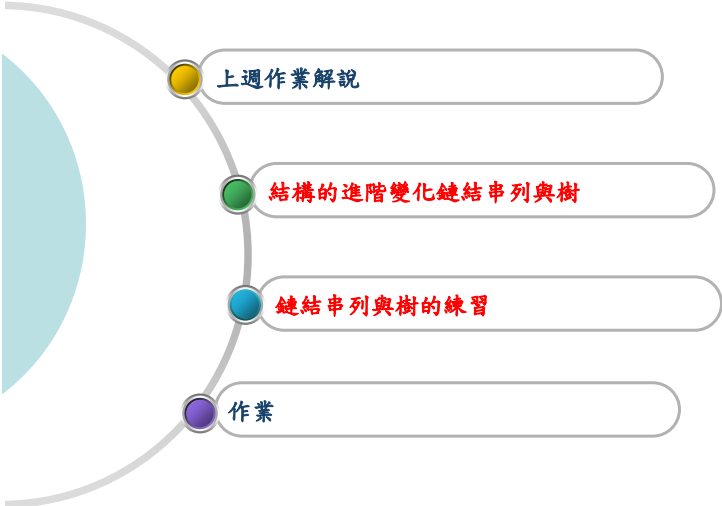


程式設計第十二章



真理大學資工系 洪麗玲
llhung@mail.au.edu.tw

摘要



題目：



- ❖ 請修改比較 N 個數字程式，除了將原數字與數字和當作一筆資料的兩個成員資料另外再加上一指標成員，程式中將輸入的資料存成一鏈結串列，使其中的串列資料是按照資料和由大到小的順序

3

WATSE

鏈結串列



❖ 鏈結

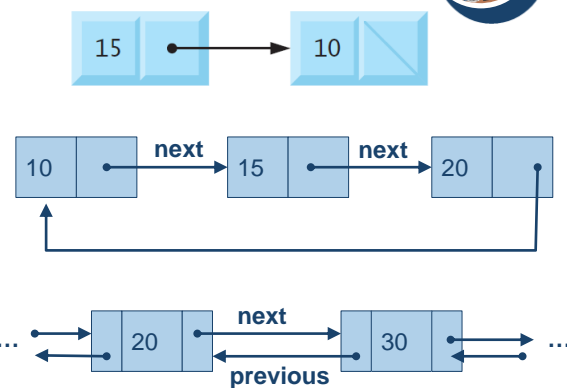
- 非環狀鏈結
- 環狀鏈結

❖ 鏈結

- 單向鏈結
- 雙向鏈結

```
struct Node{
    int data;
    struct Node *llink;
    struct Node *rlink;
};
```

10



WATSE

12.7 樹



❖ 鏈結串列是**線性資料結構 (linear data structure)**。
樹(tree)是非線性的 (nonlinear)、它是具有特殊屬性的二維資料結構。樹的節點包含兩個或多個鏈結。本節討論**二元樹 (binary trees)**，圖12.17)。

WATSE

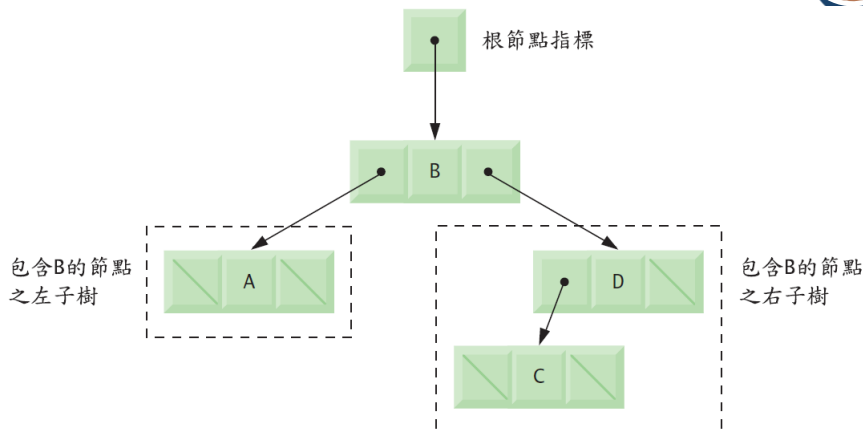
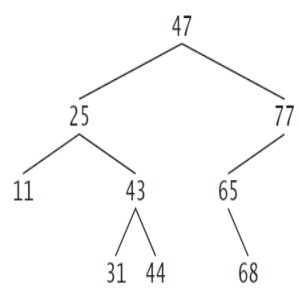


圖12.17 二元樹的圖形表示法

WATSE



❖ 圖12.18說明一個具有12個數值的二元搜尋樹。



如何建立樹結構



- ❖ 新增
- ❖ 查詢
- ❖ 修改
- ❖ 刪除



宣告



```
struct treeNode {
    struct treeNode *leftPtr; // pointer to left subtree
    int data; // node value
    struct treeNode *rightPtr; // pointer to right subtree
}; // end structure treeNode
```

```
typedef struct treeNode TreeNode;
// synonym for struct treeNode
typedef TreeNode *TreeNodePtr;
// synonym for TreeNode*
```



```
void insertNode( TreeNodePtr *treePtr, int value )
```

```
{
    if ( *treePtr == NULL ) { // if tree is empty
        *treePtr = malloc( sizeof( TreeNode ) );
        // if memory was allocated, then assign data
        if ( *treePtr != NULL ) {
            ( *treePtr )->data = value;
            ( *treePtr )->leftPtr = NULL;
            ( *treePtr )->rightPtr = NULL;
        } // end if
        else {
            printf( "%d not inserted. No memory available.\n",
value );
        } // end else
    } // end if
```



```
else { // tree is not empty
```

```
    // data to insert is less than data in current node
```

```
    if ( value < ( *treePtr )->data ) {
```

```
        insertNode( &( ( *treePtr )->leftPtr ), value );
```

```
    } // end if
```

```
    // data to insert is greater than data in current node
```

```
    else if ( value > ( *treePtr )->data ) {
```

```
        insertNode( &( ( *treePtr )->rightPtr ), value );
```

```
    } // end else if
```

```
    else { // duplicate data value ignored
```

```
        printf( "%s", "duplicated value \n" );
```

```
    } // end else
```

```
    } // end else
```

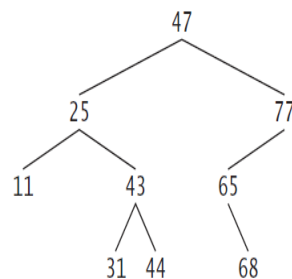
```
} // end function insertNode
```



17

WATSE

❖ 圖12.18說明一個具有12個數值的二元搜尋樹。



WATSE



❖ 12.7.3 消除資料中的重複數值

- 建立二元搜尋樹可以**消除重複數值(duplicate elimination)**。因為每次比較都有相同的決定「往左」還是「往右」。因此，這個重複的數值最終會與含有相同數值的節點進行比較。此時便可以很容易將重複的數值捨棄。

WATSE



❖ 12.7.4 二元樹搜尋

- 在二元樹內搜尋某個鍵值也很快。如果樹的結構十分緊密的話，則每一層所包含的元素個數最多大約是上一層元素個數的兩倍。所以一棵具有 n 個元素的二元搜尋樹最多只有 $\log_2 n$ 層，因此程式最多只需要 $\log_2 n$ 次的比較，便能夠找到 (或找不到) 我們所要的數值。舉例來說，當我們搜尋一棵具有1000個元素的二元搜尋樹 (緊密排列的) 時，最多不會超過10次的比較，因為 $2^{10} > 1000$ 。

WATSE



❖ 12.7.5 其他二元樹操作

- 從二元樹刪除一個節點
- 以二維樹狀格式印出一棵二元樹
- 依據階層順序來走訪二元樹。依階層順序走訪二元樹會從樹根節點的階層開始，逐層地走訪每一層的節點。
-

WATSE

本週練習（建立二元搜尋樹）



- ❖ 請修改比較 N 個數字程式，除了將原數字與數字和當作一筆資料的兩個成員資料另外再加上二指標成員，程式中將輸入的資料存成一棵樹，使其中的樹資料是按照資料和較小的放左子樹、較大的放右子樹

- ❖ 新增
- ❖ 搜尋
- ❖ 刪除

WATSE