

# 淡江大學 110 學年度日間學制寒假轉學生招生考試試題

系別：資訊管理學系三年級

科目：資料結構

19-1

考試日期：1月19日(星期三) 第2節

本試題共 2 大題， 2 頁

[注意事項]: 1. 作答時，務必清楚標示題號，否則不計分。 2. 請依序作答。

## 一、 名詞解釋(18%): 共 6 題

- 1. 繼承(Inheritance)
- 2. 封裝(Encapsulation)
- 3. 軟體工程(Software Engineering)
- 4. Big-O notation
- 5. 環形佇列(Circular queue)
- 6. 完整二元樹(Complete binary tree)

## 二、 問答題(82%): 共 7 題

1. 依照以下指示，使用你熟悉的程式語言(如 Java/C/C++等)撰寫程式。(注意: 不可使用任何語言之內建函數或 API) (10%)
  - (a) 撰寫一單層迴圈，找出整數陣列 a 中第二大的元素(陣列大小為 n)，步驟如下: (5%)
    - (1) 宣告二個整數變數 max, max2，準備用來儲存陣列中的最大值與次大值。
    - (2) 先令 max, max2 分別儲存 a[0], a[1]之較大與較小的元素
    - (3) 撰寫一迴圈，由 a[2]開始，逐一檢視陣列元素: 先令其與 max 比較，若較大，則取代 max，但使 max2 改儲存原先 max 的值；否則，再令其與 max2 比較，若較大，則取代 max2。
  - (b) 撰寫一雙層迴圈，實作 bubble sort，將一整數陣列 a 的內容由小到大排序(陣列大小為 n)。 (5%)
2. 依照以下指示，使用你熟悉的程式語言(如 Java/C/C++等)撰寫程式。(注意: 本題只能產生一個陣列，且不可使用任何語言之內建函數或 API) (10%)
  - (a) 產生一具有 9 個元素的整數陣列 a，撰寫一迴圈，依序將其內容設定為 10, 20, 30, 40, 50, 0, 0, 0, 0，並宣告一整數 size，令其值為 5。
  - (b) 令整數變數 index=2, n=25，撰寫一迴圈，將 n 插入在陣列指定註標(index)處，使陣列元素逐一向後挪移，內容成為 10, 20, 25, 30, 40, 50, 0, 0, 0，並將 size 增加為 6。
  - (c) 再令 index=3，撰寫一迴圈，在陣列中移除指定註標(index)處的元素，使陣列元素逐一向前挪移，內容成為 10, 20, 25, 40, 50, 0, 0, 0, 0，並將 size 減少為 5。
3. 串列(List): (10%)
  - (a) 請說明串列(List)、堆疊(Stack)與佇列(Queue)在操作上的差異。(3%)
  - (b) 在單向鏈結串列(Singly Linked List)加上一個 dummy head node，有何優點? (3%)
  - (c) 請繪製一個雙向鏈結串列(Doubly Linked List)之資料結構，串列元素為整數，依序為 10, 20, 30, 40。(4%)

# 淡江大學 110 學年度日間學制寒假轉學生招生考試試題

系別：資訊管理學系三年級

科目：資料結構

19-2

考試日期：1 月 19 日(星期三) 第 2 節

本試題共 2 大題， 2 頁

## 4. 堆疊(Stack): (12%)

(a) 已知有二個堆疊  $s_1, s_2$ , 請撰寫虛擬碼(pseudo code)將  $s_1, s_2$  的內容取出, 並依序存於一空堆疊  $s_3$  中。例如, 若  $s_1=[n_1, n_2, n_3], s_2=[m_1, m_2, m_3, m_4]$ , 則  $s_3$  的內容將為  $[n_1, n_2, n_3, m_1, m_2, m_3, m_4]$ 。假設堆疊僅能有以下操作:

$s.push(n)$  將元素  $n$  存於堆疊  $s$  頂端

$x = s.pop()$  移除堆疊  $s$  之頂端元素, 並將其存於  $x$

$s.isEmpty()$  判別  $s$  內容是否為空, 回傳值為 true 或 false (5%)

(b) 請將中置式  $6-(5+3)+(7*8/4)$  轉換為後置式(postfix expression)。(3%)

(c) 承(b), 將所產生的後置式配合堆疊運作, 計算運算式的值。(需寫出過程) (4%)

## 5. 已知有一整數數列 48, 93, 72, 56, 18, 99, 64, 28, (12%)

(a) 請使用 Insertion Sort 對其進行由小到大排序(寫出過程否則不給分)。

(b) 分析 Insertion Sort 之最差狀況時間複雜度, 並以 Big-O 表示法表示之。

(c) 請使用 Merge Sort 對其進行由大到小排序(寫出過程否則不給分)。

(d) 分析 Merge Sort 之最差狀況時間複雜度, 並以 Big-O 表示法表示之。

## 6. 假設一 Hash Table 之大小(TableSize)為 11, 並以 $h(key) = key \% TableSize$ 做為其 Hash function: (14%)

(a) 將鍵值 20,33,10,54,14,23,55 依序加入表格中, 並以 linear probing 做為碰撞排解 (collision resolution)方法, 最後表格的內容為何?(需寫出過程, 否則不給分) (4%)

(b) 若改將鍵值 20,33,44,42,65,13,31 依序加入表格中, 並以 quadratic probing 做為碰撞排解方法, 最後表格的內容為何?(需寫出過程, 否則不給分) (4%)

(c) 解釋使用 linear probing 進行碰撞排解時, 可能產生的 Primary Clustering 問題。(3%)

(d) 當使用 quadratic probing 時, 能否探測到表格中所有可用位置? 說明原因。(3%)

## 7. 二元搜尋樹(Binary Search Tree): (14%)

(a) 依序將鍵值 6, 11, 7, 9, 4, 5, 2, 3, 12, 10, 1, 13, 8 加入一二元搜尋樹, 畫出此樹的結構。(3%)

(b) 承(a), 寫出所建立二元樹之後序巡行(post order traversal)結果。(3%)

(c) 承(a), 若需將此二元搜尋樹的根節點刪除時, 如何透過簡單的節點移動, 使其繼續保持二元搜尋樹的特質? 寫出二種不同作法, 並畫出移除根節點後的樹狀結構。(3%)

(d) 證明一個高度為  $h$  的二元樹, 最多有  $2^h - 1$  個節點。(5%)