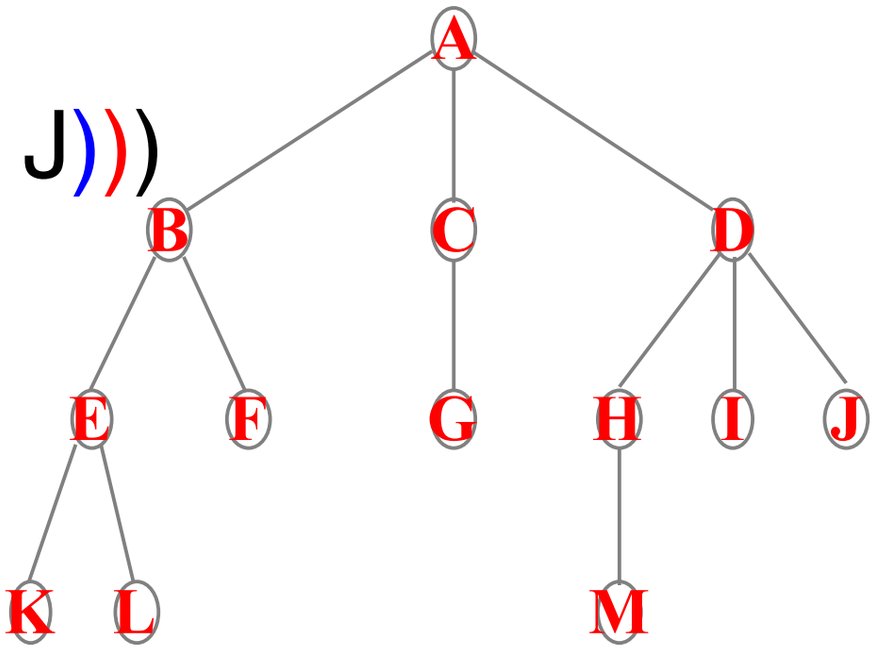
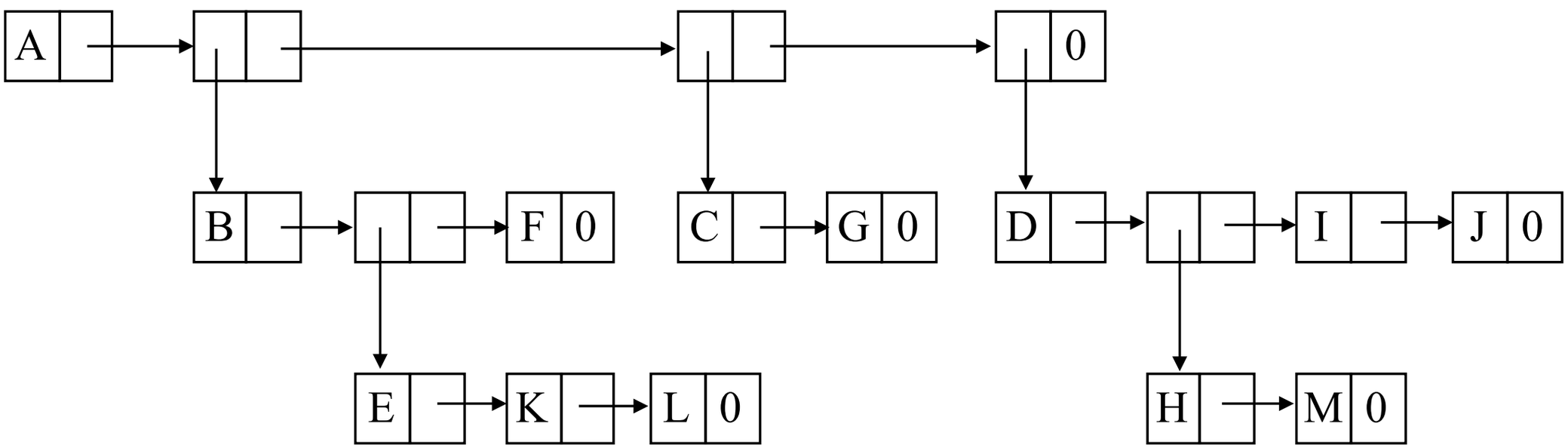
# 資料結構實習

12月8號

# Tree

The tree is represented as a list:
(A (B (E (K, L), F), C(G), D(H (M), I, J)))

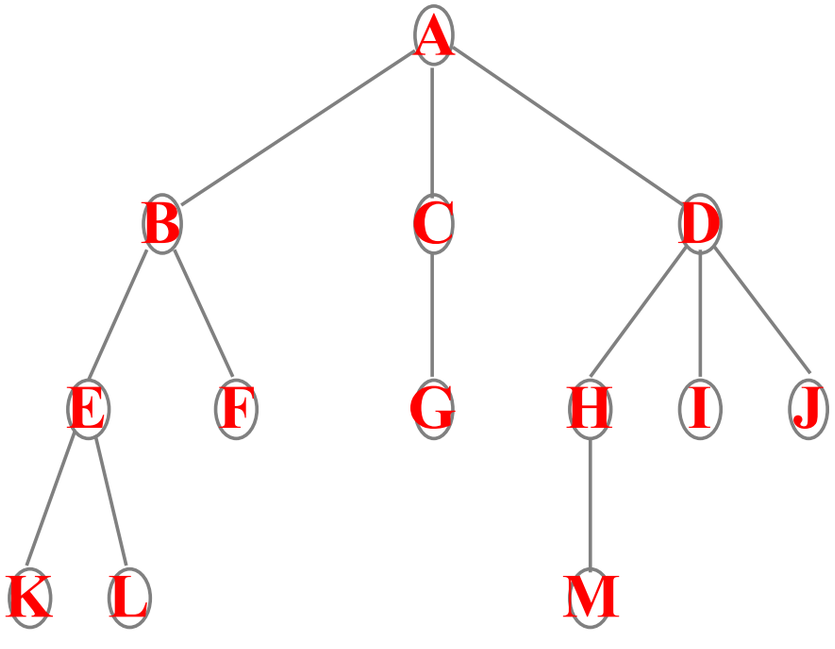


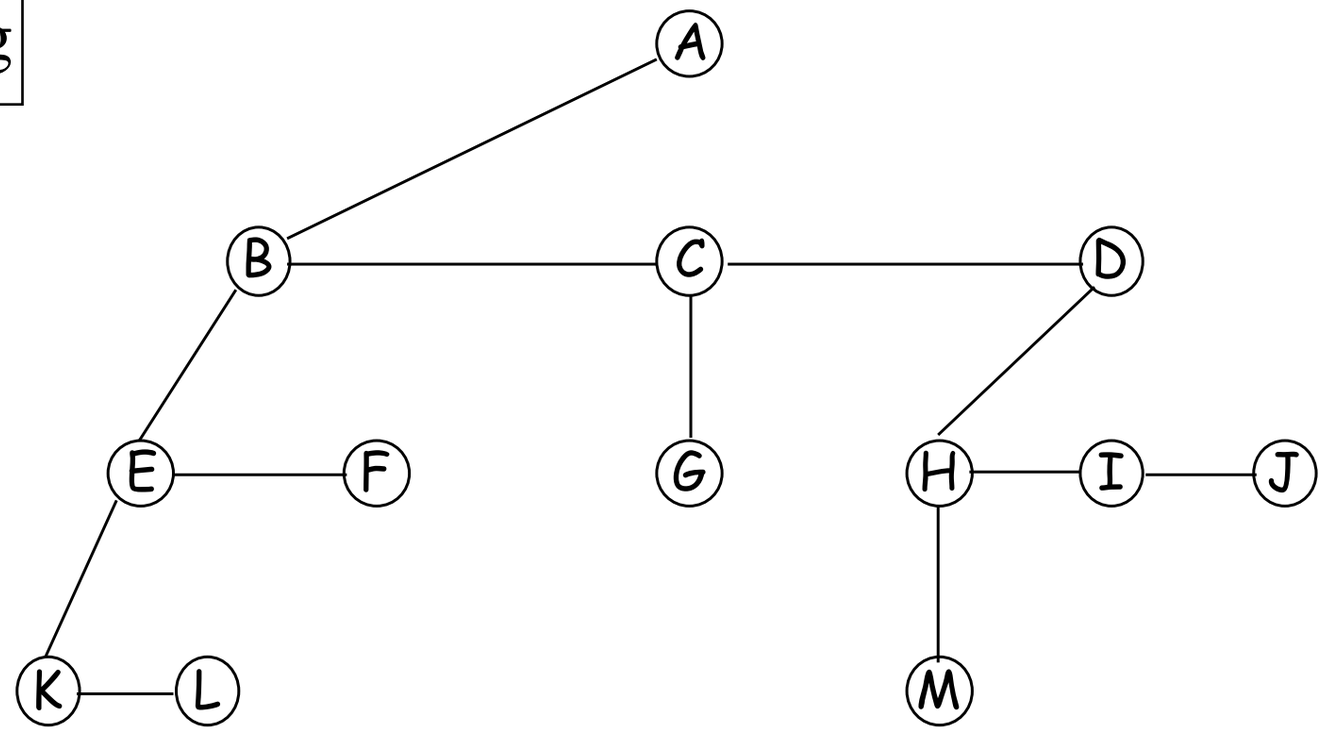



# Tree

- Left child-right sibling tree
  - –two links (or pointers): left child and right sibling

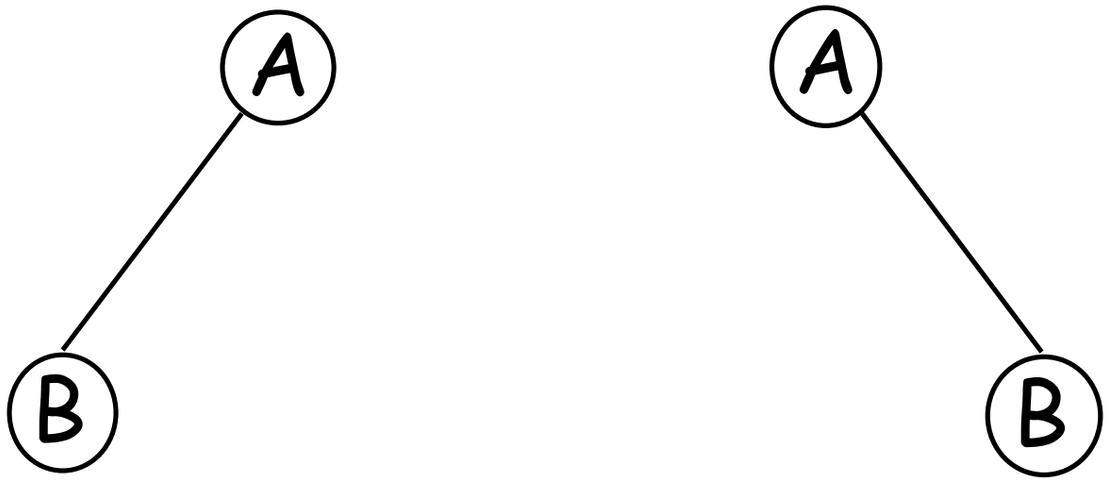
| data | |
|---|---|
| left child | right sibling |



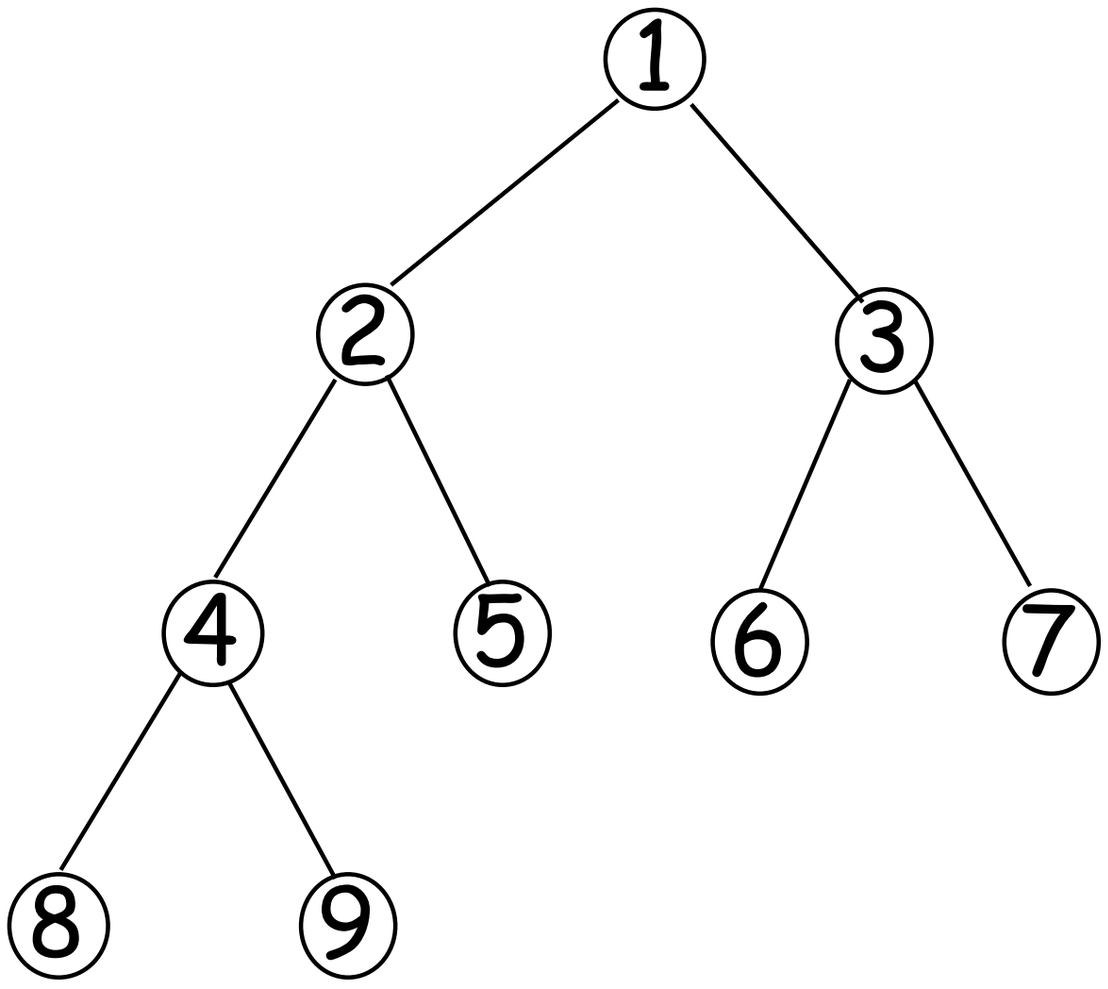
# Binary Tree

- A binary tree:
  - a finite set of nodes that is either empty, or consists of a root and two disjoint binary trees called the left subtree and the right subtree.
- In a binary tree, we distinguish between the order of the children; in a tree we do not.



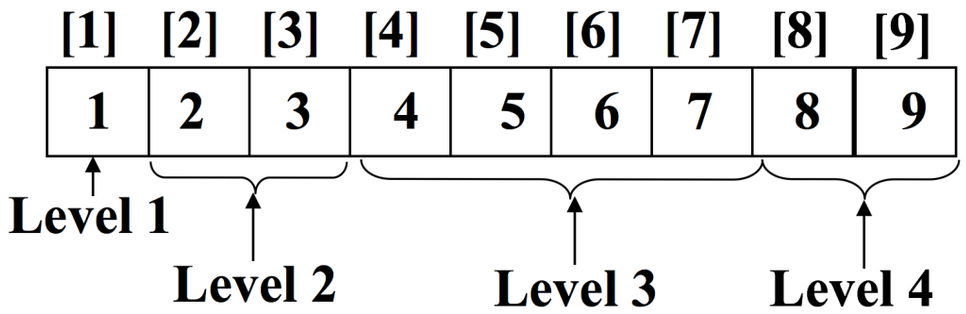

Two different binary trees

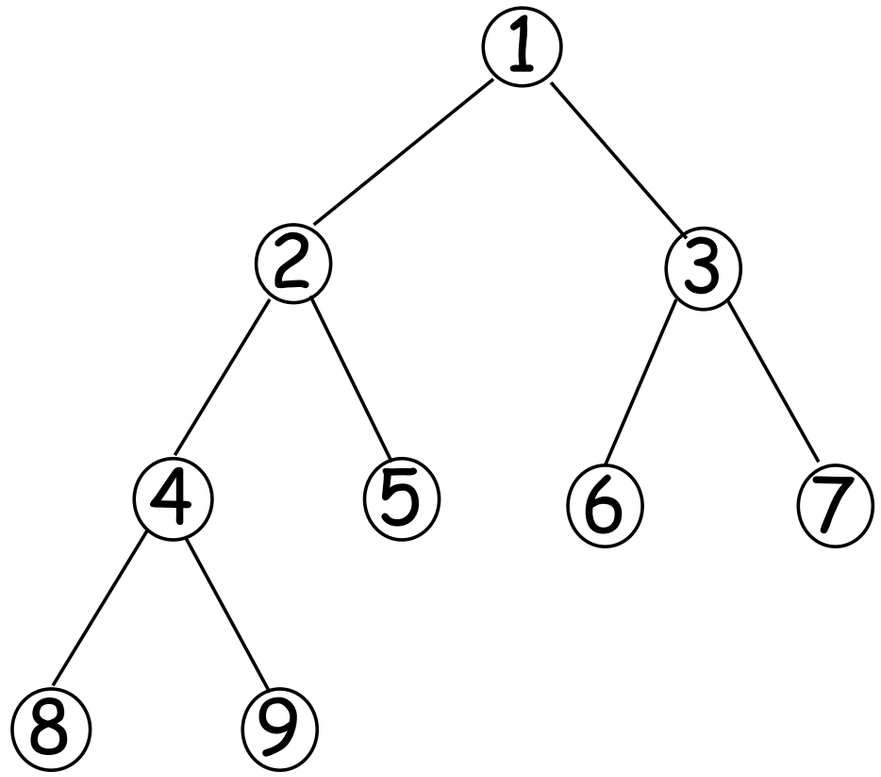
# Complete Binary Tree



# Complete Binary Tree

- It can be represented by an array.



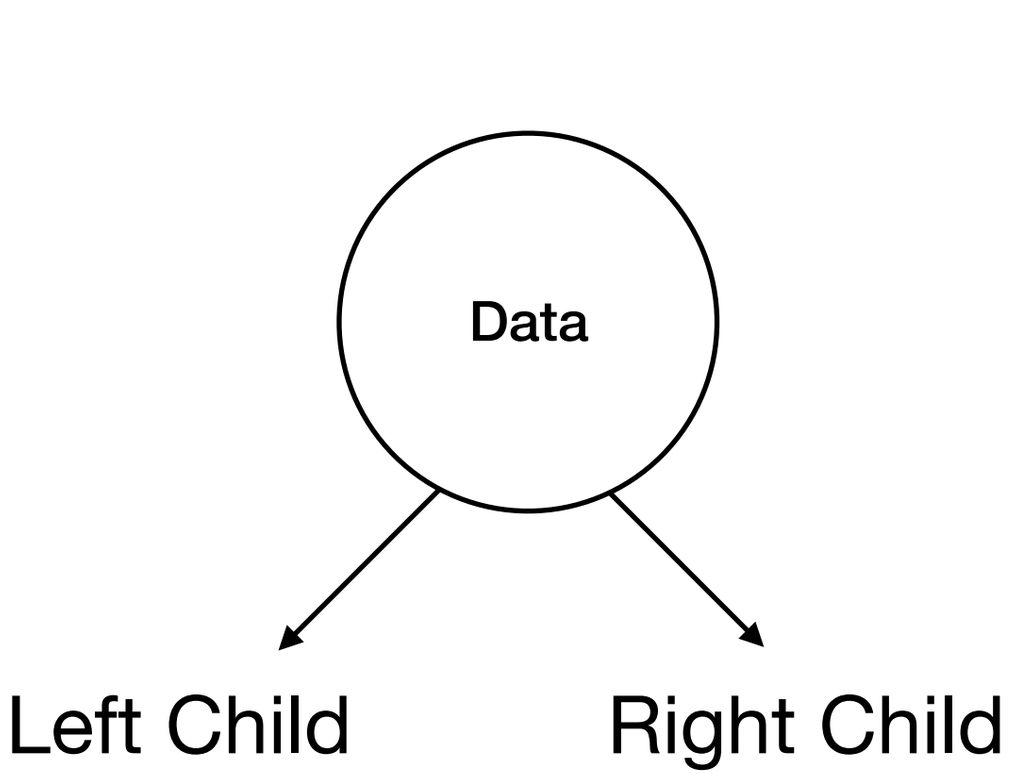



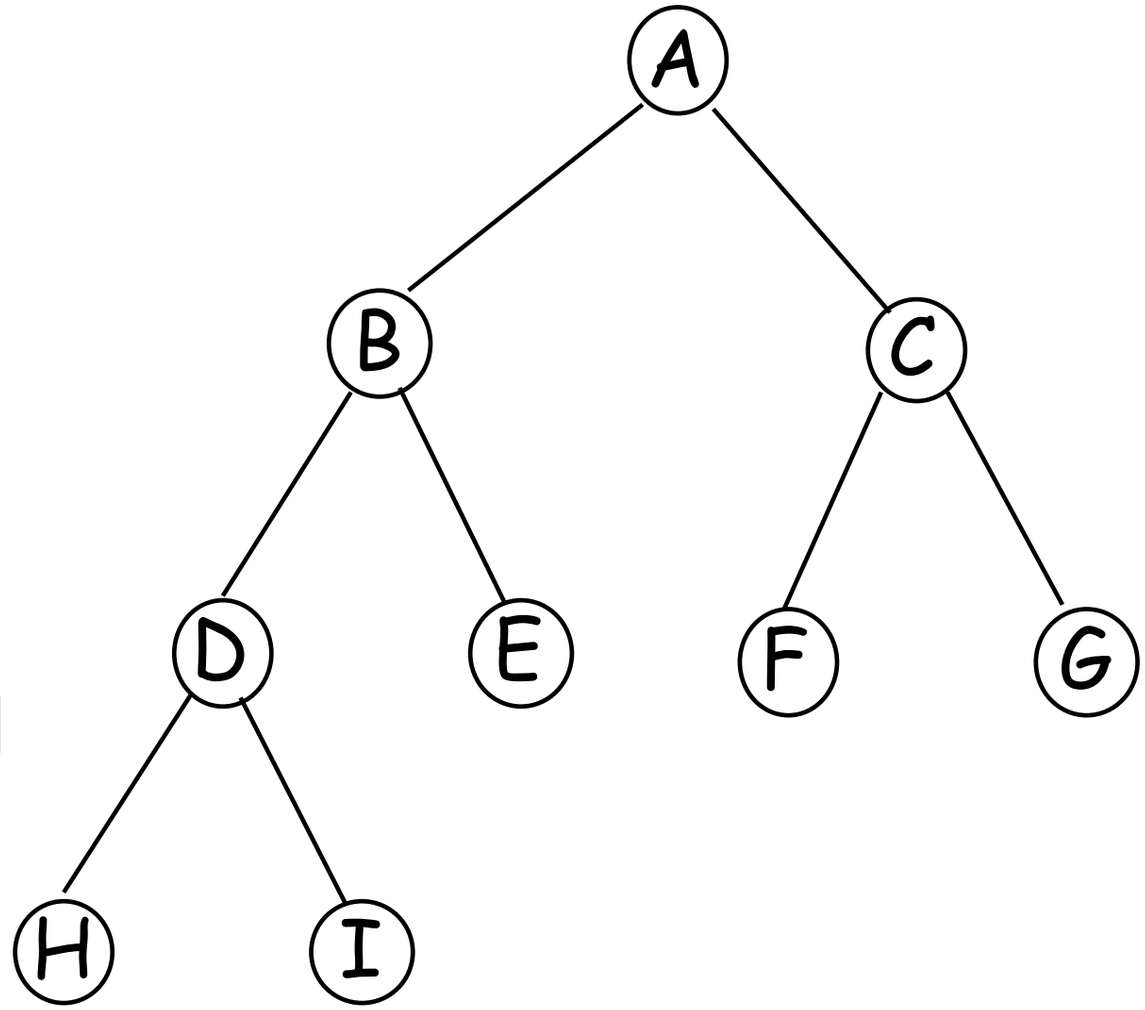

parent(*i* ) = i / 2

left_child(*i* ) = 2*i

right_child(*i* ) = 2*i+1

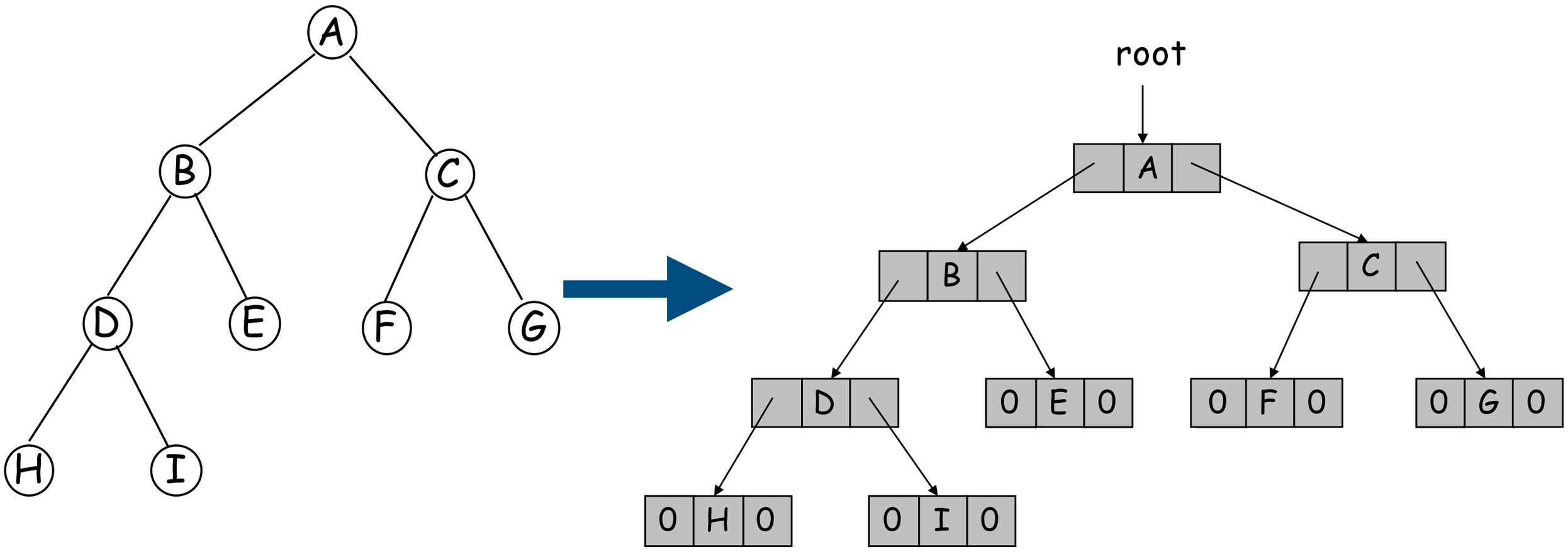
# Linked Representation of Binary Trees




| Left Child | **Data** | Right Child |
|---|---|---|

# Linked Representation of Binary Trees



# 使用鏈結串列實作二元樹

```
typedef struct node *treePointer;
typedef struct node {
    int value;
    treePointer *left;
    treePointer *right;
} tree;
```

# 新增tree

```
void insert(tree **t, int *num, int index, int n)
{
    if (index < n) {
        *t = malloc(sizeof(**t));

        (*t)->value = num[index];
        (*t)->left = NULL;
        (*t)->right = NULL;

        insert(&(*t)->left, num, 2 * index, n);
        insert(&(*t)->right, num, 2 * index + 1, n);
    }
}
```

# 練習一

- 輸入一串數字，利用Array存成Complete Binary Tree，並印出Parent、Level、Left Child、Right Child。

- 不可印出沒有Left Child、Right Child的subtree

```
Input number(Exit 0):
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 0
Parent: 1
Level: 1
Left Child: 2
Right Child: 3
-------------------------
Parent: 2
Level: 2
Left Child: 4
Right Child: 5
-------------------------
Parent: 3
Level: 2
Left Child: 6
Right Child: 7
-------------------------
Parent: 4
Level: 3
Left Child: 8
Right Child: 9
-------------------------
Parent: 5
Level: 3
Left Child: 10
Right Child: 11
-------------------------
Parent: 6
Level: 3
Left Child: 12
Right Child: 13
-------------------------
Parent: 7
Level: 3
Left Child: 14
Right Child: 15
-------------------------
Parent: 8
Level: 4
Left Child: 16
Right Child: 17
```

# 練習二

- 輸入一串數字，利用List存成Complete Binary Tree，並印出tree。

- 不可印出沒有Left Child、Right Child的subtree