

淡江大學 99 學年度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

本試題共 7 大題， 2 頁

本試題雙面印刷

1. 串列(List): (16%)

(a) 完成以下 Array List 之 Java 實作。(10%)

(b) 比較 Array List 與 Linked List 在 add(index, item), remove(index)與 get(index)等三項操作上的優缺點。(請分三項條列作答，否則不給分)。(6%)

```

class ArrayList {
    int[] items = new int[50]; // 容量為 50 的 Array List
    int cnt = 0; // 記錄串列中的元素個數
    int size() { return cnt; }
    public void add(int item) { // 將 item 附加在串列的末尾
        add(cnt, item);
    }
    public void add(int index, int item) {
        // 將元素 item 插入串列中註標為 index 之位置
        // index 由 0 算起
        if (index >= 0 && index <= cnt) {
            for (int i = cnt; i > ①; i--) ②;
            items[ ③ ] = item;
            cnt++;
        }
    }
    public void remove(int index) {
        // 刪除串列中註標 index 處的元素
        if (index >= 0 && index < cnt) {
            for (int i = index; i < ④; i++) ⑤;
            cnt--;
        }
    }
    public int get(int index) {
        if (index >= 0 && index < cnt) return items[index];
        return Integer.MIN_VALUE; // report failure
    }
}

```

2. 堆疊(Stack): (17%)

(a) 完成以下整數堆疊之 Java 實作。(8%)

(b) 參考以下某算術運算式之後置式(postfix expression)，求出其運算結果。(4%)

5 3 5 8 + 4 - * 6 % - (- 為減號, % 為求餘數運算符號)

(c) 將以下中置式(infix expression)轉換為後置式。(5%)

e+(a+b*c/d)/(a*b*(c+d)-e)

背面尚有試題

```

class IntegerStack {
    public static final int MAX_SIZE = 5;
    int[] items = new int[MAX_SIZE];
    int top = -1;

    public int size() { return top+1; }
    public boolean isFull() { return ①; }
    public boolean isEmpty() { return ②; }
    public void push(int item){
        if (isFull()) ③;
    }
    public int pop() {
        if (isEmpty()) return ④;
        return Integer.MIN_VALUE; // report failure
    }
}

```

3. 優先佇列(Priority Queue): (12%)

(a) 將以下鍵值依序加入一 MaxHeap，畫出逐步加入的過程。(6%)

18, 5, 12, 23, 8, 30, 15, 20

(b) 在堆積(heap)中，某節點的註標(index)為 k ，則其父節點與二個子節點之註標分別為何?(6%)

4. 二元搜尋樹: (12%)

(a) 依序將以下整數加入一棵空的二元搜尋樹，畫出此樹最後的外觀 (4%)

18, 9, 13, 10, 25, 20, 33, 22, 21

(b) 寫出此樹的中序(inorder)與後序(postorder)巡行的次序。(4%)

(c) 若用 Heap 表示此二元樹，共需耗用多少位置? 說明你的答案。(4%)

5. 二元樹: (12%)

(a) 完整二元樹(complete binary tree)與全滿二元樹(full binary tree)有何不同? 舉例說明之。(4%)

(b) 證明高度為 h 的全滿二元樹，共有 $2^h - 1$ 個節點。(4%)

(c) 一棵具有 n 個節點($n \geq 1$)的二元樹，存在多少個空的指標欄位? 請說明原因。(4%)

6. 時間複雜度分析(15%)

(a) 分析二元搜尋法(binary search)之最差狀況時間複雜度。

(b) 分析 Insertion Sort 的最差狀況時間複雜度。

(c) 分析 Quick Sort 的最差狀況時間複雜度。

7. 雜湊(Hashing): (16%)

(a) 將以下鍵值(key)依序加入一雜湊表(表格大小為 13)，使用 $h(\text{key}) = \text{key} \% \text{TableSize}$ 做為雜湊函數，並採用二次探測(quadratic probing)做為碰撞排解方法，畫出最後的雜湊表內容。(寫出計算過程，否則不給分) (8%)

25, 16, 21, 142, 30, 43, 12, 95

(b) 承(a)，改用 separate chaining 做為碰撞排解方法。(4%)

(c) 說明雙雜湊(double hashing)如何改善雜湊表中的群集(cluster)效應。(4%)