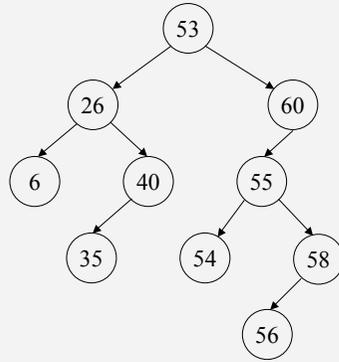


精選例題 47

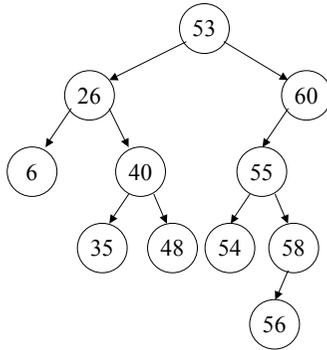
圖一為一個二元搜尋樹 (binary search tree)，每個節點含有一個整數。

- (1) 請將48加入圖一，並將結果的二元搜尋樹畫出。
- (2) 請將53從圖一刪除。假設每個數刪除後，皆由小於但最接近的數取代。請將結果的二元搜尋樹畫出。
- (3) 請將圖一以前序 (preorder) 方式表示。
- (4) 請將圖一以後序 (postorder) 方式表示。

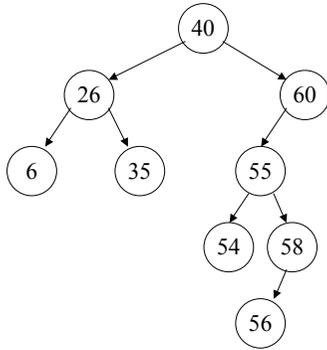


98司法人員

解答：(1)



(2)



(3) 53,26,6,40,35,60,55,54,58,56

(4) 6,35,40,26,54,56,58,55,60,53

精選例題 48

以 A、B、C、D 四個英文字母可以造出幾棵不同的二元搜尋樹 (binary search tree)? 注意：四個英文字母要同時出現在同一棵樹中。

90 中央暨地方機關公務人員升等

解答： $\frac{1}{n+1} \binom{2n}{n} = \frac{1}{5} \times \frac{8!}{4! \times 4!} = 14$ 棵。

精選例題 49

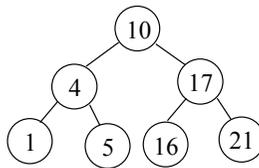
給定鍵值(key)1,4,5,10,16,17,21，請繪出：

(1) 所有高度(Height)為2的二元樹(Binary search tree)。

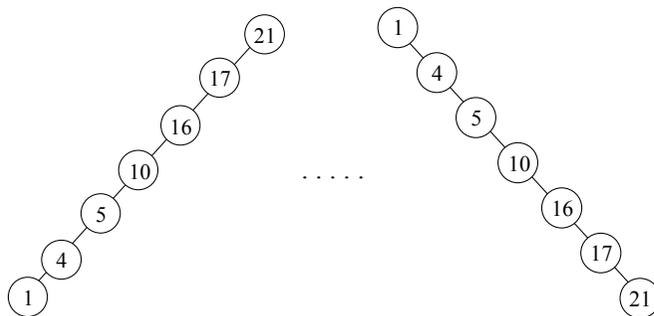
(2) 所有高度為6的二元搜尋樹。

85 基層人員

解答：(1) 一棵



(2) 共有64棵

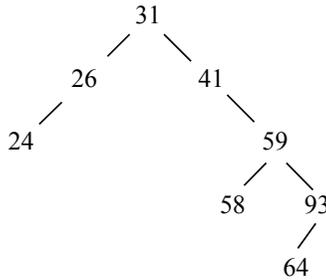


解析：本題樹根為level = 0起算。

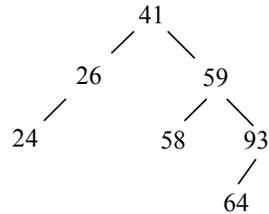
精選例題 50

- (1) 請將下列資料插入二元搜尋樹：31, 41, 59, 26, 58, 93, 24, 64。
- (2) 求出此樹的成功搜尋(successful search)的平均搜尋時間？
- (3) 由所建的樹中將 31 刪除掉，並請說明你的方法。
- (4) 試寫出一個程序以遞增的順序輸出二元搜尋樹的各節點。

解答：(1)



- (2) 搜尋時的平均比較次數 = $\frac{1}{8}(1 \times 1 + 2 \times 2 + 3 \times 2 + 4 \times 2 + 5 \times 1) = 3$ 。
- (3) 刪除一個節點 x 時，須視不同狀況來進行：
 - ① 若 x 為樹葉，則直接刪除即可。
 - ② 若 x 的 $\text{degree}=1$ 時，可以取其子節點來替代 x 。
 - ③ 若 x 的 $\text{degree}=2$ 時，先找出其 $\text{inorder predecessor}$ 或 successor (假設為 y)，將 y 的資料拷貝到 x ，並將問題轉換成爲刪除 y ，繼續處理下去。
- (4) 使用 binary tree 的 inorder traversal 即可得到由小而大的輸出順序。

**精選例題 51**

請回答下列問題：

- (1) 何謂二元搜尋樹(Binary Search Tree)？
- (2) 用二元搜尋樹去表示 n 個元素時，最小高度及最大高度的二元搜尋樹 ($\text{height of binary search tree}$) 其值分別爲何？
- (3) 設計一個演算法(algorithm)，當輸入 n 個元素時，能建立一最小高度的二元搜尋樹。

6-50 資料結構

解答：(1) 二元搜尋樹是一棵二元樹，可以是空的；若非空的二元樹，則樹根的左子樹中的資料皆小於樹根；樹根的右子樹中的資料皆大於樹根；且左、右子樹亦皆為二元搜尋樹。

(2) 最小為 $\lfloor \log_2 n \rfloor + 1$ ；最大為 n 。

(3) 先對 n 個元素先進行由小而大的排序，並存於陣列 a 中，然後使用下面各個擊破法(divide-and-conquer)程序來建立二元搜尋樹。

```
1.   treeptr BuildTree(int a[], int low, int high)
2.   {   if (low>high) return NULL;
3.       else
4.       {
5.           int mid=(low+high)/2;
6.           treeptr p=(treeptr)malloc(sizeof(struct treenode));
7.           p->data=a[mid];
8.           p->left=BuildTree(a,low,mid-1);
9.           p->right=BuildTree(a,mid+1,high);
10.          return p;
11.      }
12. }
```

精選例題 52

請回答下列各序列(sequence)可否構成二元搜尋樹(binary search tree)搜尋鍵值 363 的合法順序。

- (1) 2,252,401,398,330,344,397,363
- (2) 924,220,911,244,898,258,362,363
- (3) 925,202,911,240,912,245,363
- (4) 2,399,387,219,266,382,381,278,363
- (5) 935,278,347,621,299,392,358,363

86 升等

解答：(1) 合法。

(2) 合法。

- (3) 不合法，912 在 911 的左子樹中，但卻大於 911，故有錯。
- (4) 合法。
- (5) 不合法，299 在 347 的右子樹中，但卻小於 347，故有錯。

精選例題 53

- (1) 何謂二元搜尋樹(binary search tree)？
- (2) 試寫一程式(演算法)測試一兩元樹(binary tree)是否為一二元搜尋樹。該樹節點之設計如下：

```

type pointer=^TreeNode;
  TreeNode=record
    data:integer;
    lchild, rchild:pointer;
  end

```

89 交通事業港務人員升資

解答：(1) 二元搜尋樹是一棵二元樹，樹根的左子樹中的後代皆小於樹根，樹根的右子樹中的後代皆大於樹根。而且左、右子樹也都是二元搜尋樹。

- (2) 程式如下：

```

1. function BST_Check(t:pointer; low, high:integer):Boolean;
2. begin
3.   if t=Nil then BST_Check:=true
4.   else if t^.data<low or t^.data>high then BST_Check:=false
5.   else BST_Check:=BST_Check(t^.lchild,low,t^.data)
6.           and BST_Check(t^.rchild,t^.data,high)
7. end;
主程式： result:=BST_Check(root,-∞,∞);

```

精選例題 54

- (1) 何謂二元搜尋樹(Binary Search Tree)？
- (2) 將多筆資料建成二元搜尋樹有何優點？

88 交通事業鐵路人員

解答：(1) 二元搜尋樹是一棵二元樹，樹根的左子樹中的後代皆小於樹