

106 年公務人員特種考試關務人員考試試題

考試別：關務人員考試

等 別：三等考試

類 科：資訊處理

科 目：資料結構

一、一個二元搜尋數 (binary search tree) 初始為空的，依序插入 (insert) 5, 11, 9, 24, 10, 2, 15, 3。

(一) 請繪出完成輸入後的二元搜尋樹。

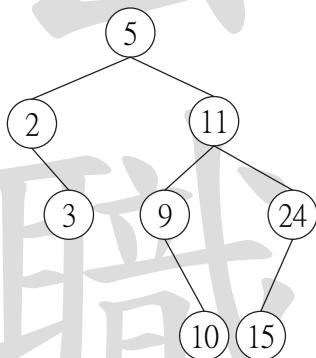
(二) 試說明如何利用一維陣列來表示 (represent) 此二元搜尋樹，並在此一維陣列中保有此樹狀結構父節點與子節點的關係性。

(三) 請設計一演算法能將此二元搜尋樹，依數值由大到小的方式輸出。

(四) 對(一)產生的二元搜尋樹，刪除數值 5。請繪出完成刪除動作後的二元搜尋樹。

擬答：

(一)



(二) 利用一個大小為 15 的一維陣列，編號由 1 開始，依序存放此二元搜尋樹。

先初始化此一維陣列之值皆為 -1，然後由 5 開始放入索引 1，依序存放，若缺少左右子樹，則陣列對應位置亦空著。

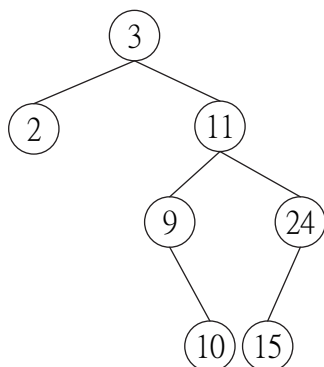
5	2	11	-1	3	9	24	-1	-1	-1	-1	-1	10	15	-1
---	---	----	----	---	---	----	----	----	----	----	----	----	----	----

父節點為 n ，左子節點為 $2n$ ，右子節點為 $2n+1$

(三) 以右子樹為優先輸出的中序追蹤可以滿足由大到小輸出。

```
inorder(tree_pointer root)
{
    if (root){
        inorder(root->rchild);
        printf("%d ", root->data);
        inorder(root->lchild);
    }
}
```

(四) 刪掉 5，由左子樹中之最大值取代，也就是由 3 取代



公職王歷屆試題 (106 關務特考)

二、(一)請使用 C 或 Java 語言寫一副程式 void FindMinMax (int [] A, int n, int Min, int Max) , 對一個未排序的 (unsorted) 且長度為 n 的陣列 $A[0:n-1]$, 尋找陣列中的只花費 $n-1$ 次的數值比較運算 (comparison) 。

(二)請舉例說明此副程式最差情況 (worst case) 所花費的數值比較運算 (comparison) 次數。

擬答：

(一)

```
void FindMinMax(int A[],int n,int &min,int &max)
{
    max=-INF;
    min=INF;
    for(int i=0;i<n-1;i++)
    {
        if(A[i]<A[i+1])
        {
            if(A[i+1]>max)
                max=A[i+1];
            if(A[i]<min)
                min=A[i];
        }
        else
        {
            if(A[i]>max)
                max=A[i];
            if(A[i+1]<min)
                min=A[i+1];
        }
    }
}
```

(二)基於最大的數和最小的數不會是同一個數，因此可以把數組分成兩部分，然後再從這兩部分中分別找出最大的數和最小的數，直接用兩個變量 Max 和 Min 來存儲當前的最大值和最小值。同一組的兩個數比較之後，不再調整順序，而是用較小者與當前的 Min 比較，如果該數小於當前 Min，則更新 Min。Max 同理。該演算法的最多比較次數為 $1.5N$ 。

三、一個工廠有 n 台機器 M_1, M_2, \dots, M_n 及 k 份工作 J_1, J_2, \dots, J_k ，每份工作都有其所需的執行時間 $T(J_1), T(J_2), \dots, T(J_k)$ 。每一台機器一次只能執行一份工作，每份工作只能交給一台機器執行， n 台機器可同時執行 n 份不同的工作。

(一)請設計一個 Greedy (貪婪) 的演算法，來解決工作排程的問題，使得完成 k 份工作的時間最短。

(二)此 Greedy 演算法適合使用何種資料結構來完成？

(三)此 Greedy 演算法的解法是否能保證為最佳解？請舉例說明。

擬答：

貪婪演算：設定一個特定的選擇規範，稱為貪婪準則，以便在每一個步驟中做出目前看起來最好選擇，即局部最佳解。一旦做出了選擇，就不再更改，並希望這樣的選擇可以得到全域的最佳解。在一般情況下，其結果大多是非常接近最佳解。是有效率的方法，但是並不保證永遠可得到(全域)最佳解。

(一)因題目未交代：

1. k 與 n 的關係，假設 $k \geq n$ 情況下；

2. 每件工作開始時間是否一致，假設每件工作開始時間不一樣，因此會有重疊問題，也就

公職王歷屆試題 (106 關務特考)

是兩個工作的時間區間有重疊。

基於以上的假設，貪婪準則為每次分配一件工作，而且按照每件工作的開始時間為次序來進行工作分配。而選擇機器的準則則根據欲分配工作的開始時間，若此時有舊的機器可用，則將工作分配給舊的機器，否則就將任務分配給一台新的機器。

演算法如下：

步驟 1：先依照工作的開始時間對每個工作做排序，開始時間較早的為優先。

步驟 2：依次取出每一件工作

2.1 若有舊機器登記的執行時間在此工作的開始時間之前，則將工作分配給舊的機器，並更新該機器的執行時間為此工作的結束時間

2.2 否則就將任務分配給一台新的機器，並更新該機器的執行時間

(二)可使用陣列儲存每台機器執行工作的開始或時間

(三)不保證可以得到最佳解，因為當一台機器執行工作之後，若有時間更短的工作進來，此台機器無法停止目前工作而去執行其他更短的工作。這是貪婪演算法的特性，一旦做出了選擇，就不再更改。

四、有一雜湊表格 (hash table) 包含 11 個桶 (buckets)，位址編號由 0 至 10，每個桶有一個槽 (slot)。雜湊函數 h 的定義為 $h(\text{key}) = \text{key} \% 11$ (註： $a \% b$ 表示 a 除以 b 的餘數)。當有碰撞 (collision) 發生時，採用線性探測 (linear probing) 解決碰撞問題。從空的雜湊表格開始，依序加入 10 個整數 5, 51, 23, 68, 12, 36, 6, 30, 32, 10。

(一)請繪出加入 10 個整數後的雜湊表格。

(二)欲在此雜湊表格中尋找資料值 35，請說明須經過幾次的資料值比對，才能確定資料值 35 不在此雜湊表格中。

擬答：

(一)

$$h(\text{key}) = \text{key} \% 11$$

$$5 \% 11 = 5$$

$$51 \% 11 = 7$$

$$23 \% 11 = 1$$

$$68 \% 11 = 2$$

$$12 \% 11 = 1 \text{ (碰撞)}, \text{找尋下一個 } 2 \text{ (碰撞)}, \text{找尋下一個 } 3$$

$$36 \% 11 = 3 \text{ (碰撞)}, \text{找尋下一個 } 4$$

$$6 \% 11 = 6$$

$$30 \% 11 = 8$$

$$32 \% 11 = 10$$

$$10 \% 11 = 10 \text{ (碰撞)}, \text{找尋下一個 } 0$$

雜湊表格內容

0	1	2	3	4	5	6	7	8	9	10
10	23	68	12	36	5	6	51	30		32

(二) $35 \% 11 = 2$ (比對失敗)，找尋下一個 3 (比對失敗)，找尋下一個 4 (比對失敗)，找尋下一個 5 (比對失敗)，找尋下一個 6 (比對失敗)，找尋下一個 7 (比對失敗)，找尋下一個 8 (比對失敗)，找尋下一個 9 (沒有值)

共比對 7 次