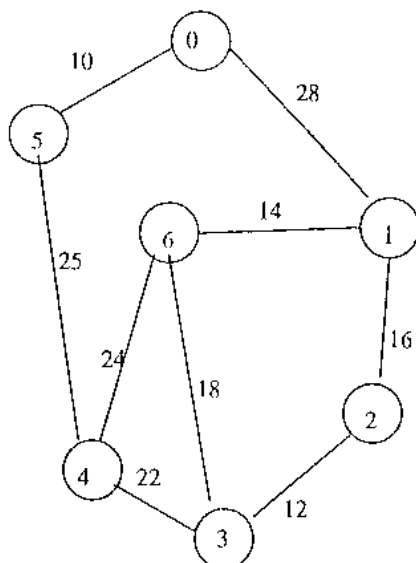
40 20 10 30 55 70 50 45 52

並且以一個 Binary Search Tree 的資料結構儲存; 也就是說, 一個二元樹 (binary tree), 每個元素的左下方元素都比它自己小, 而右下方元素都比它自己大. 請

(a) 畫出這個二元樹.

(b) 這個二元樹的層數是多少? 它是否為一個完整的 (complete) 二元樹?

7 (20 points) 令  $G = (V, E, w)$  是一個加權無方向性圖, 如下.



參考用

(a) 寫出  $G$  的 adjacency matrix with weight.

(b) 畫出  $G$  的 adjacency list with weight.

(c) 定義 minimum cost spanning tree, 並畫出一個  $G$  的 minimum cost spanning tree.