

# 数据结构 hw8

刘良宇 PB20000180

## 6.68:

已知一棵树的由根至叶子结点按层次输入的结点序列及每个结点的度（每层中自左至右输入），试写出构造此树的孩子-兄弟链表的算法。

```
// 约定传入参数为结点数组和度数数组，现在的要求是把它们合理的连接起来。
// 这里假设结点数组最后一项为空指针(重要!)
int treeInit(Tree trees[], int degrees[]) {
    if (!trees[0])
        return 1;
    trees[0]->nextSibling = nullptr;
    int j = 0;
    for (int i = 0; trees[i]; i++, j += degrees[i]) {
        trees[i]->firstChild = degrees[i] ? trees[j + 1] : nullptr;
        // 这个结点的孩子从 j + 1 到 j + degrees[i]
        for (int k = j + 1; k < j + degrees[i]; k++)
            trees[k]->nextSibling = trees[k + 1];
        trees[j + degrees[i]]->nextSibling = nullptr;
    }
    return 0;
}
```

## 6.71:

假设树上每个结点所含的数据元素为一个字母，并且以孩子-兄弟链表为树的存储结构，试写一个按凹入表方式打印一棵树的算法。

```
void showTree(Tree tree, int depth = 0) {    // 默认参数为 0
    if (!tree)
        return;
    for (int i = 0; i < depth; i++) {
        putchar(' ');
    }
    putchar(tree->data);
    putchar('\n');
    // 接下来遍历所有孩子
    for (Tree t = tree->firstChild; t; t = t->nextSibling) {
        showTree(t, depth + 1);
    }
}
```