

HW8

非遞迴版前序追蹤

班級:資訊三丙

姓名:楊耀寧

學號:D0745765

題目定義

1.資料結構

本次程式使用的資料結構為含有結構指標的結構串列 (Struct Linked List)。結構中包含的值有 Key、Left 指標、Right 指標。

以此結構來做出在二元樹上的節點，並且每一個在結構中的變數分別代表為，Key 為節點的鍵值，Left 為左子節點的指標變數，Right 為右子節點指標變數。除此之外，在進行前序追蹤演算法時，會用到 Stack 來幫助演算法達成目的。因此，我們也建立了 Stack 資料結構。

2.演算法

本次程式中的主要演算法為「非遞迴版的前序追蹤」演算法，程式碼如下圖。

```
109
110 // preorder print 前序追蹤
111 void preorder(treePtr node){
112     while(1){
113         for(;node;node=node->left){
114             printf("%d ", node->key);
115             if(node->right){
116                 push(node->right);
117             }
118         }
119         node = pop();
120         if(!node){
121             break;
122         }
123     }
124 }
```

原始程式碼

```
1
2 #include <stdlib.h>
3 #include <stdio.h>
4 #define MAX_SIZE 50
5
6 // tree struct
7 struct tree{
8     int key;
9     struct tree *left;
10    struct tree *right;
11 };
12 typedef struct tree* treePtr;
13
14 // root node
15 treePtr root;
16
17 // stack
18 treePtr stack[MAX_SIZE];
19 int top = -1;
20
21 /*
22  tree Operation [start]
```

```
22  tree Operation [start]
23  */
24
25  // 新增節點
26 treePtr newNode(int n){
27     treePtr add = (treePtr)malloc(sizeof(struct tree));
28
29     if(add){
30         add->key = n;
31         add->left = NULL;
32         add->right = NULL;
33         return add;
34     }else{
35         return NULL;
36     }
37 }
38
39 // 插入節點至樹中
40 void add(int n){
41     treePtr ptr = root;
42     treePtr add = newNode(n);
```

```
43
44     if(root == NULL){
45         root = add;
46     }else{
47         while(1){
48             if(add->key > ptr->key){
49                 // 比較大
50                 if(ptr->right != NULL){
51                     ptr = ptr->right;
52                 }else{
53                     ptr->right = add;
54                     break;
55                 }
56             }else if(add->key < ptr->key){
57                 // 比較小
58                 if(ptr->left != NULL){
59                     ptr = ptr->left;
60                 }else{
61                     ptr->left = add;
62                     break;
63                 }
64             }else if(add->key == ptr->key){
```




```

~ Errors: 0
~ Warnings: 0
~ Output Filename: C:\Users\User\Desktop\HW8.exe
~ Output size: 130,118,164,025 KiB
~ Compilation Time: 5.38s

```

執行結果

```
C:\Users\User\Desktop\HW8.exe
[1]input node [2]preorder: 1
num:10
[1]input node [2]preorder: 1
num:4
[1]input node [2]preorder: 1
num:3
[1]input node [2]preorder: 1
num:1
[1]input node [2]preorder: 1
num:6
[1]input node [2]preorder: 1
num:15
[1]input node [2]preorder: 1
num:13
[1]input node [2]preorder: 1
num:14
[1]input node [2]preorder: 1
num:22
[1]input node [2]preorder: 2
10 4 3 1 6 15 13 14 22
[1]input node [2]preorder: S_
```

```
C:\Users\User\Desktop\HW8.exe
[1]input node [2]preorder: 1
num:33
[1]input node [2]preorder: 1
num:19
[1]input node [2]preorder: 1
num:2
[1]input node [2]preorder: 1
num:25
[1]input node [2]preorder: 1
num:24
[1]input node [2]preorder: 1
num:21
[1]input node [2]preorder: 1
num:26
[1]input node [2]preorder: 1
num:27
[1]input node [2]preorder: 2
33 19 2 25 24 21 26 27
[1]input node [2]preorder:
```

時間複雜度

假設樹中的節點個數為 n 。此程式的時間複雜度在 preorder 副程式中，可以發現在 preorder 副程式中，因每一個節點都會被執行到，因此對於 preorder 來說，時間複雜度為 $O(n)$ 。

因此可以判定，此程式的時間複雜度為： $O(n)$

心得

這次的功課蠻有趣的，一般來說遞迴對於我來說，邏輯實作上都會比迭代還來得難思考、撰寫程式碼，結果在二元樹的範疇，竟然遞迴才是最好理解的，反而要用迭代來處理還更加困難。不過好在也解決困難了。對於樹的領域，我認為我還有很多不理解、不熟悉的地方，看來我是得找個時間好好的再熟悉一下樹的各種特性與定義。