

第十六週

王子銓, 陳毅軒, 吳尚龍

電機通訊程式設計

June 3, 2024

Outline

- 1 Tree
- 2 Binary Tree
- 3 作業加分

Tree

Tree(樹) 的定義以及規則在正課中的講義已完整敘述，實習課不會再加贅述，以下會解釋一些撰寫本週題目的方法。

建立 Tree

一般的 Tree 不太會在建的時候指定誰是 root，會以圖的方式儲存，在 DFS 或 BFS 才會指定要以哪個節點為根遍歷。

我們對每一個節點，使用 linked list 建立鄰邊節點，在 struct 那週有提到 adjacency matrix 跟 adjacency list，這裡使用的就是用 linked list 建立 adjacency list，可以讓節點數多的情況下不會 MLE。

```
1 typedef struct ListNode {
2     int val;
3     struct ListNode *nxt;
4 } ListNode;
5 typedef struct {
6     ListNode *head;    // 子節點的鏈表頭指針
7 } Treenode;
8
9 int main() {
10     Treenode tree[n];
11     AddChild(tree, u, v);
12 }
```

Binary Tree

Binary Tree(二元樹) 的定義以及規則在正課中的講義以完整敘述，實習課不會再多加贅述，以下會解釋一些撰寫的方法。

建立 Binary Tree

```
1 // 定義一個二叉樹節點結構
2 typedef struct TreeNode {
3     int data; // 節點儲存的數據
4     struct TreeNode *left; // 指向左子節點的指針
5     struct TreeNode *right; // 指向右子節點的指針
6 } TreeNode;
```


遍歷 Binary Tree

```
1 void preorder(TreeNode *root) { // 中-左-右
2     if (root == NULL) return;
3     printf("%d ", root->data);
4     preorder(root->left);
5     preorder(root->right);
6 }
7 void inorder(TreeNode *root) { // 左-中-右
8     if (root == NULL) return;
9     inorder(root->left);
10    printf("%d ", root->data);
11    inorder(root->right);
12 }
13 void postorder(TreeNode *root) { // 左-右-中
14     if (root == NULL) return;
15     postorder(root->left);
16     postorder(root->right);
17     printf("%d ", root->data);
18 }
```

作業加分

作業加分

因為本週沒辦法 demo，所以只要作業有 100 分，總成績就加 1 分。