



Visual Basic

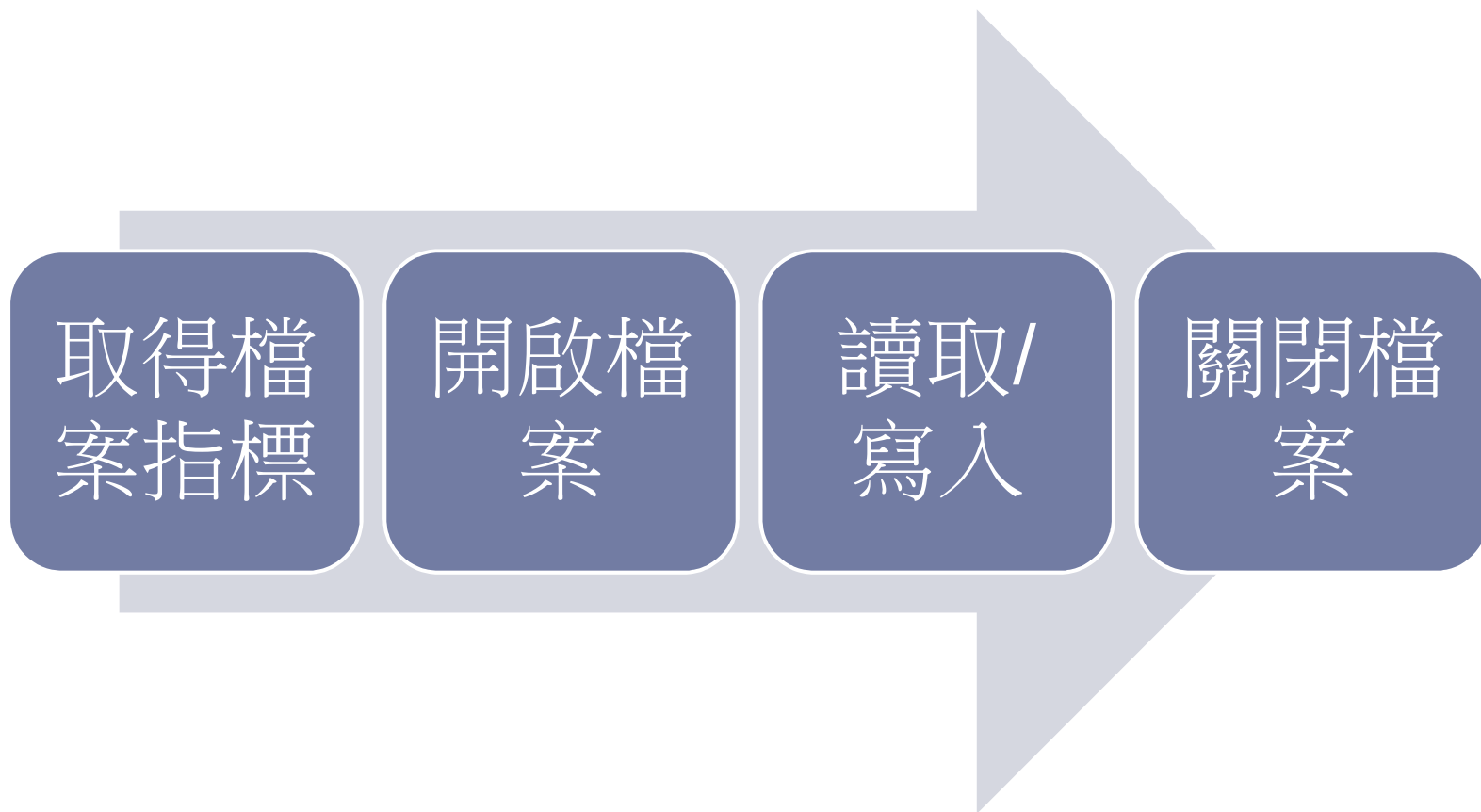
Visual Basic程式語言 part2

劉和師

2019/03/18

Ch7.讀取檔案

- ▶ 文字檔(.txt)讀取與寫入：



Ch7.讀取檔案

▶ 取得檔案指標

```
Dim fp As Integer  
:  
fp = FreeFile()
```

- ▶ 在程式中要開啟一個檔案必須先取得一個指標代碼，然後用此代碼在程式中代表這個檔案
- ▶ `FreeFile()` 函數會自動取得可用的代碼
- ▶ `fp` 是個整數變數，名稱依變數命名規則即可

Ch7.讀取檔案

▶ 開啟檔案

```
Dim fname As String  
fname = "d:\aaa.txt"      ‘要開啟的檔案  
:  
:  
FileOpen(fp, fname, OpenMode.Output)
```

檔案指標

檔案名稱

檔案開啟模式

▶ 注意必須有磁碟機及檔案存取的權限

Ch7.讀取檔案

▶ 檔案開啟模式

模 式	說 明
Output	開啟檔案進行寫入存取權。
Input	開啟檔案進行讀取存取權。
Append	若要將附加至開啟的檔案。(預設值)
Binary	二進位存取所開啟的檔案。
Random	開啟檔案進行隨機存取。

- ▶ Input、Output和 Append 時，會使用循序存取檔案，例如文字檔，而 Binary 用於二進位檔案存取，Random 用於隨機檔案存取
- ▶ 當循序存取檔案時，無法變更其資料。只可以讀取資料、附加到檔案，或以新的資料覆寫它。

Ch7.讀取檔案

▶ 寫入檔案：

寫入檔案，並加入換行符號

```
WriteLine(fp, "要寫入的文字")  
Write(fp, "要寫入的文字")
```

寫入檔案，沒有換行符號

▶ 讀取檔案：

```
Input(fp, 變數)
```

讀取的文字會放入變數中，每次讀取一行或逗點分隔之文字段落

若某一行之內容為：
"AAA", "BBB", "CCC"
則一次Input只讀入"AAA"

Ch7.讀取檔案

- ▶ 關閉檔案：`FileClose(fp)`
- ▶ 檔案存取完畢應確實關閉檔案，釋放相關資源。
- ▶ 如果讀取檔案時不知道有多少資料，就不能使用FOR迴圈了
- ▶ 可以使用`EOF()`函數(`End Of File`)，判斷是否已到達檔案的最後

`EOF(檔案指標)`

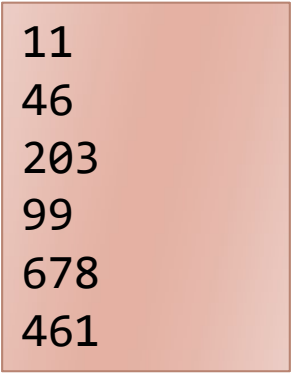
- ▶ 當到達檔案結尾時傳回True，否則傳回False

Ch7.讀取檔案

- ▶ 讀取一個檔案內的數字，求其平均值。

```
Sub Main()  
    Dim fname As String  
    Dim fp, sum, Alen, x As Integer  
    fname = "Adata.txt"  
    '檔案與程式需在同一個資料夾  
    fp = FreeFile() : sum = 0 : Alen = 0  
    FileOpen(fp, fname, OpenMode.Input)  
    While Not EOF(fp)  
        Input(fp, x)    '讀入一行資料到x  
        sum += Val(x)  
        Alen += 1  
    End While  
    FileClose(fp)  
    Console.WriteLine("Average: " & sum / Alen)  
    Console.Read()  
End Sub
```

Adata.txt內容



11
46
203
99
678
461

Ch7.讀取檔案

- ▶ 將兩個檔案中的數字加起來，再將結果存入第三個檔案。(為方便測試，Adata.txt與Bdata.txt內容相同)

Adata.txt內容

```
11
46
203
99
678
461
```

+

Bdata.txt內容

```
11
46
203
99
678
461
```

=

Cdata.txt內容

```
22
92
406
198
1356
922
```



Ch7.讀取檔案

▶ 參考程式：

```
Sub open_and_print_file(filename)
    Dim fp As Integer : Dim x As String
    fp = FreeFile() : FileOpen(fp, filename, OpenMode.Input)
    Console.WriteLine("File: " & filename)
    While Not EOF(fp)
        Input(fp, x) '讀入資料並顯示在螢幕上
        Console.WriteLine(x)
    End While
    Console.WriteLine()
    FileClose(fp)
End Sub

Sub addfile(x, y, z)
    Dim fp1, fp2, fp3 As Integer
    Dim u, v As String
    '取得檔案配置編號，然後開啟檔案
    fp1 = FreeFile() : FileOpen(fp1, x, OpenMode.Input)
    fp2 = FreeFile() : FileOpen(fp2, y, OpenMode.Input)
    fp3 = FreeFile() : FileOpen(fp3, z, OpenMode.Output)
    While Not (EOF(fp1) Or EOF(fp2))
        Input(fp1, u) '讀入資料
        Input(fp2, v)
        WriteLine(fp3, Val(u) + Val(v)) '寫入資料加換行符號
    End While
    FileClose(fp1) : FileClose(fp2) : FileClose(fp3)
End Sub

Sub Main()
    open_and_print_file("d:\Adata.txt")
    open_and_print_file("d:\Bdata.txt")
    addfile("d:\Adata.txt", "d:\Bdata.txt", "d:\Cdata.txt")
    open_and_print_file("d:\Cdata.txt")
    Console.Read()
End Sub
```

Ch7. 習題

- 7-1. 寫一程式，從檔案中讀入 n 及 a_1, a_2, \dots, a_n 。 n 和所有的 a_i 都要是正整數，計算出 $a_1^2, a_2^2, \dots, a_n^2$ 。再將 $a_i, a_i^2, i=1$ 到 $i=n$ 寫到另一個檔案上。

ex7-1內容

10
1
2
3
4
5
6
7
8
9
10

ex7-1a內容

1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100

Ch7. 習題

► 7-1 參考程式：

```
Sub Main()  
    Dim f1, f2, n, u, v As Integer  
    f1 = FreeFile() : FileOpen(f1, "d:\ex7-1.txt", OpenMode.Input)  
    f2 = FreeFile() : FileOpen(f2, "d:\ex7-1a.txt", OpenMode.Output)  
    Input(f1, n)  
    For i = 1 To n  
        Input(f1, u)  
        v = u ^ 2  
        WriteLine(f2, u & vbTab & v)  
        Console.WriteLine(u & vbTab & v)  
    Next  
    FileClose(f1)  
    FileClose(f2)  
    Console.Read()  
End Sub
```



Ch7. 習題

- 7-2. 寫一程式，從檔案中讀入 n 及 a_1, a_2, \dots, a_n 。 n 和所有的 a_i 都要是正整數，計算出 $b_i = \sqrt{a_i} \times 10$ ，將 $a_i, b_i, i=1$ 到 $i=n$ 寫到另一個檔案上去。

ex7-2內容

```
10
1
2
3
4
5
6
7
8
9
10
```

ex7-2a內容

```
1      10.0
2      14.142135623730951
3      17.32050807568877
4      20.0
5      22.360679774997898
6      24.49489742783178
7      26.457513110645905
8      28.284271247461902
9      30.0
10     31.622776601683796
```

Ch7. 習題

► 7-2 參考程式：

```
Sub Main()  
    Dim f1, f2, n, u As Integer  
    Dim v As Single  
    f1 = FreeFile() : FileOpen(f1, "d:\ex7-2.txt", OpenMode.Input)  
    f2 = FreeFile() : FileOpen(f2, "d:\ex7-2a.txt", OpenMode.Output)  
    Input(f1, n)  
    For i = 1 To n  
        Input(f1, u)  
        v = u ^ 0.5 * 10  
        WriteLine(f2, u & vbTab & v)  
        Console.WriteLine(u & vbTab & v)  
    Next  
    FileClose(f1)  
    FileClose(f2)  
    Console.Read()  
End Sub
```



Ch7. 習題

- ▶ 7-3. 寫一程式，從檔案中讀入一個3x5的矩陣，求此矩陣的transpose(轉置)，然後將此結果寫到另一個檔案。
- ▶ 檔案內數字以跳格符號(Tab)分開。

ex7-3內容

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

ex7-3a內容

1	6	11
2	7	12
3	8	13
4	9	14
5	10	15

Ch7. 習題

► 7-3. 參考程式：

```
Sub print_matrix(x, a, b) '印出陣列
    For i = 1 To a
        For j = 1 To b
            Console.Write(x(i, j) & vbTab)
        Next
        Console.WriteLine()
    Next
    Console.WriteLine()
End Sub

Sub Main()
    Dim f1, f2, A(3, 5), B(5, 3) As Integer
    Dim s As String
    f1 = FreeFile() : FileOpen(f1, "d:\ex7-3.txt", OpenMode.Input)
    f2 = FreeFile() : FileOpen(f2, "d:\ex7-3a.txt", OpenMode.Output)
    For i = 1 To 3 '讀入檔案至陣列A
        For j = 1 To 5
            Input(f1, A(i, j))
        Next
    Next
    print_matrix(A, 3, 5)
    For j = 1 To 5 '轉置陣列
        s = ""
        For i = 1 To 3
            B(j, i) = A(i, j)
            s = s & B(j, i) & vbTab
        Next
        WriteLine(f2, s) '寫入一行至檔案
    Next
    print_matrix(B, 5, 3)
    FileClose(f1) : FileClose(f2)
    Console.Read()
End Sub
```


Ch7. 習題

- ▶ 7-4. 寫一程式，從一檔案中，讀入一組一元二次方程式 $ax^2+bx+c=0$ 的係數 a ， b ， c ，其中 $(b^2-4ac) \geq 0$ 。解此組方程式，並將中間過程及最後答案寫入另一檔案。
- ▶ (本題略過)



Ch7. 習題

- ▶ 7-5. 寫一程式，從檔案中讀入 n 及 a_1, a_2, \dots, a_n 。 n 和所有的 a_i 都要是正整數，將 a_n, a_{n-1}, \dots, a_1 寫到另一個檔案上去。

ex7-5內容

```
10  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

ex7-5a內容

```
10  
9  
8  
7  
6  
5  
4  
3  
2  
1
```

Ch7.讀取檔案

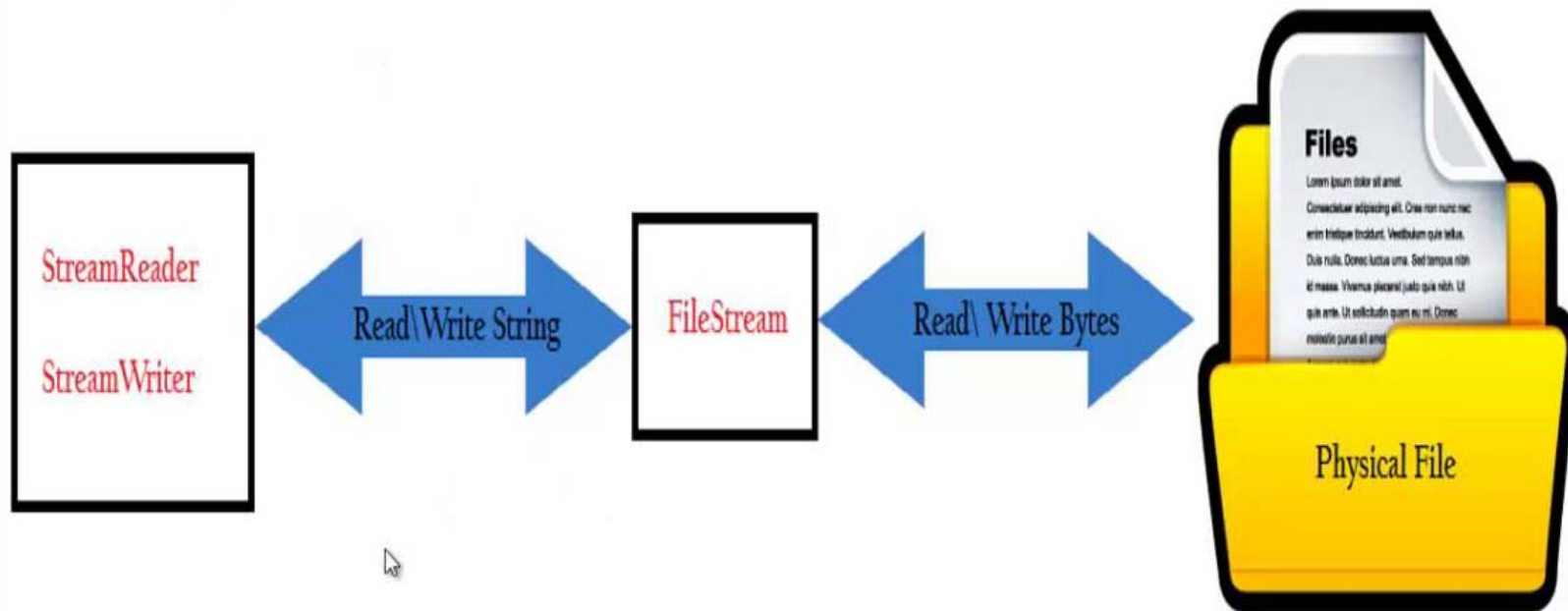
► 7-5.參考程式：

```
Sub Main()  
    Dim f1, f2, A(), n As Integer  
    f1 = FreeFile() : FileOpen(f1, "d:\ex7-5.txt", OpenMode.Input)  
    f2 = FreeFile() : FileOpen(f2, "d:\ex7-5a.txt", OpenMode.Output)  
    Input(f1, n)  
    ReDim A(n)  
    For i = 1 To n  
        Input(f1, A(i))  
        Console.WriteLine(A(i))  
    Next  
    Console.WriteLine()  
    For i = n To 1 Step -1  
        WriteLine(f2, A(i))  
        Console.WriteLine(A(i))  
    Next  
    FileClose(f1) : FileClose(f2)  
    Console.Read()  
End Sub
```



Ch7.讀取檔案

- ▶ 以上介紹及使用的方法是比較傳統的，由於VB已經高度物件化了，所以現在都已採用”資料流“的方式來存取檔案。



Ch7.讀取檔案

▶ 檔案讀取範例：

```
Imports System.IO '需引入參考
Module Module1
    Sub main()
        '檔案相關宣告
        Dim myFile As New FileInfo("d:\Ddata.txt")
        '建立資料流物件
        Dim fs As FileStream = myFile.Open(FileMode.Open)
        '建立讀取資料流
        Dim ds As New StreamReader(fs, System.Text.Encoding.Default)
        Dim str As String = ""
        If fs.Length > 0 Then
            Do
                str = str & ds.ReadLine & vbCrLf '讀取檔案一行
            Loop While ds.Peek <> -1 '偵測是否到達檔案尾端
            Console.WriteLine(str)
        End If
        '關閉資料流
        ds.Close()
        fs.Close()
        Console.Read()
    End Sub
End Module
```

Ch7.讀取檔案

▶ 檔案寫入範例：

```
Imports System.IO      '需引入參考
Module Module1
    Sub main()
        '檔案相關宣告
        Dim myFile As New FileInfo("d:\Ddata1.txt")
        '建立資料流物件
        Dim fs As FileStream = myFile.Create
        '建立寫入資料流
        Dim ds As New StreamWriter(fs, System.Text.Encoding.Default)
        '寫入檔案
        ds.Write("test")
        '關閉資料流
        ds.Close()
        fs.Close()
    End Sub
End Module
```

休息一下~



Ch8.結構

- ▶ 又稱自訂型態(Structure)
- ▶ 當資料是由不同的資料型別組成時，可以使用結構將一個或多個資料型態包裝在一起，定義成一個新的形態。
- ▶ 李伯伯的書裡本章是介紹配置(configuration)，VB沒有這東西，但有點類似C或VB語言的結構(structure)。所以我們用結構來實作本章範例。



Ch8.結構

- ▶ 自己定義新的複合資料當作一個新的形態

- ▶ 例如：

```
Structure student
    Dim name As String
    Dim age As Integer
    Dim BMI As Single
End Structure
```

- ▶ 建立一個新的資料型態叫 student

```
Dim stu1 As student
John.name = "John Carter"
John.age = 20
John.BMI = 22.5
```

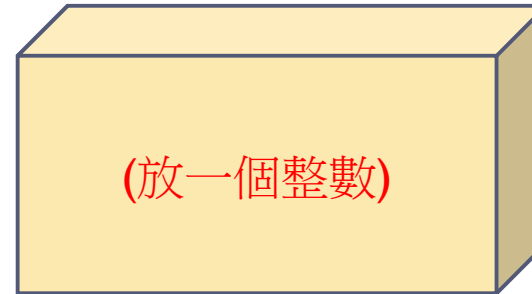
- ▶ 宣告stu1這個變數的型態是student，並用句點 . 來存取student內的成員

Ch8.結構

► Dim A As Integer

► (單一變數)

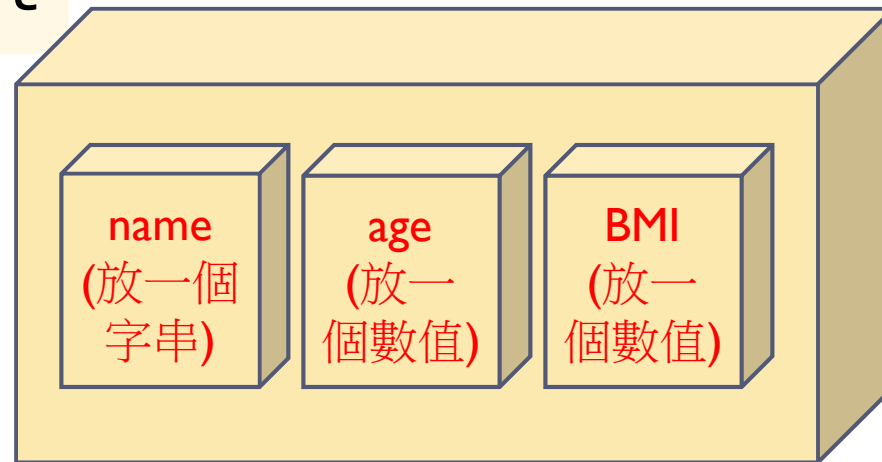
A



► Dim B As student

► (結構變數)

B



Ch8.結構

- ▶ 例如一個學生的課業資料如下：

```
name = 劉和師  
mathscore = 80  
chinesescore = 99  
engscore = 97
```

- ▶ 我們可以利用自行定義一個structure來結合這些資料，就叫studentdata好了，我們把它放在檔案內，檔名叫student_score.txt，我們可以很方便的來分析、搜尋資料。



Ch8.結構

- ▶ 在磁碟內建立一個檔案，並寫出輸入並儲存班級學生各科成績等功能，內容如下。

學生姓名, 數學成績, 國文成績, 英文成績

劉和師, 77, 90, 90
林志玲, 67, 76, 57
劉德華, 78, 90, 78

student_score.txt內容

Ch8.結構

- ▶ 我們要做出以下各項功能：

```
----- 功能表 -----  
1. 新增一個學生的資料  
2. 查詢一個學生的資料  
3. 修改一個學生的資料  
4. 刪除一個學生的資料  
5. 印出所有學生的資料  
6. 求班級各科成績平均  
7. 結束程式  
-----  
請選擇：
```

- ▶ 為簡化程式，我們將每一個功能寫成一個副程式。



Ch8.結構

- ▶ 先定義結構及全域變數：

```
'定義資料格式
Structure studentdata
    Dim name As String      '學生姓名
    Dim mathscore As Single '數學成績
    Dim chiscore As Single  '國文成績
    Dim engscore As Single  '英文成績
End Structure

Dim stu() As studentdata
```

- ▶ 要寫在所有的程式段之外(sub、Function、main等)，通常在程式一開始的地方。



Ch8.結構

'讀入學生資料檔至結構陣列副程式

```
Sub read_student()  
    Dim f1, n As Integer  
    f1 = FreeFile()  
    FileOpen(f1, "d:\student_score.txt", OpenMode.Input)  
    n = 0  
    While Not EOF(f1)  
        n += 1  
        ReDim Preserve stu(n) '隨著讀入的資料筆數增加陣列  
        Input(f1, stu(n).name) '用 . 來存取結構內的成員  
        Input(f1, stu(n).mathscore)  
        Input(f1, stu(n).chiscore)  
        Input(f1, stu(n).engscore)  
    End While  
    FileClose(f1)  
End Sub
```



Ch8.結構

'將陣列資料寫入檔案

```
Sub write_to_file()  
    Dim f1 As Integer  
    f1 = FreeFile() 'Output模式會清除檔案裡原資料  
    FileOpen(f1, "d:\student_score.txt", OpenMode.Output)  
    For i = 1 To UBound(stu)  
        Write(f1, stu(i).name)  
        Write(f1, stu(i).mathscore)  
        Write(f1, stu(i).chiscore)  
        WriteLine(f1, stu(i).engscore) '最後一項資料加上換行  
    Next  
    FileClose(f1)  
End Sub
```



Ch8.結構

► 主程式：

```
Sub Main()  
    Dim sel As Integer  
    read_student() '讀入檔案資料至結構陣列  
    While True  
        Console.WriteLine("----- 功能表 -----")  
        Console.WriteLine("1. 新增一個學生的資料")  
        Console.WriteLine("2. 查詢一個學生的資料")  
        Console.WriteLine("3. 修改一個學生的資料")  
        Console.WriteLine("4. 刪除一個學生的資料")  
        Console.WriteLine("5. 印出所有學生的資料")  
        Console.WriteLine("6. 求班級各科成績平均")  
        Console.WriteLine("7. 結束程式")  
        Console.WriteLine("-----")  
        Console.Write("請選擇:")  
        sel = Val(Console.ReadLine())  
        Select Case sel '每一項功能都去呼叫副程式  
            Case 1  
                input_studentdata()  
            Case 2  
                search_student()  
            Case 3  
                modify_student()  
            Case 4  
                remove_student()  
            Case 5  
                print_student()  
            Case 6  
                score_average()  
            Case 7  
                Exit While  
            Case Else  
                Console.WriteLine("錯誤，請輸入 1 ~ 7")  
        End Select  
    End While  
End Sub
```

Ch8.結構

```
'新增一個學生的資料
Sub input_studentdata()
    Dim name As String
    Dim flag, i As Integer
    Console.Write("請輸入學生姓名: ")
    name = Console.ReadLine()
    flag = 0
    For i = 1 To UBound(stu)
        If stu(i).name = name Then
            Console.WriteLine("該名學生已存在，無法新增")
            flag = 1
            Exit For
        End If
    Next
    If flag = 0 Then
        i = UBound(stu) + 1
        ReDim Preserve stu(i)
        stu(i).name = name
        Console.Write("請輸入數學成績: ")
        stu(i).mathscore = Console.ReadLine()
        Console.Write("請輸入國文成績: ")
        stu(i).chiscore = Console.ReadLine()
        Console.Write("請輸入英文成績: ")
        stu(i).engscore = Console.ReadLine()
        write_to_file()
        Console.WriteLine("新增學生" & name & "資料成功!")
    End If
End Sub
```

Ch8.結構

'查詢一個學生的資料

```
Sub search_student()  
    Dim name As String  
    Dim flag As Integer  
    Console.Write("請輸入學生姓名: ")  
    name = Console.ReadLine()  
    flag = 0  
    For i = 1 To UBound(stu)  
        If stu(i).name = name Then  
            Console.WriteLine("數學成績: " & stu(i).mathscore)  
            Console.WriteLine("國文成績: " & stu(i).chiscore)  
            Console.WriteLine("英文成績: " & stu(i).engscore)  
            flag = 1  
            Exit For  
        End If  
    Next  
    If flag = 0 Then Console.WriteLine("無此學生")  
End Sub
```



Ch8.結構

```
Sub modify_student() '修改一個學生的資料
    Dim name, temp As String
    Dim flag As Integer = 0
    Console.Write("請輸入學生姓名: ")
    name = Console.ReadLine()
    For i = 1 To UBound(stu)
        If stu(i).name = name Then
            Console.Write("原學生姓名: " & stu(i).name & ",請輸入修改後姓名: ")
            temp = Console.ReadLine() '要考慮若不修改，直接按Enter的狀況
            If Len(temp) > 0 Then stu(i).name = temp
            Console.Write("原數學成績: " & stu(i).mathscore & ",請輸入修改後成績: ")
            temp = Console.ReadLine()
            If Len(temp) > 0 Then stu(i).mathscore = Val(temp)
            Console.Write("原國文成績: " & stu(i).chiscore & ",請輸入修改後成績: ")
            temp = Console.ReadLine()
            If Len(temp) > 0 Then stu(i).chiscore = Val(temp)
            Console.Write("原英文成績: " & stu(i).engscore & ",請輸入修改後成績: ")
            temp = Console.ReadLine()
            If Len(temp) > 0 Then stu(i).engscore = Val(temp)
            write_to_file() '寫入檔案
            Console.WriteLine("修改學生" & name & "資料成功!")
            flag = 1 : Exit For
        End If
    Next
    If flag = 0 Then Console.WriteLine("無此學生")
End Sub
```

Ch8.結構

```
Sub remove_student() '刪除一個學生的資料
    Dim name As String
    Dim flag As Integer = 0
    Console.Write("請輸入學生姓名: ")
    name = Console.ReadLine()
    For i = 1 To UBound(stu)
        If stu(i).name = name Then '找到資料後將後面的資料依序向前移動一個位置
            For j = i To UBound(stu) - 1
                stu(j).name = stu(j + 1).name '覆蓋掉要刪除的資料
                stu(j).mathscore = stu(j + 1).mathscore
                stu(j).chiscore = stu(j + 1).chiscore
                stu(j).engscore = stu(j + 1).engscore
            Next
            ReDim Preserve stu(UBound(stu) - 1)
            write_to_file() '寫入檔案
            Console.WriteLine("刪除學生" & name & "成功!")
            flag = 1
            Exit For
        End If
    Next
    If flag = 0 Then Console.WriteLine("無此學生")
End Sub
```

Ch8.結構

'印出所有學生的資料

```
Sub print_student()  
    Console.WriteLine("-----學生成績資料-----")  
    For i = 1 To UBound(stu)  
        Console.WriteLine("學生姓名: " & stu(i).name)  
        Console.WriteLine("數學成績: " & stu(i).mathscore)  
        Console.WriteLine("國文成績: " & stu(i).chiscore)  
        Console.WriteLine("英文成績: " & stu(i).engscore)  
        Console.WriteLine()  
    Next  
End Sub
```



Ch8.結構

'求班級各科成績平均

```
Sub score_average()  
    Dim math_sum, chi_sum, eng_sum As Single  
    Dim n As Integer  
    n = UBound(stu)  
    For i = 1 To n  
        math_sum += stu(i).mathscore  
        chi_sum += stu(i).chiscore  
        eng_sum += stu(i).engscore  
    Next  
    Console.WriteLine("班級數學平均: " & math_sum / n)  
    Console.WriteLine("班級國文平均: " & chi_sum / n)  
    Console.WriteLine("班級英文平均: " & eng_sum / n)  
End Sub
```



Ch8.結構

- ▶ 設計一個簡易借還書程式。
- ▶ 這個程式有以下功能：
 - ▶ 1.新增書籍
 - ▶ 2.借書
 - ▶ 3.還書
 - ▶ 4.結束程式
- ▶ 結構的格式如下：
 - ▶ 1.書名
 - ▶ 2.出版社
 - ▶ 3.作者
 - ▶ 4.頁數
 - ▶ 5.借閱人(為none時表示尚無人借閱)



Ch8.結構

- ▶ 程式執行後，檔案booklist.txt的內容參考。

資料依序為：書名,出版社,作者,頁數,借閱人

迷霧之子I,皇冠出版社,布蘭登山德勒,684,劉和師
哈利波特I,CROWN,JK羅琳,1532,none
西遊記,三民書局,吳承恩,1515,none

Ch8.結構

- ▶ 先定義結構及全域變數：

```
'宣告全域變數
Dim books() As booklist

'定義資料格式
Structure booklist
    Dim title As String    '書名
    Dim com As String      '出版社
    Dim writer As String   '作者
    Dim page As Integer    '頁數
    Dim name As String     '借閱人
End Structure
```

Ch8.結構

'讀入檔案資料

```
Sub readBook()  
    Dim f1, n As Integer  
    f1 = FreeFile()  
    FileOpen(f1, "d:\booklist.txt", OpenMode.Input)  
    n = 0  
    While Not EOF(f1)  
        n += 1  
        ReDim Preserve books(n)      '隨著讀入的資料筆數增加陣列  
        Input(f1, books(n).title)    '用 . 來存取結構內的成員  
        Input(f1, books(n).com)  
        Input(f1, books(n).writer)  
        Input(f1, books(n).page)  
        Input(f1, books(n).name)  
    End While  
    FileClose(f1)  
End Sub
```



Ch8.結構

'寫入檔案資料

```
Sub writeBook()
```

```
    Dim f1 As Integer
```

```
    f1 = FreeFile() 'Output模式會清除檔案裡原資料
```

```
    FileOpen(f1, "d:\booklist.txt", OpenMode.Output)
```

```
    For i = 1 To UBound(books)
```

```
        Write(f1, books(i).title)
```

```
        Write(f1, books(i).com)
```

```
        Write(f1, books(i).writer)
```

```
        Write(f1, books(i).page)
```

```
        WriteLine(f1, books(i).name) '最後一項資料加上換行
```

```
    Next
```

```
    FileClose(f1)
```

```
End Sub
```



Ch8.結構

```
Sub addBook() '新增書籍
    Dim title As String
    Dim flag, i As Integer
    Console.Write("請輸入欲新增書籍的書名: ")
    title = Console.ReadLine()
    flag = 0
    For i = 1 To UBound(books)
        If books(i).title = title Then
            Console.WriteLine("這本書已經存在了，無法新增")
            flag = 1 : Exit For
        End If
    Next
    If flag = 0 Then
        i = UBound(books) + 1
        ReDim Preserve books(i)
        books(i).title = title
        Console.Write("請輸入" & title & "的出版社: ") : books(i).com = Console.ReadLine()
        Console.Write("請輸入" & title & "的作者: ") : books(i).writer = Console.ReadLine()
        Console.Write("請輸入" & title & "的頁數: ") : books(i).page = Console.ReadLine()
        books(i).name = "none" '新增書本，尚無人借閱，故填入none
        writeBook()
        Console.WriteLine("新增書籍「" & title & "」資料成功!")
    End If
End Sub
```

Ch8.結構

```
Sub borrowBook() '借書
    Dim title As String
    Dim flag As Integer
    Console.Write("請輸入欲借閱的書名: ")
    title = Console.ReadLine()
    flag = 0
    For i = 1 To UBound(books)
        If books(i).title = title Then
            flag = 1
            If books(i).name = "none" Then
                Console.Write("請輸入借閱人姓名:")
                books(i).name = Console.ReadLine()
                Console.WriteLine("借書成功")
                writeBook()
            Else
                Console.WriteLine("這本書已經被借走了")
            End If
        End If
    Exit For
    End If
Next
If flag = 0 Then Console.WriteLine("沒有這本書!")
End Sub
```

Ch8.結構

```
Sub returnBook() '還書
    Dim title, name As String
    Dim flag As Integer
    Console.Write("請輸入歸還的書名: ")
    title = Console.ReadLine()
    flag = 0
    For i = 1 To UBound(books)
        If books(i).title = title Then
            flag = 1
            Console.Write("請輸入借閱人姓名:")
            name = Console.ReadLine()
            If books(i).name = name Then
                books(i).name = "none"
                Console.WriteLine("還書成功")
                writeBook()
            Else
                Console.WriteLine(title & "不是被" & name & "所借走的!")
            End If
        End If
    Exit For
    End If
Next
If flag = 0 Then Console.WriteLine("沒有這本書!")
End Sub
```

Ch8.結構

'印出所有書籍資料

```
Sub listBook()  
    Console.WriteLine("-----所有書籍資料-----")  
    For i = 1 To UBound(books)  
        Console.WriteLine("書名: " & books(i).title)  
        Console.WriteLine("出版社: " & books(i).com)  
        Console.WriteLine("作者: " & books(i).writer)  
        Console.WriteLine("頁數: " & books(i).page)  
        Console.WriteLine("借閱人: " & books(i).name)  
        Console.WriteLine()  
    Next  
End Sub
```



Ch8.結構

```
Sub Main() '主程式
    Dim sel As Integer
    readBook() '讀入檔案資料至結構陣列
    While True
        Console.WriteLine("----- 功能表 -----")
        Console.WriteLine("1. 新增書籍")
        Console.WriteLine("2. 借書")
        Console.WriteLine("3. 還書")
        Console.WriteLine("4. 印出所有書籍資料")
        Console.WriteLine("5. 結束程式")
        Console.WriteLine("-----")
        Console.Write("請選擇:")
        sel = Val(Console.ReadLine())
        Select Case sel '每一項功能都去呼叫副程式
            Case 1
                addBook()
            Case 2
                borrowBook()
            Case 3
                returnBook()
            Case 4
                listBook()
            Case 5
                Exit While
            Case Else
                Console.WriteLine("錯誤，請輸入 1 ~ 5")
        End Select
    End While
End Sub
```

Ch8. 習題

- ▶ 我們要替一家醫院處理一些病患資料如下：
 - ▶ 1. 健保號碼
 - ▶ 2. 姓名
 - ▶ 3. 性別
 - ▶ 4. 電話
 - ▶ 5. 醫療科別
- ▶ 寫一程式，以結構的方式，輸入並儲存病人資料，要有以下功能：
 - ▶ 1. 新增一位病人資料
 - ▶ 2. 修改一位病人資料
 - ▶ 3. 查詢一位病人資料
 - ▶ 4. 刪除一位病人資料
 - ▶ 5. 列印所有女性病人姓名
 - ▶ 6. 結束程式



Ch8. 習題

- ▶ 程式執行後，檔案patientlist.txt的內容參考。

資料依序為：健保號碼,姓名,性別,電話,醫療科別

1001,張三,男,03-8312225,內科
1002,李四,男,03-8312227,外科
1003,隋棠,女,02-32569874,婦科
1004,林志玲,女,02-98745632,家醫科

Ch8. 習題

```
'宣告全域變數
Dim patients() As patientlist

'定義資料格式
Structure patientlist
    Dim PID As String           '健保號碼
    Dim name As String          '姓名
    Dim sex As String           '性別
    Dim tel As String           '電話
    Dim division As String      '醫療科別
End Structure
```



Ch8. 習題

'讀入檔案資料

```
Sub readPatient()  
    Dim f1, n As Integer  
    f1 = FreeFile()  
    FileOpen(f1, "d:\patientlist.txt", OpenMode.Input)  
    n = 0  
    While Not EOF(f1)  
        n += 1  
        ReDim Preserve patients(n) '隨著讀入的資料筆數增加陣列  
        Input(f1, patients(n).PID) '用 . 來存取結構內的成員  
        Input(f1, patients(n).name)  
        Input(f1, patients(n).sex)  
        Input(f1, patients(n).tel)  
        Input(f1, patients(n).division)  
    End While  
    FileClose(f1)  
End Sub
```



Ch8. 習題

'寫入檔案資料

```
Sub writePatient()  
    Dim f1 As Integer  
    f1 = FreeFile()  
    FileOpen(f1, "d:\patientlist.txt", OpenMode.Output)  
    For i = 1 To UBound(patients)  
        Write(f1, patients(i).PID)  
        Write(f1, patients(i).name)  
        Write(f1, patients(i).sex)  
        Write(f1, patients(i).tel)  
        WriteLine(f1, patients(i).division)  
    Next  
    FileClose(f1)  
End Sub
```



Ch8. 習題

```
'新增一位病人資料
Sub addPatient()
    Dim PID As String
    Dim flag, i As Integer
    Console.Write("請輸入新病患的健保號碼： ")
    PID = Console.ReadLine()
    flag = 0
    For i = 1 To UBound(patients)
        If patients(i).PID = PID Then
            Console.WriteLine("本病患資料已存在了")
            flag = 1
            Exit For
        End If
    Next
    If flag = 0 Then
        i = UBound(patients) + 1
        ReDim Preserve patients(i)
        patients(i).PID = PID
        Console.Write("請輸入" & PID & "的姓名： ")
        patients(i).name = Console.ReadLine()
        Console.Write("請輸入" & PID & "的性別： ")
        patients(i).sex = Console.ReadLine()
        Console.Write("請輸入" & PID & "的電話： ")
        patients(i).tel = Console.ReadLine()
        Console.Write("請輸入" & PID & "的醫療科別： ")
        patients(i).division = Console.ReadLine()
        writePatient()
        Console.WriteLine("新增成功!")
    End If
End Sub
```

Ch8. 習題

```
'修改一位病人資料
Sub editPatient()
    Dim PID, name, sex, tel, division As String
    Dim flag As Integer
    Console.WriteLine("請輸入欲修改的病患的健保號碼： ")
    PID = Console.ReadLine()
    flag = 0
    For i = 1 To UBound(patients)
        If patients(i).PID = PID Then
            flag = 1
            Console.WriteLine("請輸入新資料，空白則不修改")
            Console.WriteLine("原姓名：" & patients(i).name & ",新姓名:")
            name = Console.ReadLine()
            If name <> "" Then patients(i).name = name
            Console.WriteLine("原性別：" & patients(i).sex & ",新性別:")
            sex = Console.ReadLine()
            If sex <> "" Then patients(i).sex = sex
            Console.WriteLine("原電話：" & patients(i).tel & ",新電話:")
            tel = Console.ReadLine()
            If tel <> "" Then patients(i).tel = tel
            Console.WriteLine("原醫療科別：" & patients(i).division & ",新醫療科別:")
            division = Console.ReadLine()
            If division <> "" Then patients(i).division = division
            writePatient()
            Console.WriteLine("資料修改完畢!")
            Exit For
        End If
    Next
    If flag = 0 Then Console.WriteLine("查無本病患資料")
End Sub
```


Ch8. 習題

'查詢一位病人資料

```
Sub searchPatient()  
    Dim PID As String  
    Dim flag As Integer  
    Console.Write("請輸入欲查詢病患的健保號碼: ")  
    PID = Console.ReadLine()  
    flag = 0  
    For i = 1 To UBound(patients)  
        If patients(i).PID = PID Then  
            flag = 1  
            Console.WriteLine("-----病歷號碼" & PID & "的資料-----")  
            Console.WriteLine("姓名:" & patients(i).name)  
            Console.WriteLine("性別:" & patients(i).sex)  
            Console.WriteLine("電話:" & patients(i).tel)  
            Console.WriteLine("醫療科別:" & patients(i).division)  
            Exit For  
        End If  
    Next  
    If flag = 0 Then Console.WriteLine("查無本病患資料!")  
End Sub
```



Ch8. 習題

'刪除一位病人資料

```
Sub deletePatient()  
    Dim PID As String  
    Dim flag As Integer  
    Console.Write("請輸入欲刪除病患的健保號碼: ")  
    PID = Console.ReadLine()  
    flag = 0  
    For i = 1 To UBound(patients)  
        If patients(i).PID = PID Then  
            For j = i To UBound(patients) - 1  
                patients(j).PID = patients(j + 1).PID '覆蓋掉要刪除的資料  
                patients(j).name = patients(j + 1).name  
                patients(j).sex = patients(j + 1).sex  
                patients(j).tel = patients(j + 1).tel  
                patients(j).division = patients(j + 1).division  
            Next  
            ReDim Preserve patients(UBound(patients) - 1)  
            writePatient()  
            Console.WriteLine("本筆資料已刪除!")  
            Exit For  
        End If  
    Next  
    If flag = 0 Then Console.WriteLine("查無本病患資料!")  
End Sub
```

Ch8. 習題

'列印所有女性病人姓名

```
Sub printFemalePatient()  
    Console.WriteLine("-----所有女性病人姓名-----")  
    For i = 1 To UBound(patients)  
        If patients(i).sex = "女" Then  
            Console.WriteLine("健保號碼: " & patients(i).PID &  
                               ", 姓名: " & patients(i).name)  
        End If  
    Next  
End Sub
```



Ch8. 習題

```
Sub Main() '主程式
    Dim sel As Integer
    readPatient() '讀入檔案資料至結構陣列
    While True
        Console.WriteLine("----- 功能表 -----")
        Console.WriteLine("1. 新增一位病人資料")
        Console.WriteLine("2. 修改一位病人資料")
        Console.WriteLine("3. 查詢一位病人資料")
        Console.WriteLine("4. 刪除一位病人資料")
        Console.WriteLine("5. 列印所有女性病人姓名")
        Console.WriteLine("6. 結束程式")
        Console.WriteLine("-----")
        Console.Write("請選擇:")
        sel = Val(Console.ReadLine())
        Select Case sel '每一項功能都去呼叫副程式
            Case 1
                addPatient()
            Case 2
                editPatient()
            Case 3
                searchPatient()
            Case 4
                deletePatient()
            Case 5
                printFemalePatient()
            Case 6
                Exit While
            Case Else
                Console.WriteLine("錯誤，請輸入 1 ~ 6")
        End Select
    End While
End Sub
```

休息一下~



Ch9.遞迴(recursive)

- ▶ 副程式可以被呼叫，當副程式裡又呼叫自己時，這種情況稱為遞迴。
- ▶ 有些問題用遞迴可以得到更簡潔的程式，但不一定會有高效率。
- ▶ 使用遞迴時要注意：
 - ▶ 1. 傳入的參數：要能符合要解決的問題。
 - ▶ 2. 終止條件：錯誤或沒有終止條件，將造成無盡迴圈甚至當機
- ▶ 經典題型：最大公因數 (GCD)、費波納西數列 (Fibonacci Sequence)、河內塔 (Hanoi Tower)、N 個字元的排列組合等。



Ch9.遞迴(recursive)

- ▶ 求 $1 + 2 + 3 + \dots + N$ 的和。

終止條件

結束呼叫，
回傳確定值

傳遞參數，呼叫自己

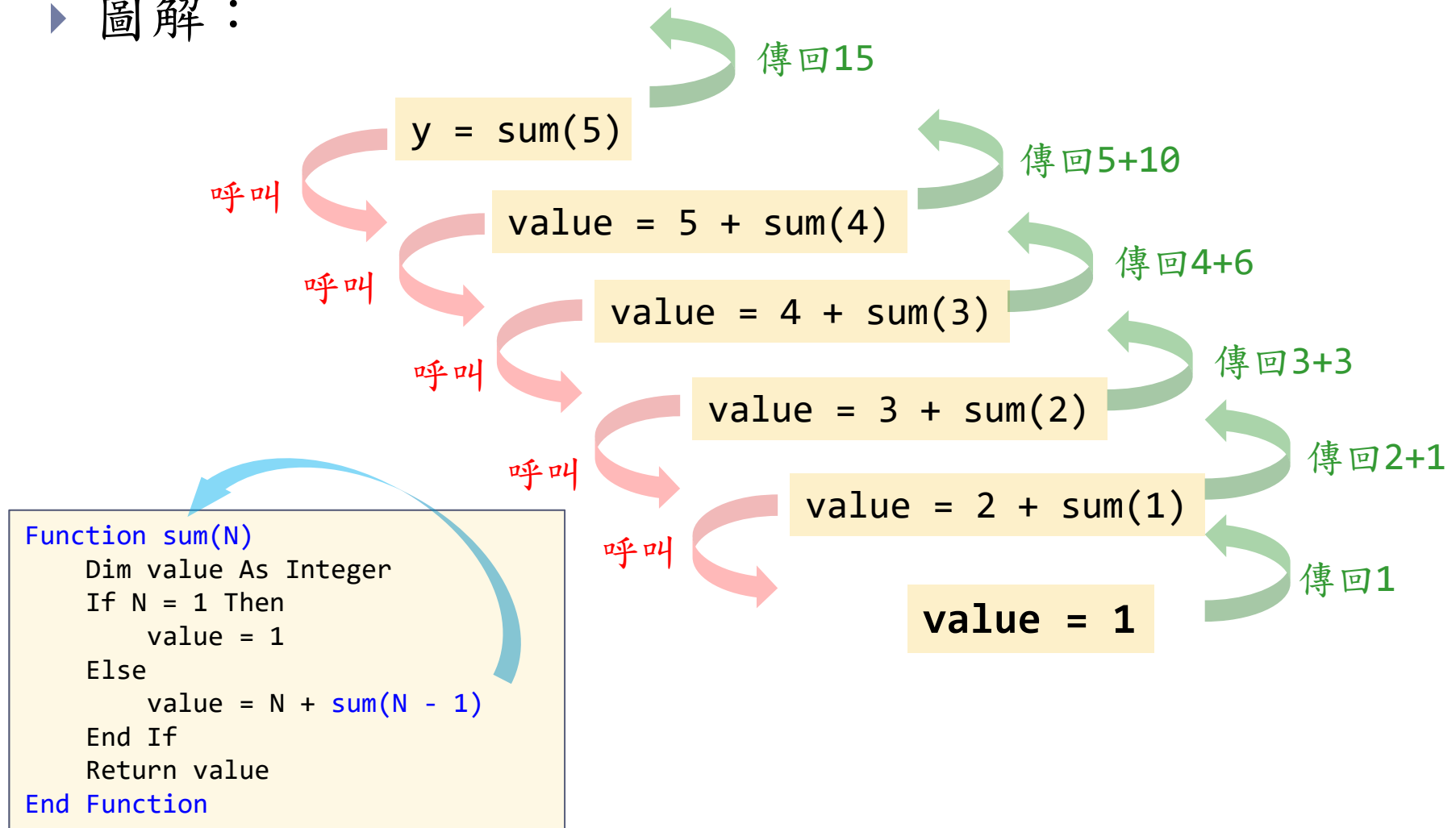
```
Function sum(N)
    Dim value As Integer
    If N = 1 Then
        value = 1
    Else
        value = N + sum(N - 1)
    End If
    Return value
End Function

Sub Main()
    Dim x, y As Integer
    Console.WriteLine("計算1到N的總和，請輸入N:")
    x = Console.ReadLine
    y = sum(x)
    Console.WriteLine("1到" & x & "的總和為" & y)
    Console.Read()
End Sub
```

- ▶ 假設輸入5

Ch9.遞迴(recursive)

► 圖解：



Ch9.遞迴(recursive)

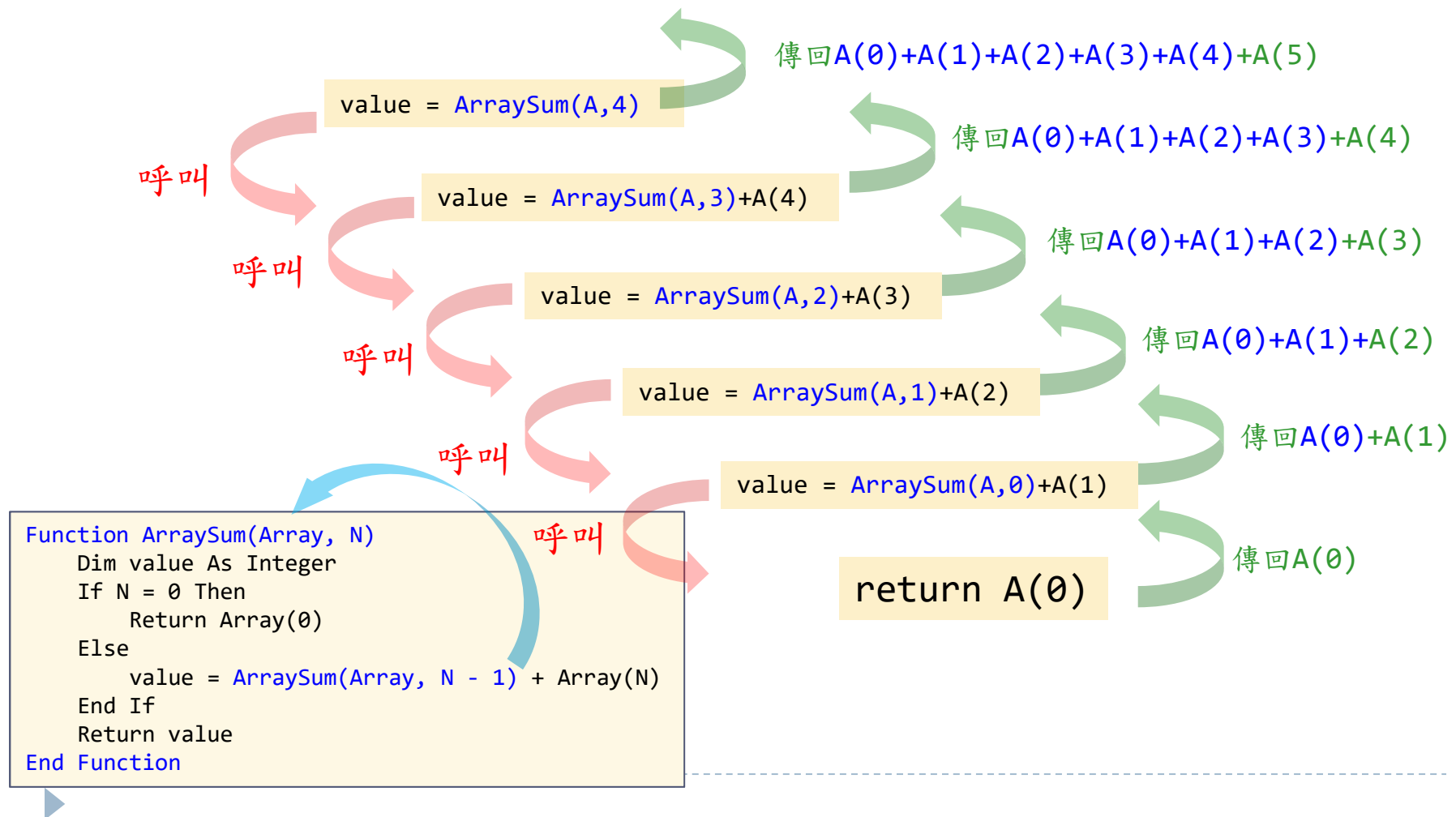
- ▶ 求陣列A中所有數值的和。

```
Function ArraySum(Array, N)
    Dim value As Integer
    If N = 0 Then
        Return Array(0)
    Else
        value = ArraySum(Array, N - 1) + Array(N)
    End If
    Return value
End Function

Sub Main()
    Dim A(), N As Integer
    Console.Write("請輸入陣列大小:")
    N = Console.ReadLine
    ReDim A(N)
    For i = 0 To N - 1
        Console.Write("請輸入A(" & i & ")的值:")
        A(i) = Console.ReadLine
    Next
    Console.WriteLine("Array內所有數值的總和為 " & ArraySum(A, N - 1))
    Console.Read()
End Sub
```

Ch9.遞迴(recursive)

► 圖解：(假設N=5)



Ch9.遞迴(recursive)

▶ 計算N!。

```
Function factorial(N)
    If N = 1 Then
        Return 1
    Else
        Return N * factorial(N - 1)
    End If
End Function

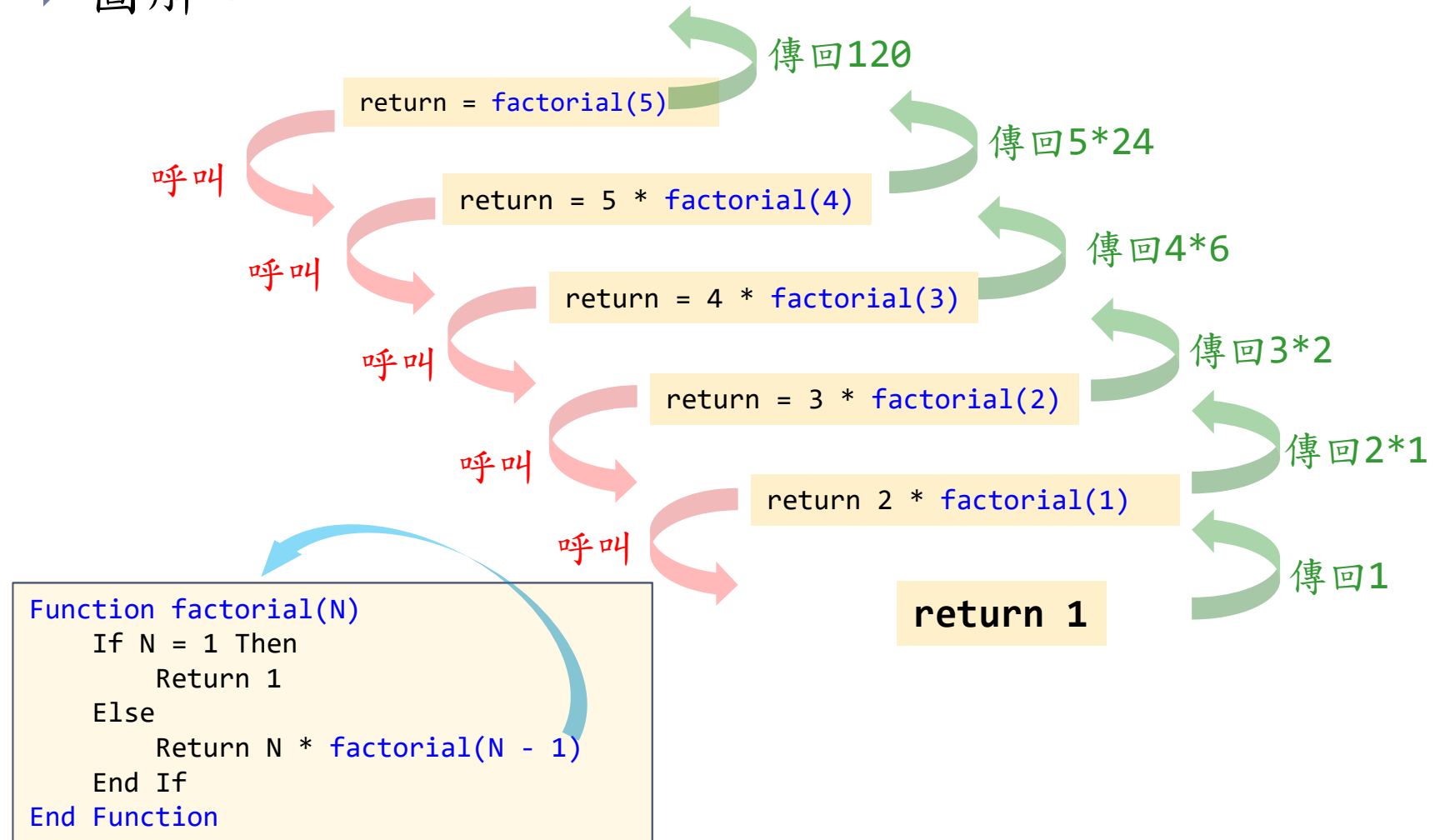
Sub Main()
    Dim x As Integer
    Console.Write("要計算幾階層?: ")
    x = Console.ReadLine
    Console.WriteLine(x & "! = " & factorial(x))
    Console.Read()
End Sub
```

▶ 假設輸入5



Ch9.遞迴(recursive)

► 圖解：



Ch9.遞迴(recursive)

- ▶ 求費氏數列F(n)。

```
Function F(N)
    Dim val As Integer
    If N = 1 Then
        Return 0
    ElseIf N = 2 Then
        Return 1
    Else
        val = F(N - 1) + F(N - 2)
    End If
    Return val
End Function

Sub Main()
    Dim x As Integer
    Console.Write("請輸入求F(N)的N為: ")
    x = Console.ReadLine
    Console.WriteLine("F(" & x & ") = " & F(x))
    Console.Read()
End Sub
```

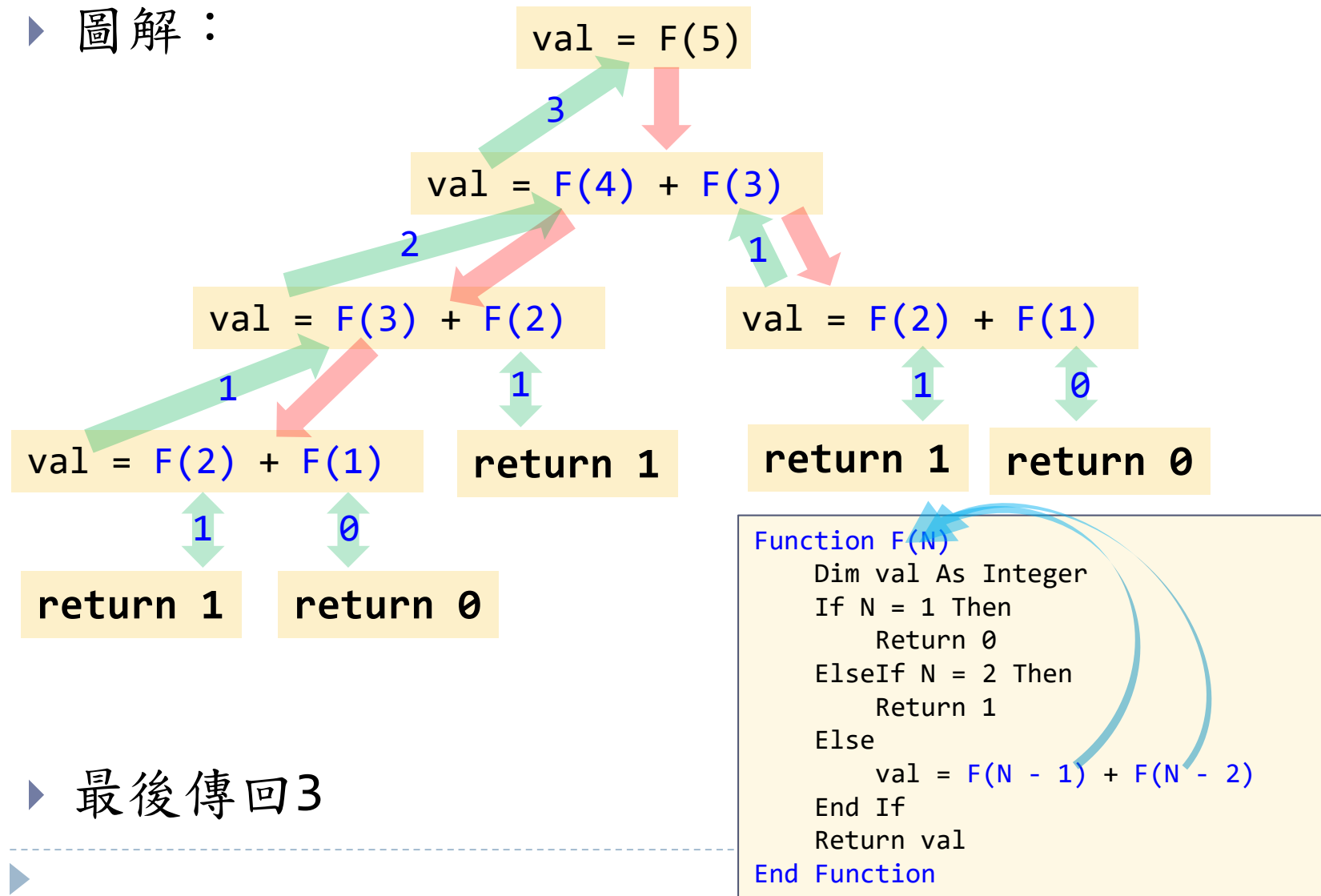
終止條件1

終止條件2

- ▶ 假設輸入5

Ch9.遞迴(recursive)

► 圖解：



► 最後傳回3

Ch9. 習題

- ▶ 9-1. 以遞迴方式求 $1+b+b^2+\dots+b^n$ 。
- ▶ 9-2. 以遞迴方式求 $1\times 2+2\times 3+\dots+n(n+1)$ 。
- ▶ 9-3. 以遞迴方式求 $a+(a+b)+(a+2b)+\dots+(a+(n-1)b)$ 。
(注意 b 的係數在增加)



Ch9.遞迴(recursive)

- ▶ 9-1. 以遞迴方式求 $1+b+b^2+\dots+b^n$ 。

```
Function nF(b, n)
    If n = 0 Then
        Return 1
    Else
        Return b ^ n + nF(b, n - 1)
    End If
End Function

Sub Main()
    Dim b, n As Integer
    Console.Write("請輸入基底b:")
    b = Console.ReadLine
    Console.Write("請輸入次方n:")
    n = Console.ReadLine
    Console.WriteLine("1+b+b^2+.....+b^n= " & nF(b, n))
    Console.Read()
End Sub
```


Ch9.遞迴(recursive)

- ▶ 9-2. 以遞迴方式求 $1 \times 2 + 2 \times 3 + \dots + n(n+1)$ 。

```
Function f(n)
    If n = 2 Then
        Return 2
    Else
        Return n * (n - 1) + f(n - 1)
    End If
End Function
```

End Function

```
Sub Main()
    Dim n As Integer
    Console.Write("請輸入N:")
    n = Console.ReadLine
    Console.WriteLine("(1*2)+(2*3)+...+n*(n+1)= " & f(n + 1))
    Console.Read()
End Sub
```



Ch9.遞迴(recursive)

- ▶ 9-3. 以遞迴方式求 $a+(a+b)+(a+2b)+\dots+(a+(n-1)b)$ 。
(注意 b 的係數在增加)

```
Function f(a, b, n)
    If n = 0 Then
        Return a
    Else
        Return a + n * b + f(a, b, n - 1)
    End If
End Function

Sub Main()
    Dim a, b, n As Integer
    Console.Write("請輸入a:") : a = Console.ReadLine
    Console.Write("請輸入b:") : b = Console.ReadLine
    Console.Write("請輸入n:") : n = Console.ReadLine
    Console.WriteLine("結果為:" & f(a, b, n - 1))
    Console.Read()
End Sub
```

休息一下~



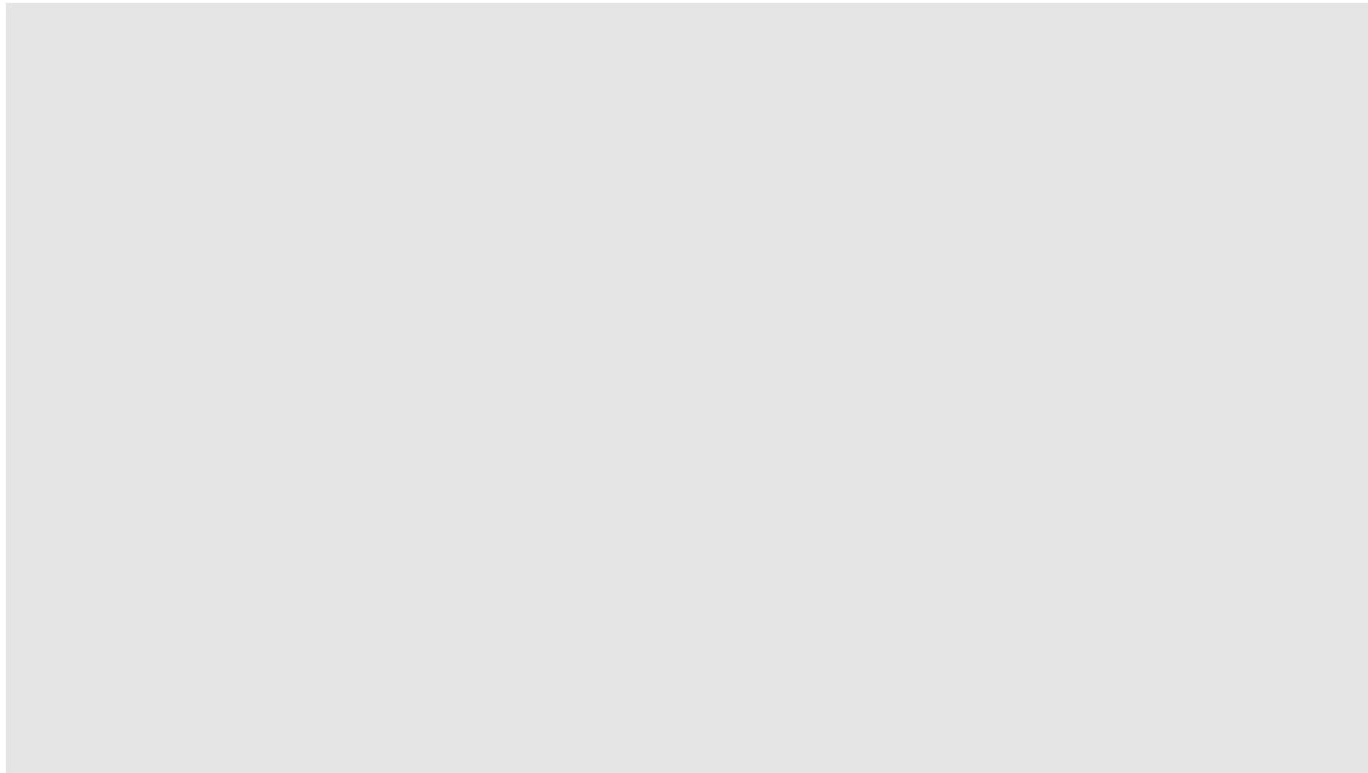
Ch.10 排序

- ▶ 排序(Sort)是將資料依某種規則重新安排其先後順序。
- ▶ 最常見的是依大小或字母排序。
- ▶ 常見排序方法：
 - ▶ 1. 氣泡排序法(Bubble Sort)
 - ▶ 2. 選擇排序法(Selection Sort)
 - ▶ 3. 插入排序法(Insertion Sort)
 - ▶ 4. 快速排序法(Quick Sort)
 - ▶ 5. 合併排序法(Merge Sort)
- ▶ 不同的排序方法效率不同，
應用的地方也不同。



Ch10. 氣泡排序法(Bubble Sort)

- ▶ 是一種簡單的排序演算法。它重複地走訪過要排序的數列，每次比較相鄰的兩個元素，如果他們的順序錯誤就把他們交換過來。



Ch10. 氣泡排序法(Bubble Sort)

▶ 參考程式：

```
Sub Main() '主程式
    Dim A(), N As Integer
    Console.Write("請輸入陣列大小:")
    N = Console.ReadLine
    ReDim A(N)
    For i = 1 To N '輸入數字並存入陣列
        Console.Write("請輸入第" & i & "個數字:")
        A(i) = Console.ReadLine
    Next
    For i = 1 To N '呼叫排序副程式
        printArray(A) '每一巡迴就印出陣列現況
        bubble(A, i, N) '呼叫氣泡排序
    Next
    printArray(A) '印出陣列最後結果
    Console.Read()
End Sub
```

Ch10. 氣泡排序法(Bubble Sort)

▶ 參考程式：

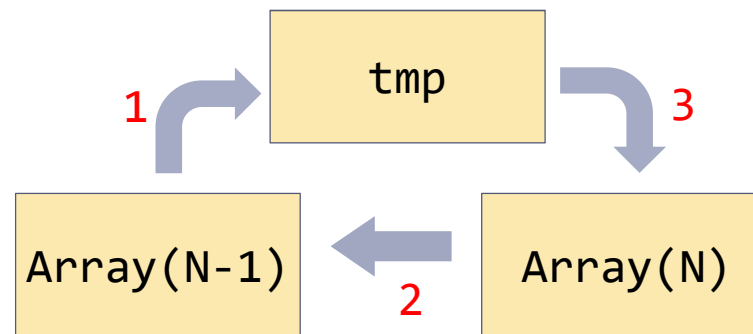
```
Sub bubble(Array, i, N) '氣泡排序一個循環
    Dim tmp As Integer
    While N > i
        If Array(N - 1) > Array(N) Then
            tmp = Array(N - 1)
            Array(N - 1) = Array(N)
            Array(N) = tmp
        End If
        N = N - 1
    End While
End Sub

Sub printArray(Array) '印出陣列現況
    For i = 1 To UBound(Array)
        Console.Write(Array(i) & vbTab)
    Next
    Console.WriteLine()
End Sub
```

Ch10. 氣泡排序法(Bubble Sort)

- ▶ 兩變數互換需要藉助另一個變數當中介來交換兩變數。

```
If Array(N - 1) > Array(N) Then  
    tmp = Array(N - 1)  
    Array(N - 1) = Array(N)  
    Array(N) = tmp  
End If
```



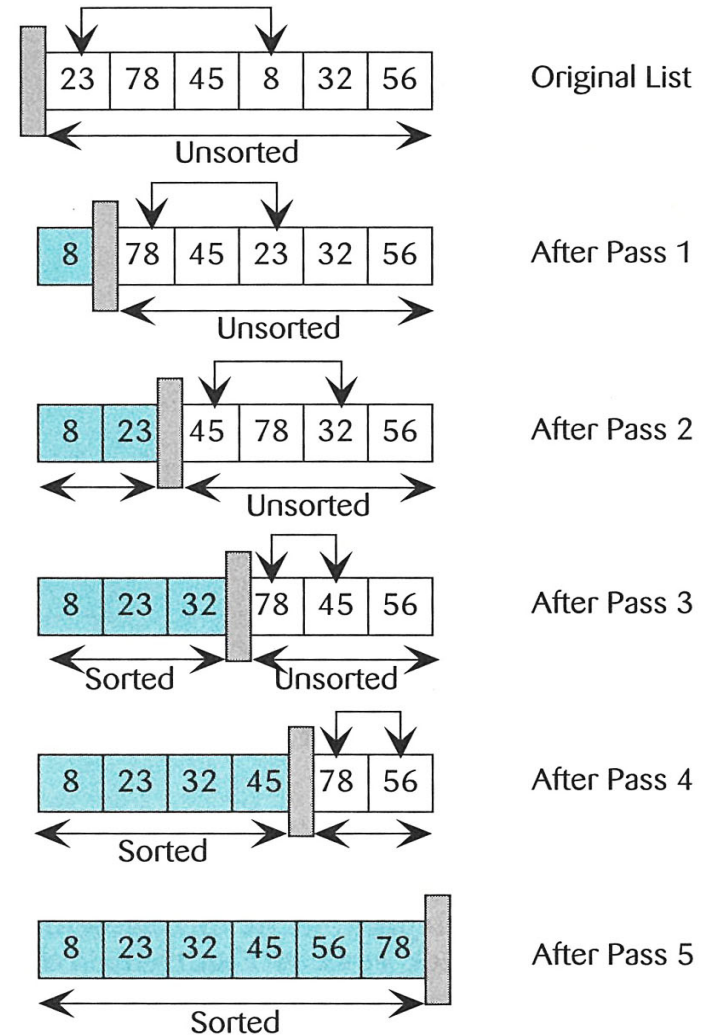
Ch10. 氣泡排序法(Bubble Sort)

► 執行結果：

```
file:///C:/Users/Orion Liu/AppData/Local/Temporary Pr...
請輸入陣列大小:6
請輸入第1個數字:9
請輸入第2個數字:4
請輸入第3個數字:8
請輸入第4個數字:2
請輸入第5個數字:7
請輸入第6個數字:3
9      4      8      2      7      3
2      9      4      8      3      7
2      3      9      4      8      7
2      3      4      9      7      8
2      3      4      7      9      8
2      3      4      7      8      9
2      3      4      7      8      9
```

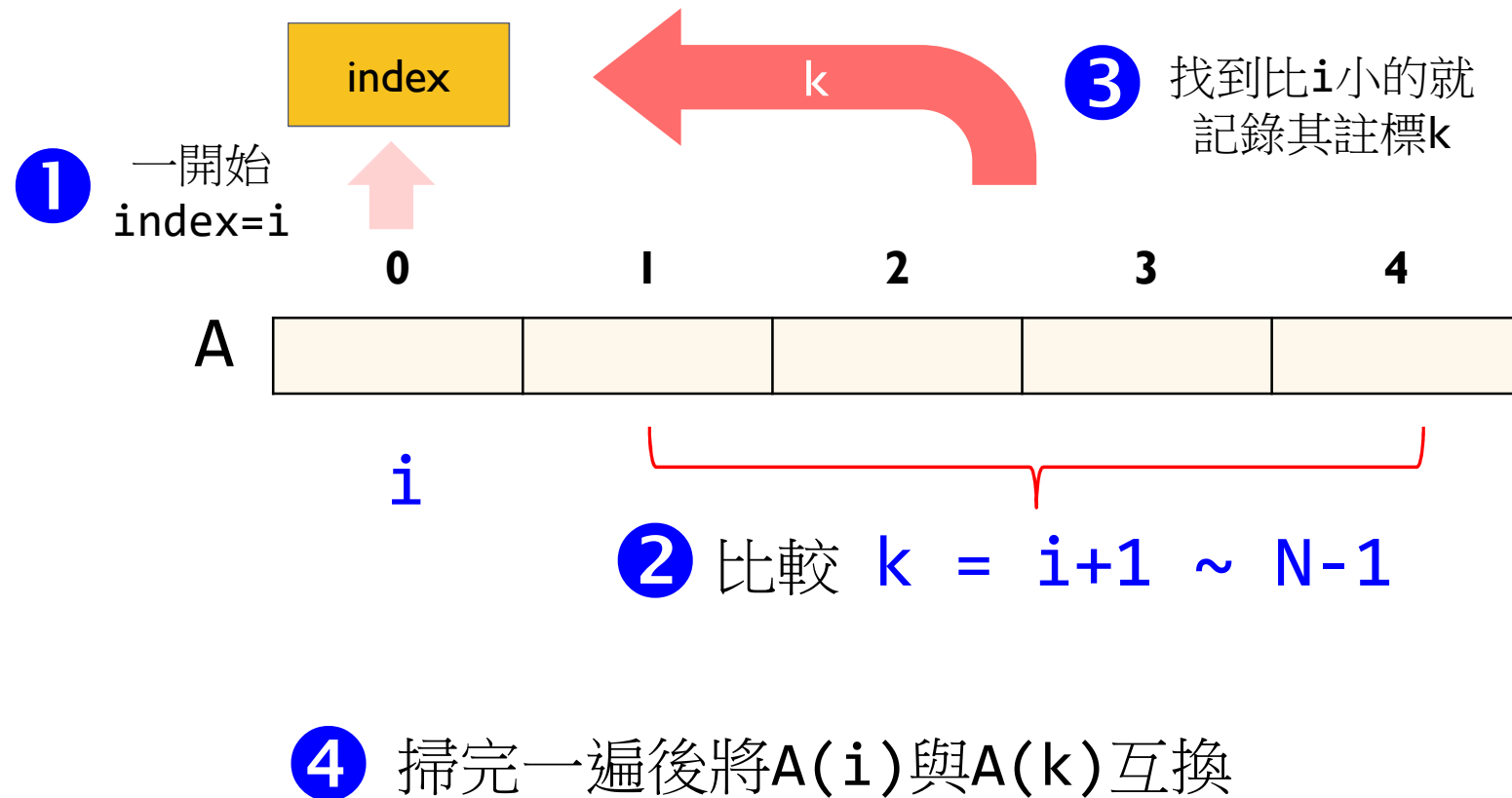
Ch10.選擇排序法(Selection Sort)

- ▶ 先假設第一個元素是最小的，依序向後掃描個元素，若有更小的就記錄起來，最後和第一個元素互換位置，完成一個巡迴，然後第二個元素，依此類推。



Ch10.選擇排序法(Selection Sort)

► 圖示說明：



Ch10.選擇排序法(Selection Sort)

▶ 參考程式：

```
Sub Main()    '主程式
    Dim A(), N As Integer
    Console.Write("請輸入陣列大小:")
    N = Console.ReadLine
    ReDim A(N)
    For i = 1 To N    '輸入數字並存入陣列
        Console.Write("請輸入第" & i & "個數字:")
        A(i) = Console.ReadLine
    Next
    For i = 1 To N    '呼叫排序副程式
        printArray(A)    '每一巡迴就印出陣列現況
        selection(A, i, N)    '呼叫選擇排序
    Next
    printArray(A)    '印出陣列最後結果
    Console.Read()
End Sub
```

Ch10.選擇排序法(Selection Sort)

▶ 參考程式：

```
Sub selection(Array, i, N)    '選擇排序一個循回
    Dim index, k, tmp As Integer
    index = i                '現在位置
    k = i + 1                '尋找範圍：現在位置之後到陣列結尾
    While k <= N              '尋找比現在元素更小的值，並記錄位置
        If Array(k) < Array(index) Then index = k
        k += 1
    End While
    If index <> i Then        '如果有記錄到更小的值就交換，否則略過
        tmp = Array(i)
        Array(i) = Array(index)
        Array(index) = tmp
    End If
End Sub


Sub printArray(Array)        '印出陣列現況
    For i = 1 To UBound(Array)
        Console.Write(Array(i) & vbTab)
    Next
    Console.WriteLine()
End Sub
```

Ch10.選擇排序法(Selection Sort)


► 執行結果：

請輸入陣列大小:6
請輸入第0個數字:9
請輸入第1個數字:4
請輸入第2個數字:7
請輸入第3個數字:2
請輸入第4個數字:8
請輸入第5個數字:3
[9, 4, 7, 2, 8, 3]

Pass 1 [2, 4, 7, 9, 8, 3]


Pass 2 [2, 3, 7, 9, 8, 4]


Pass 3 [2, 3, 4, 9, 8, 7]


Pass 4 [2, 3, 4, 7, 8, 9]


Pass 5 [2, 3, 4, 7, 8, 9]


最後一次無須交換

result [2, 3, 4, 7, 8, 9]



Ch10.插入排序法(Insertion Sort)

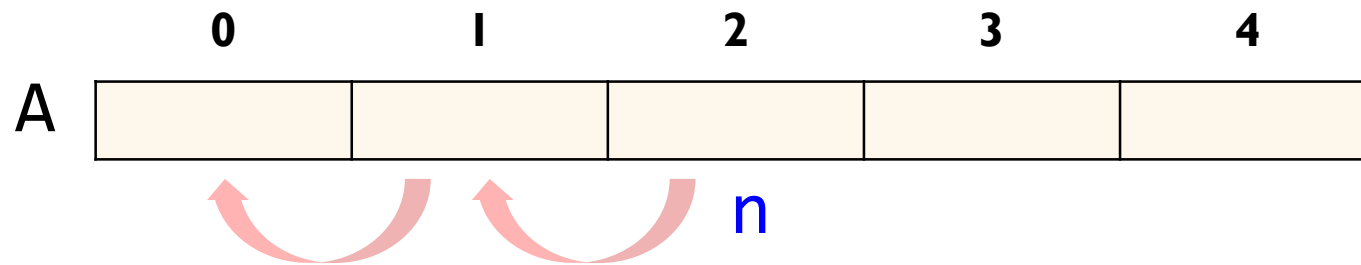
- ▶ 假設由小到大排序，每個元素和其左邊比較，若比左側小則交換，直到比左側大為止。

Insertion Sort							
35	97	19	4	57	27	98	36
0	1	2	3	4	5	6	7



Ch10.插入排序法(Insertion Sort)

► 交換原則：



比較 $A(n)$ 和 $A(n-1)$

- ❶ if $A(n) < A(n-1) \rightarrow$ 交換
- ❷ 直到 $A(1)$ 或 $A(n) \geq A(n-1) \rightarrow$ 停止
- ❸ 下一個，重複直到陣列最後一個元素



Ch10.插入排序法(Insertion Sort)

▶ 參考程式：

```
Sub Main() '主程式
    Dim A(), N, k As Integer
    Console.WriteLine("請輸入陣列大小:")
    N = Console.ReadLine
    ReDim A(N)
    For i = 1 To N '輸入數字並存入陣列
        Console.WriteLine("請輸入第" & i & "個數字:")
        A(i) = Console.ReadLine
    Next
    k = 2
    While k <= N
        printArray(A) '每一巡迴就印出陣列現況
        insert(A, k) '呼叫選擇排序
        k += 1
    End While
    printArray(A) '印出陣列最後結果
    Console.Read()
End Sub
```

Ch10.插入排序法(Insertion Sort)

▶ 參考程式：

```
Sub insert(Array, i) '插入排序法一個循環
    Dim k, tmp As Integer
    k = i
    While k >= 1 '從第i個位置開始向前依序比較
        If Array(k) < Array(k - 1) Then '比左側小則交換兩元素
            tmp = Array(k - 1)
            Array(k - 1) = Array(k)
            Array(k) = tmp
        Else '否則結束本pass
            Exit While
        End If
        k = k - 1
    End While
End Sub

Sub printArray(Array) '印出陣列現況
    For i = 1 To UBound(Array)
        Console.Write(Array(i) & vbTab)
    Next
    Console.WriteLine()
End Sub
```

Ch10.插入排序法(Insertion Sort)

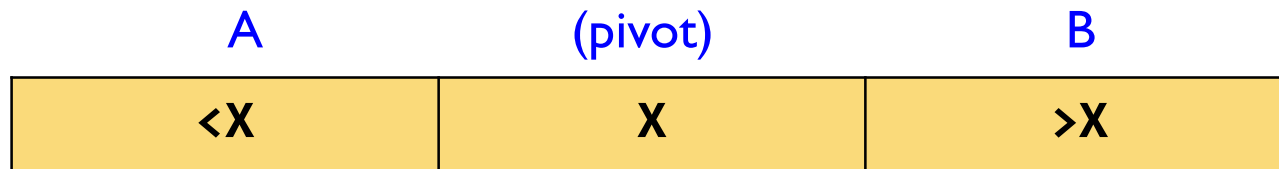
▶ 執行結果：

請輸入陣列大小:6
請輸入第0個數字:9
請輸入第1個數字:4
請輸入第2個數字:1
請輸入第3個數字:5
請輸入第4個數字:13
請輸入第5個數字:8

Pass 1 [9, 4, 1, 5, 13, 8]
Pass 2 [4, 9, 1, 5, 13, 8]
Pass 3 [1, 4, 9, 5, 13, 8]
Pass 4 [1, 4, 5, 9, 13, 8] 無須交換
Pass 5 [1, 4, 5, 9, 13, 8]
result [1, 4, 5, 8, 9, 13]

Ch10.快速排序法(Quick Sort)

- ▶ 快速排序法的原理是選定一個數 X (稱「基準」pivot)，然後設法將數列分成三份，如圖：



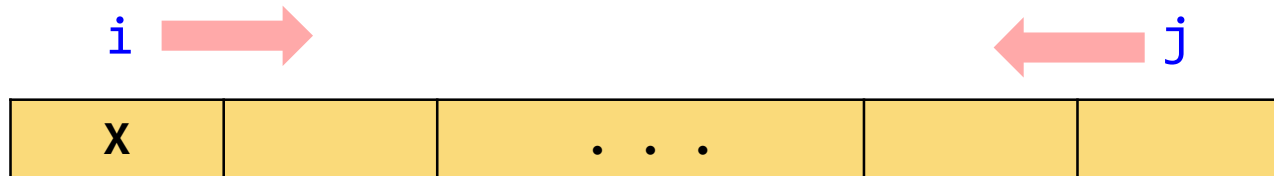
- ▶ 使用遞迴，以一樣的方式再分別處理A段和B段，直到所有數字都處理完畢。
- ▶ 遞迴到最底部時，數列的大小是零或一，也就是已經排序好了。
- ▶ 快速排序通常明顯比其他演算法更快。



Ch10.快速排序法(Quick Sort)

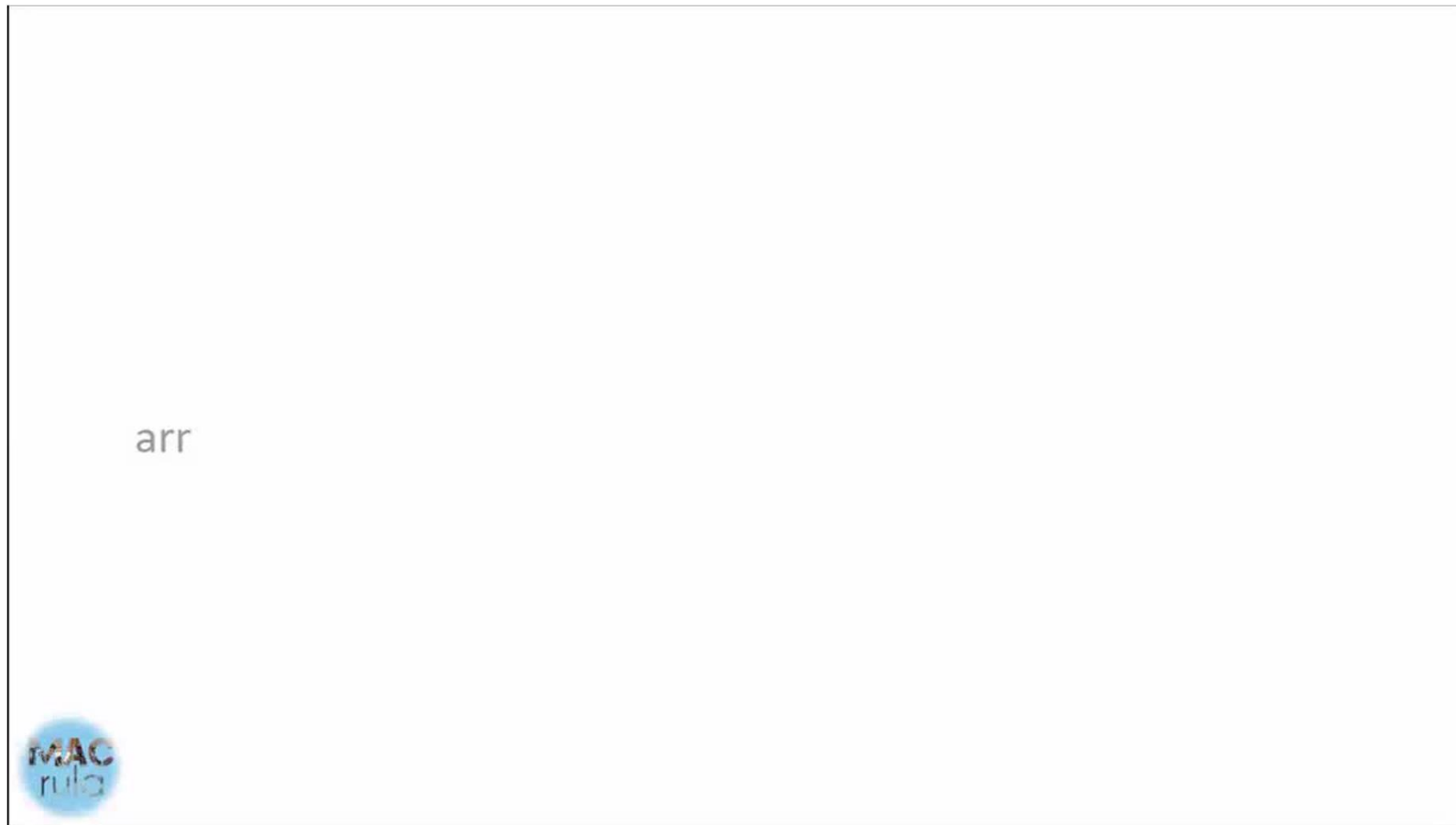
▶ 演算過程如下：

- ▶ 1. 令數列中最左邊的數為 $X(\text{pivot})$ 。
- ▶ 2. 設立兩個指標 i 和 j 。
- ▶ 3. 將 i 往右移，直到找到第一個 $A(i) \geq X$ 為止。
- ▶ 4. 將 j 往左移，直到找到第一個 $A(j) \leq X$ 為止。
- ▶ 5. 將 $A(i)$ 與 $A(j)$ 互換。
- ▶ 6. 重複以上動作直到 $i \geq j$ 為止。
- ▶ 7. 將 X 與 $A(i)$ 互換。



Ch10.快速排序法(Quick Sort)

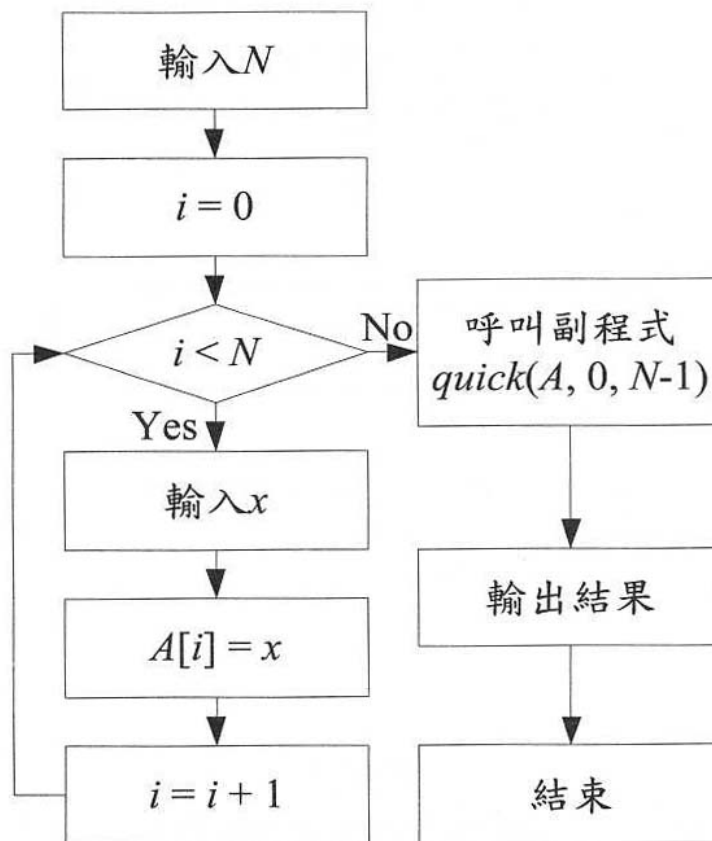
▶ 原理示範說明：



Ch10.快速排序法(Quick Sort)

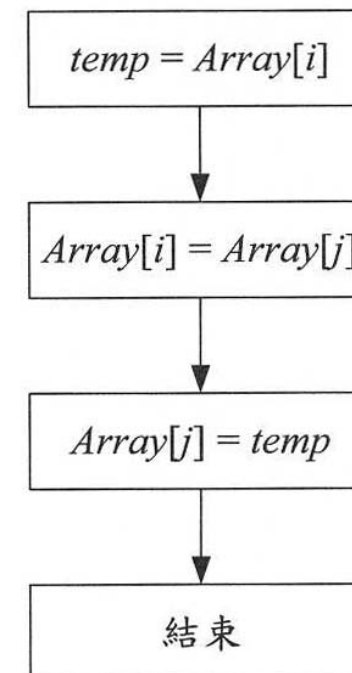
► 流程圖：

主程式



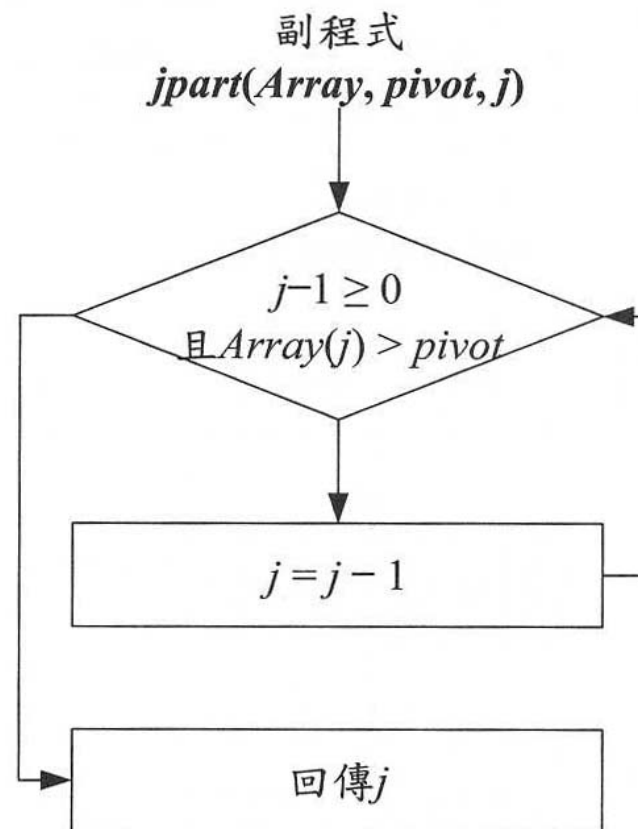
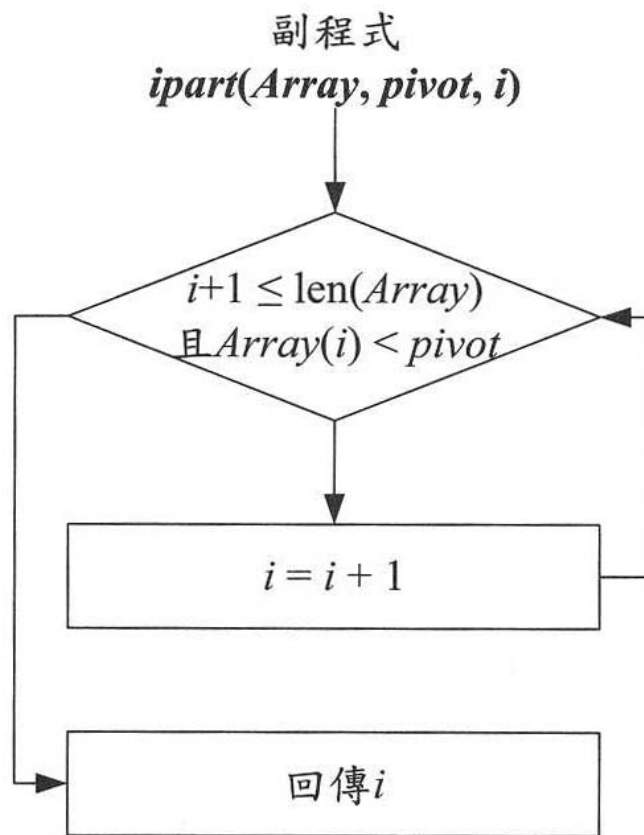
副程式

swap(Array, i, j)



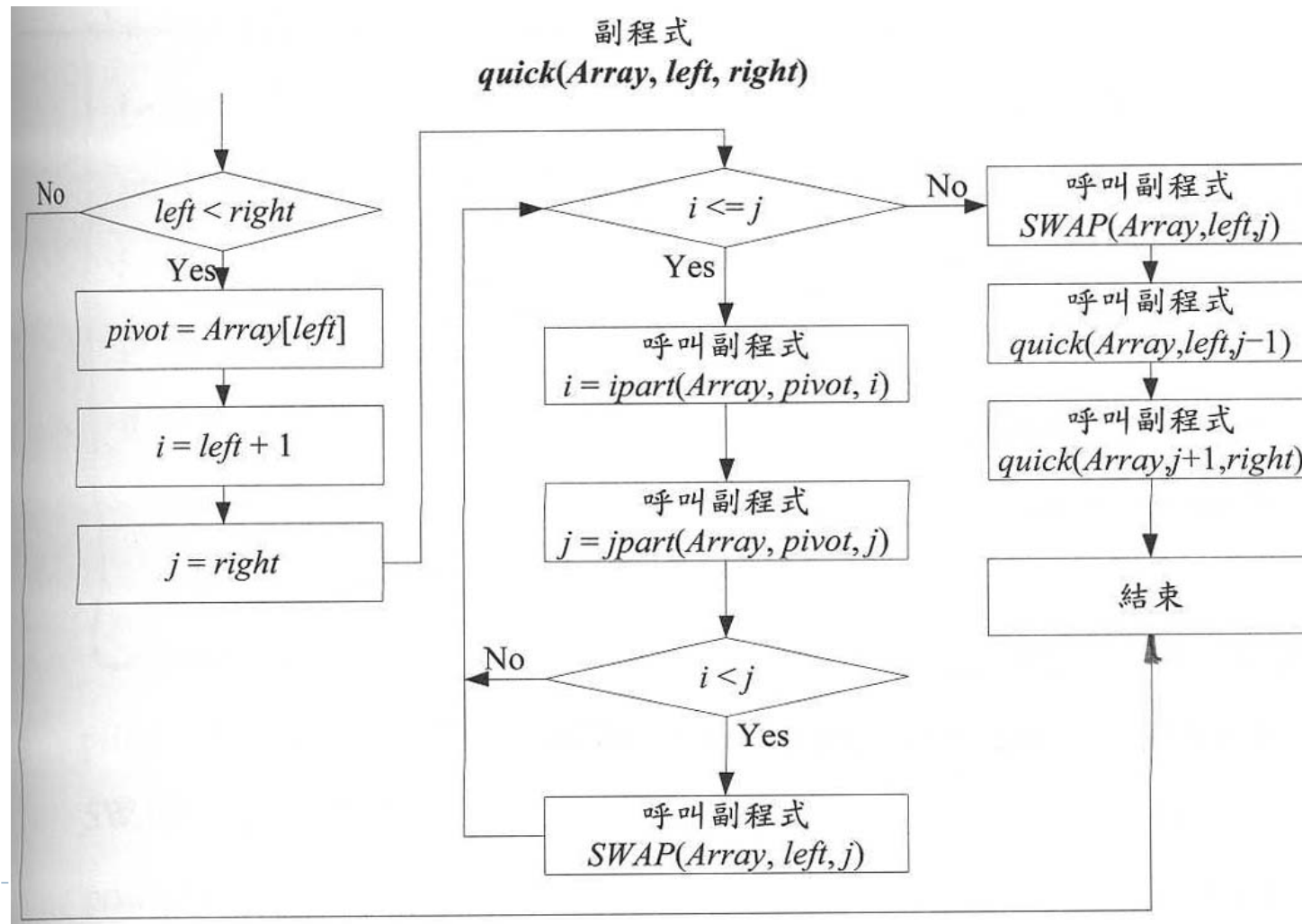
Ch10.快速排序法(Quick Sort)

► 流程圖：



Ch10.快速排序法(Quick Sort)

► 流程圖：



Ch10.快速排序法(Quick Sort)

▶ 參考程式：

```
Sub quick(Array, left, right) '快速排序，程式較冗長是因為要印出執行的過程
    Dim pivot, i, j As Integer
    Console.WriteLine("Q: left= " & left & ",right= " & right)
    If left < right Then
        pivot = Array(left)
        Console.WriteLine("pivot = A(" & left & ") = " & pivot)
        i = left + 1
        j = right
        While i <= j
            i = ipart(Array, pivot, i)
            Console.WriteLine("i = " & i)
            j = jpart(Array, pivot, j)
            Console.WriteLine("j = " & j)
            If i < j Then
                Console.WriteLine("i<j，執行交換A(" & i & ")和A(" & j & ")")
                SWAP(Array, i, j)
                printArray(Array)
            End If
        End While
        Console.WriteLine("交換pivot和A(" & j & ")")
        SWAP(Array, left, j)
        printArray(Array)
        quick(Array, left, j - 1)
        quick(Array, j + 1, right)
    End If
End Sub
```

Ch10.快速排序法(Quick Sort)

▶ 參考程式：

'交換兩元素

```
Sub SWAP(Array, i, j)
    Dim temp As Integer
    temp = Array(i)
    Array(i) = Array(j)
    Array(j) = temp
End Sub
```

'將i往右移，直到找到第一個A(i)>=X為止

```
Function ipart(Array, pivot, i)
    While i + 1 <= UBound(Array) And Array(i) < pivot
        i += 1
    End While
    Return i
End Function
```

'將j往左移，直到找到第一個A[j]<=X為止

```
Function jpart(Array, pivot, j)
    While j - 1 >= 0 And Array(j) > pivot
        j -= 1
    End While
    Return j
End Function
```

Ch10.快速排序法(Quick Sort)

▶ 參考程式：

```
Sub printArray(A) '印出陣列現況
    For i = 0 To UBound(A) - 1
        Console.Write(A(i) & vbTab)
    Next
    Console.WriteLine()
End Sub

Sub Main() '主程式
    Dim A(), N As Integer
    Console.Write("請輸入陣列大小:")
    N = Console.ReadLine
    ReDim A(N)
    For i = 0 To N - 1 '輸入陣列資料
        Console.Write("請輸入第" & i & "個數字:")
        A(i) = Console.ReadLine
    Next
    printArray(A) '印出原陣列
    quick(A, 0, N - 1) '呼叫快速排序
    printArray(A) '印出排序結果
    Console.Read()
End Sub
```

Ch10.快速排序法(Quick Sort)

▶ 執行結果：

```
file:///C:/Users/Orion Liu/AppData/Local/Temporary Projects/ConsoleApplication4/bin/Debug/ConsoleA...
請輸入陣列大小:10
請輸入第0個數字:11
請輸入第1個數字:20
請輸入第2個數字:3
請輸入第3個數字:15
請輸入第4個數字:16
請輸入第5個數字:24
請輸入第6個數字:9
請輸入第7個數字:31
請輸入第8個數字:25
請輸入第9個數字:28
11    20    3    15    16    24    9    31    25    28
Q: left= 0,right= 9
pivot = A(0) = 11
i = 1
j = 6
i<j, 執行交換A(1)和A(6)
11    9    3    15    16    24    20    31    25    28
i = 3
j = 2
交換pivot和A(2)
3    9    11    15    16    24    20    31    25    28
Q: left= 0,right= 1
pivot = A(0) = 3
i = 1
j = 0
交換pivot和A(0)
3    9    11    15    16    24    20    31    25    28
Q: left= 0,right= -1
Q: left= 1,right= 1
Q: left= 3,right= 9
pivot = A(3) = 15
i = 4
j = 3
交換pivot和A(3)
3    9    11    15    16    24    20    31    25    28
```

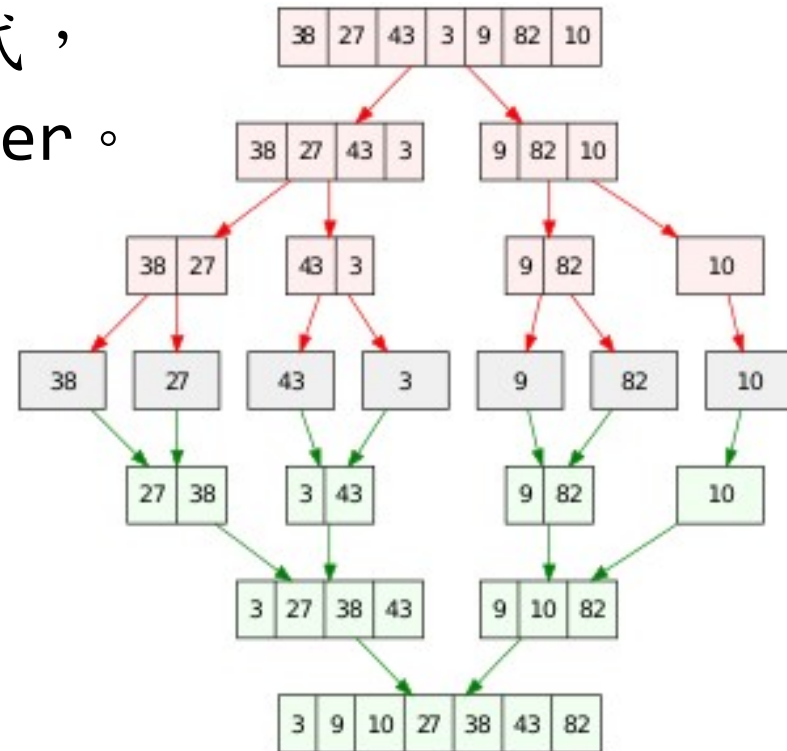
Ch10.快速排序法(Quick Sort)

► 執行結果：

```
file:///C:/Users/Orion Liu/AppData/Local/Temporary Projects/ConsoleApplication4/bin/Debug/ConsoleA...
交換pivot和A(3)
3      9      11      15      16      24      20      31      25      28
Q: left= 3,right= 2
Q: left= 4,right= 9
pivot = A(4) = 16
i = 5
j = 4
交換pivot和A(4)
3      9      11      15      16      24      20      31      25      28
Q: left= 4,right= 3
Q: left= 5,right= 9
pivot = A(5) = 24
i = 7
j = 6
交換pivot和A(6)
3      9      11      15      16      20      24      31      25      28
Q: left= 5,right= 5
Q: left= 7,right= 9
pivot = A(7) = 31
i = 10
j = 9
交換pivot和A(9)
3      9      11      15      16      20      24      28      25      31
Q: left= 7,right= 8
pivot = A(7) = 28
i = 9
j = 8
交換pivot和A(8)
3      9      11      15      16      20      24      25      28      31
Q: left= 7,right= 7
Q: left= 9,right= 8
Q: left= 10,right= 9
3      9      11      15      16      20      24      25      28      31
```

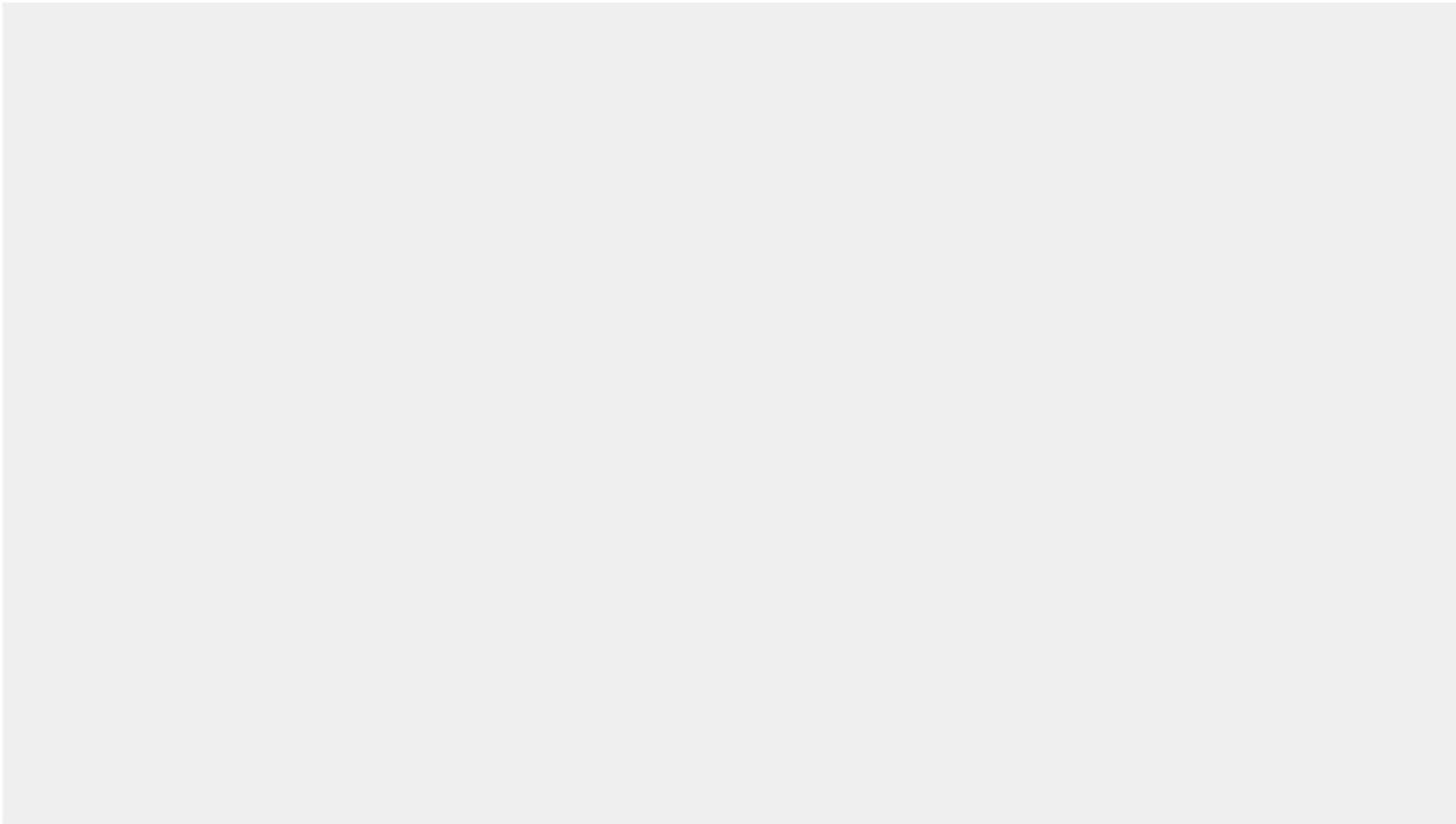
Ch10. 合併排序法(Merge Sort)

- ▶ 合併排序法是依據”合併”來排序的，假設我們有兩個已經排好的數列，我們就可以很容易的將兩個數列合併成一個排序好的數列。
- ▶ 採用先打散，在合併的方式，稱為divide and conquer。
- ▶ 一般都以遞迴來實現。



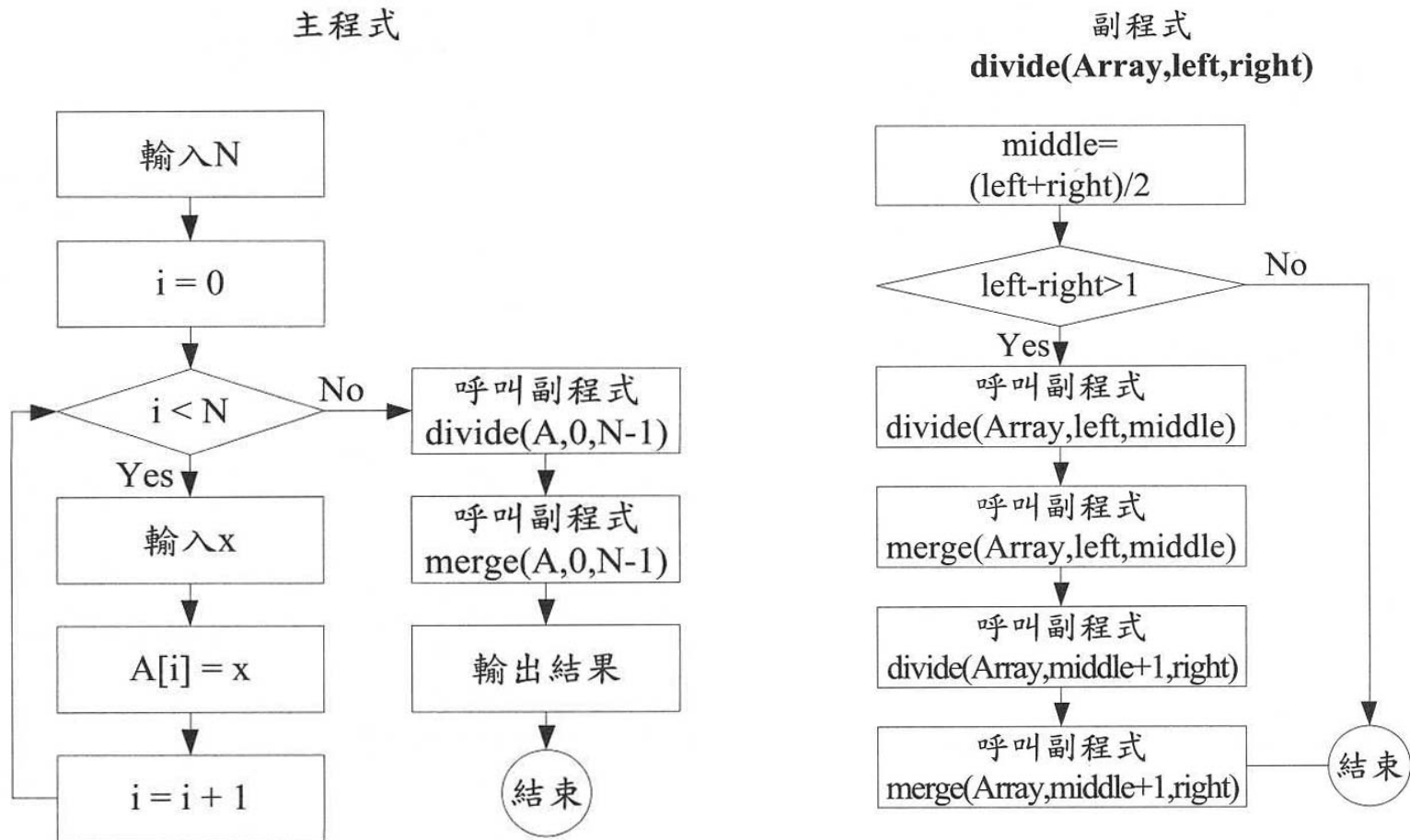
Ch10. 合併排序法(Merge Sort)

▶ 原理示範說明：



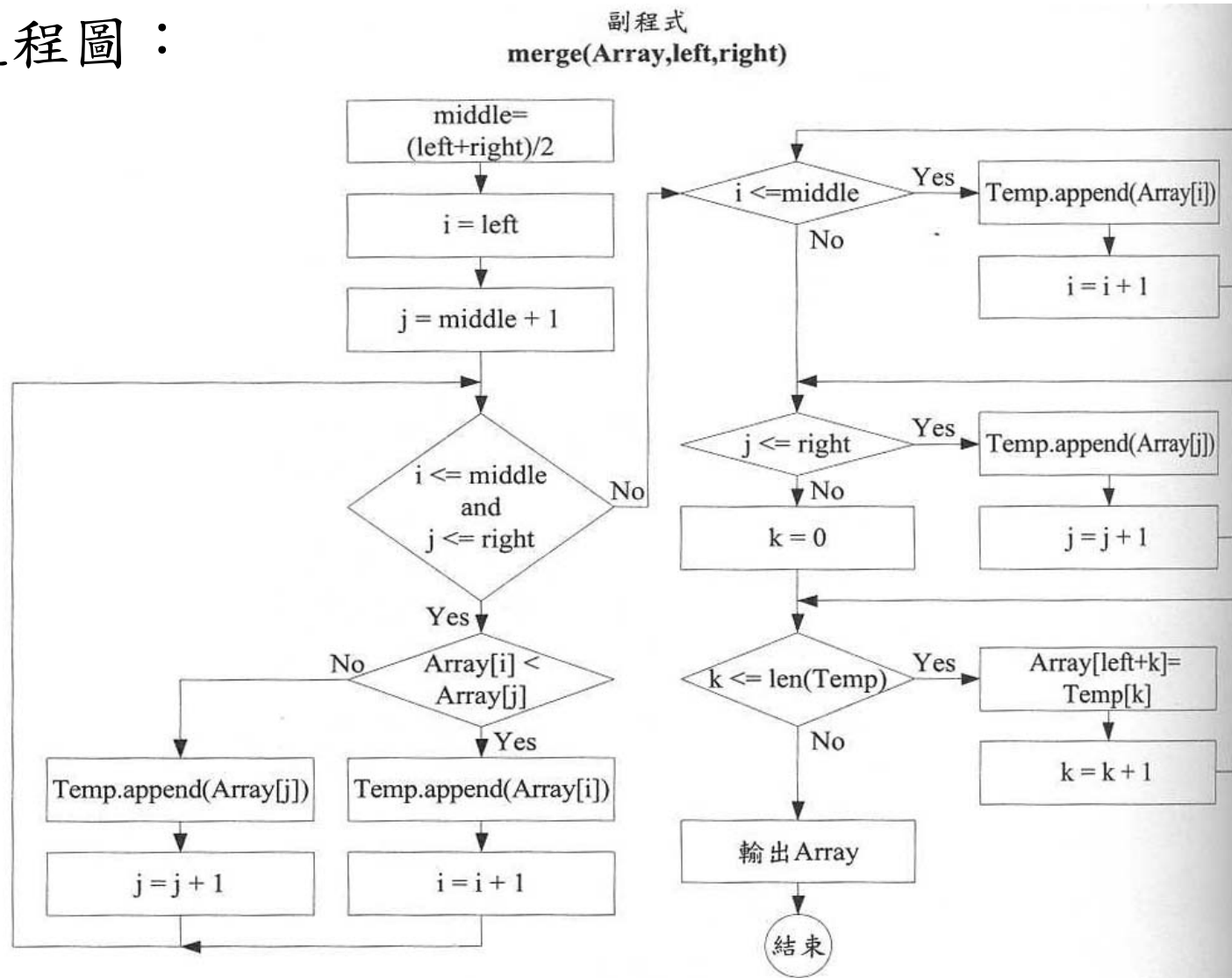
Ch10. 合併排序法(Merge Sort)

► 流程圖：



Ch10. 合併排序法(Merge Sort)

► 流程圖：



Ch10. 合併排序法(Merge Sort)

▶ 參考程式：

```
Sub divide(Array, left, right) '分割副程式
    Dim middle, leftRange As Integer
    Console.WriteLine("D: left= " & left & ", right= " & right)
    middle = Int((left + right) / 2)
    If (right - left) > 1 Then
        divide(Array, left, middle)
        merge(Array, left, middle)
        divide(Array, middle + 1, right)
        merge(Array, middle + 1, right)
        '印出合併的提示訊息，左半邊要與右半邊合併
        Console.Write("接著將左邊(")
        leftRange = left
        While leftRange <= middle
            Console.Write(Array(leftRange))
            leftRange += 1
        End While
        Console.Write(")及右邊(")
        leftRange = middle + 1
        While leftRange <= right
            Console.Write(Array(leftRange))
            leftRange += 1
        End While
        Console.WriteLine(")合併...")
    End If
End Sub
```

Ch10. 合併

▶ 參考程式：

```
Sub merge(Array, left, right) '合併副程式
    Dim Temp As New ArrayList '此處使用了ArrayList物件
    Dim middle, i, j, k, leftRange As Integer
    Console.WriteLine("M: left= " & left & ", right= " & right)
    middle = Int((left + right) / 2)
    i = left : j = middle + 1
    Console.WriteLine("i= " & i & " j= " & j)
    While i <= middle And j <= right
        If Array(i) < Array(j) Then
            Temp.Add(Array(i)) : i += 1
        Else
            Temp.Add(Array(j)) : j += 1
        End If
    End While
    printTemp(Temp)
    While i <= middle
        Temp.Add(Array(i)) : i += 1
    End While
    While j <= right
        Temp.Add(Array(j)) : j += 1
    End While
    For k = 0 To Temp.Count - 1
        Array(left + k) = Temp(k)
    Next
    printArray(Array) '將排序好的內容印出
    Console.WriteLine("(")
    leftRange = left
    While leftRange <= right
        Console.WriteLine(Array(leftRange)) : leftRange += 1
    End While
    Console.WriteLine(")排序好" & vbCrLf)
End Sub
```

Ch10. 合併排序法(Merge Sort)

▶ 參考程式：

```
Sub printArray(A) '印出排序陣列現況
    For i = 0 To UBound(A) - 1
        Console.Write(A(i) & " ")
    Next
End Sub

Sub printTemp(A) '印出Temp陣列現況
    Console.Write("Temp= ")
    For Each n In A
        Console.Write(n & " ")
    Next
    Console.WriteLine(")")
End Sub
```



Ch10.合併排序法(Merge Sort)

▶ 參考程式：

```
Sub Main() '主程式
    Dim A(), N As Integer
    Console.Write("請輸入陣列大小:")
    N = Console.ReadLine
    ReDim A(N)
    For i = 0 To N - 1 '輸入陣列資料
        Console.Write("請輸入第" & i & "個數字:")
        A(i) = Console.ReadLine
    Next
    Console.Write("排序前： ")
    printArray(A) '印出原陣列
    Console.WriteLine()
    divide(A, 0, N - 1) '呼叫分割
    merge(A, 0, N - 1) '呼叫合併
    Console.Write("排序完成： ")
    printArray(A) '印出排序結果
    Console.Read()
End Sub
```

Ch10.合併排序法(Merge Sort)

▶ 執行結果：

```
file:///C:/Users/Orion Liu...  — □ ×
請輸入陣列大小:7
請輸入第0個數字:6
請輸入第1個數字:3
請輸入第2個數字:9
請輸入第3個數字:5
請輸入第4個數字:2
請輸入第5個數字:7
請輸入第6個數字:1
排序前: 6 3 9 5 2 7 1
D: left= 0, right= 6
D: left= 0, right= 3
D: left= 0, right= 1
M: left= 0, right= 1
i= 0 j= 1
Temp= (3 )
3 6 9 5 2 7 1 (36)排序好

D: left= 2, right= 3
M: left= 2, right= 3
i= 2 j= 3
Temp= (5 )
3 6 5 9 2 7 1 (59)排序好

接著將左邊(36)及右邊(59)合併...
M: left= 0, right= 3
i= 0 j= 2
Temp= (3 5 6 )
3 5 6 9 2 7 1 (3569)排序好
```

```
選取 file:///C:/Users/Orio...  — □ ×
D: left= 4, right= 6
D: left= 4, right= 5
M: left= 4, right= 5
i= 4 j= 5
Temp= (2 )
3 5 6 9 2 7 1 (27)排序好

D: left= 6, right= 6
M: left= 6, right= 6
i= 6 j= 7
Temp= ( )
3 5 6 9 2 7 1 (1)排序好

接著將左邊(27)及右邊(1)合併...
M: left= 4, right= 6
i= 4 j= 6
Temp= (1 )
3 5 6 9 1 2 7 (127)排序好

接著將左邊(3569)及右邊(127)合併...
M: left= 0, right= 6
i= 0 j= 4
Temp= (1 2 3 5 6 7 )
1 2 3 5 6 7 9 (1235679)排序好

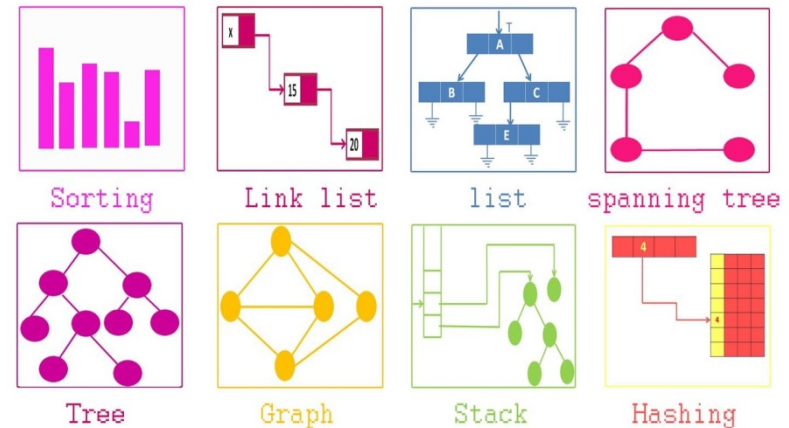
排序完成: 1 2 3 5 6 7 9
```


休息一下~



Ch11. 資料結構

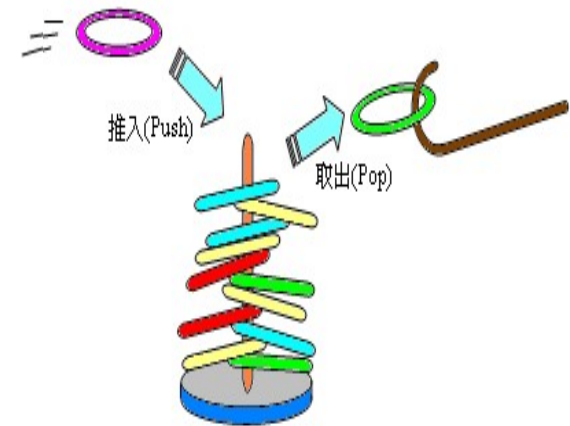
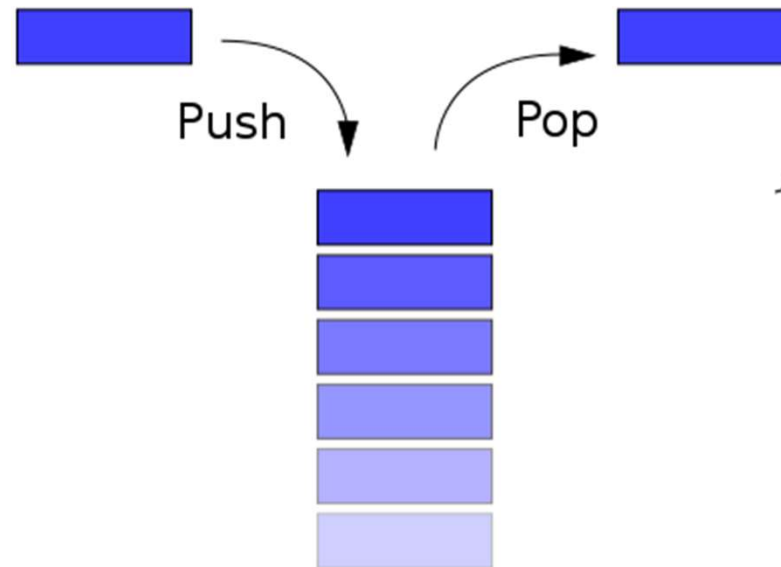
- ▶ 資料結構(data structure)是電腦中儲存、組織資料的方式。
- ▶ 正確的資料結構選擇可以提高演算法的效率。在電腦程式設計的過程中，選擇適當的資料結構是一項重要工作。
- ▶ 常見的資料結構：
 - ▶ 陣列 (Array)
 - ▶ 堆疊 (Stack)
 - ▶ 佇列 (Queue)
 - ▶ 連結串列 (Linked List)
 - ▶ 樹 (Tree)
 - ▶ 圖 (Graph)
 - ▶ 堆積 (Heap)
 - ▶ 雜湊表 (Hash)



By...naviskumarphotography.com

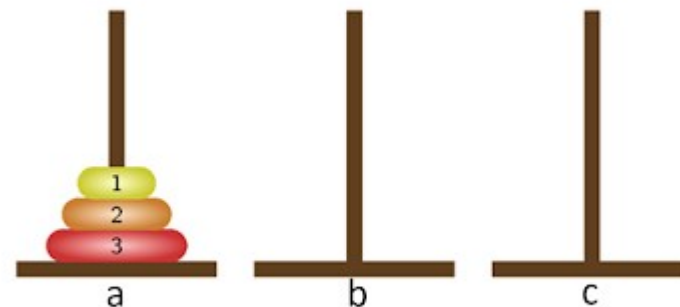
Ch11. 資料結構 – 堆疊(Stack)

- ▶ 在日常生活中，把物品(如餐盤、書本)由桌面一個一個向上疊放，取用時由最上面一個個向下拿去，這種觀念稱為堆疊(stack)
- ▶ 存入稱**PUSH**，取出稱**POP**



Ch11. 資料結構 – 堆疊(Stack)

- ▶ 堆疊觀念：**先進後出**(First In Last Out, **FILO**)
或稱為後進先出(Last In First Out, **LIFO**)
- ▶ 堆疊觀念在計算機中使用很廣，例如主副程式間訊息傳送，CPU中斷處理、遞迴程式的呼叫及返還
- ▶ 最著名的問題就是：河內塔



Ch11. 資料結構 – 堆疊(Stack)

▶ 使用堆疊結構，寫一程式，使用者可以有四種選擇：

- ▶ 1. 加入數字
- ▶ 2. 取出數字
- ▶ 3. 查看Stack
- ▶ 4. 離開程式

▶ (我們使用VB的Stack物件來實做。)

▶ 將資料y存入堆疊s，使用指令：

```
s.Push(y)
```

▶ 將最後一個資料pop出來：

```
y = s.Pop()
```



Ch11. 資料結構 – 堆疊(Stack)

▶ 參考程式：

```
Sub Main() '主程式
    Dim s As Stack(Of Integer) = New Stack(Of Integer)() '宣告一個堆疊物件S
    Dim x As Integer
    While True
        Console.WriteLine("[STACK] 1.加入數字 2.取出數字  
3.查看 4.離開程式 || 請選擇功能:")
        x = Console.ReadLine
        Select Case x
            Case 1 '加入一個資料到堆疊
                addNumberToStack(s)
            Case 2 '從堆疊取出一個資料
                popNumberFromStack(s)
            Case 3 '印出堆疊內容
                showStack(s)
            Case 4 '離開程式
                Exit While
        End Select
    End While
End Sub
```

Ch11. 資料結構 – 堆疊(Stack)

▶ 參考程式：

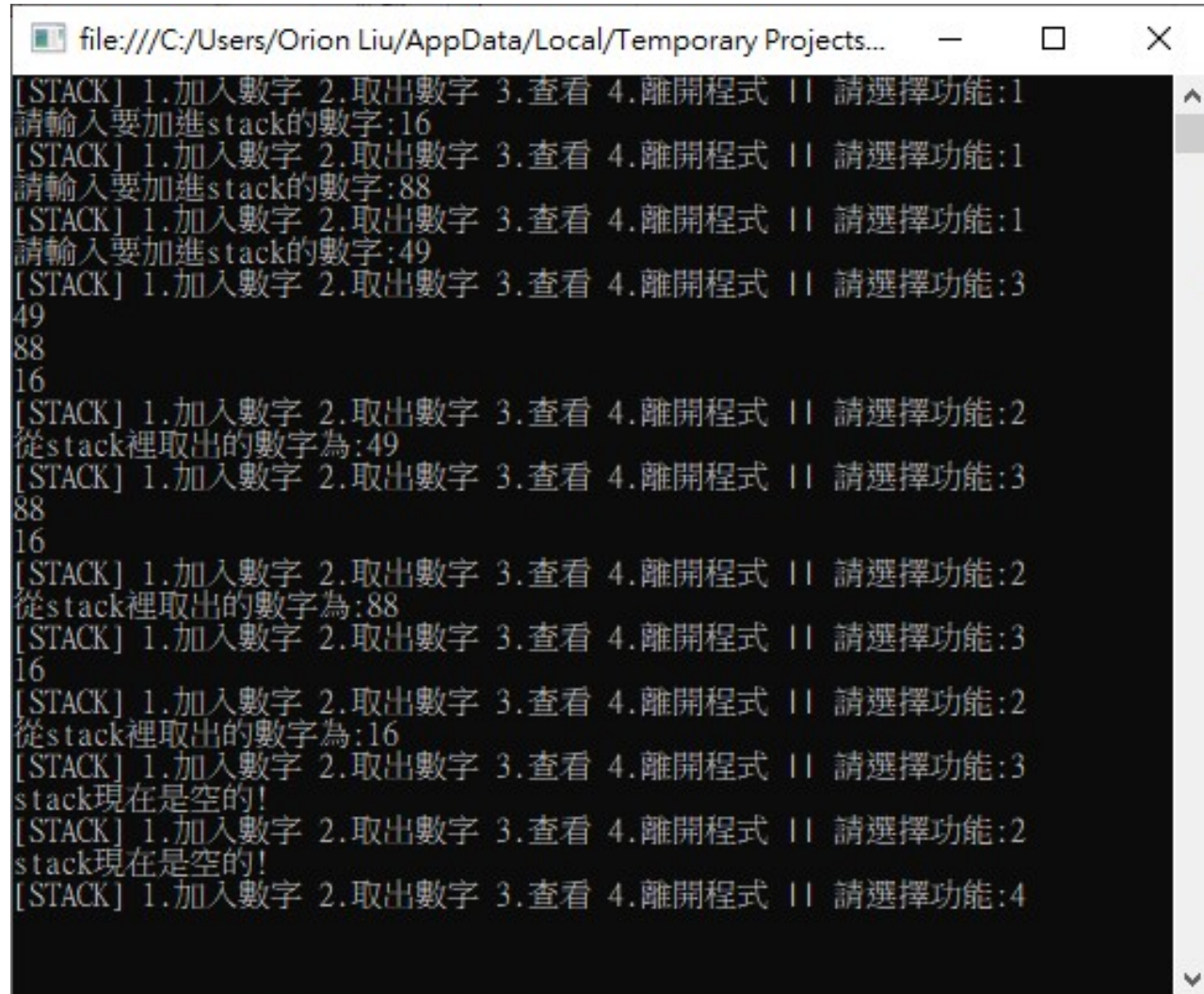
```
Sub addNumberToStack(s) '加入一個資料到堆疊
    Dim n As Integer
    Console.Write("請輸入要加進Stack的數字:")
    n = Console.ReadLine
    s.Push(n)
End Sub

Sub popNumberFromStack(s) '從堆疊取出一個資料
    Dim y As Integer
    If s.count > 0 Then
        y = s.Pop()
        Console.WriteLine("從Stack裡取出的數字為:" & y)
    Else
        Console.WriteLine("Stack現在是空的!")
    End If
End Sub

Sub showStack(s) '印出堆疊內容
    If s.count > 0 Then
        For Each element As Integer In s
            Console.WriteLine(element)
        Next
    Else
        Console.WriteLine("Stack現在是空的!")
    End If
End Sub
```

Ch11.資料結構 – 堆疊(Stack)

▶ 執行結果：



```
file:///C:/Users/Orion Liu/AppData/Local/Temporary Projects...  
[STACK] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:1  
請輸入要加進stack的數字:16  
[STACK] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:1  
請輸入要加進stack的數字:88  
[STACK] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:1  
請輸入要加進stack的數字:49  
[STACK] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:3  
49  
88  
16  
[STACK] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:2  
從stack裡取出的數字為:49  
[STACK] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:3  
88  
16  
[STACK] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:2  
從stack裡取出的數字為:88  
[STACK] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:3  
16  
[STACK] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:2  
從stack裡取出的數字為:16  
[STACK] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:3  
stack現在是空的!  
[STACK] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:2  
stack現在是空的!  
[STACK] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:4
```

Ch11. 資料結構 – 佇列(Queue)

- ▶ 顧名思義是一種像排隊一樣的概念
- ▶ 在這個模式下我們可以知道它是一種先進先出(First-In-First-Out, **FIFO**)的排程
- ▶ 存入稱**Enqueue**，取出稱**Dequeue**



Ch11. 資料結構 – 佇列(Queue)

▶ 使用佇列結構，寫一程式，使用者可以有四種選擇：

- ▶ 1. 加入數字
- ▶ 2. 取出數字
- ▶ 3. 查看Queue
- ▶ 4. 離開程式

▶ (我們使用VB的Queue物件來實做。)

▶ 將資料y存入佇列q，使用指令：

```
q.Enqueue(y)
```

▶ 將最前面的資料取出來：

```
y = q.Dequeue()
```



Ch11. 資料結構 – 佇列(Queue)

▶ 參考程式：

```
Sub Main() '主程式
    Dim q As Queue(Of Integer) = New Queue(Of Integer)() '宣告一個佇列物件q
    Dim x As Integer
    While True
        Console.WriteLine("[QUEUE] 1.加入數字 2.取出數字  
3.查看 4.離開程式 || 請選擇功能:")

        x = Console.ReadLine
        Select Case x
            Case 1 '加入一個資料到佇列
                addNumberToQueue(q)
            Case 2 '從佇列取出一個資料
                popNumberFromQueue(q)
            Case 3 '印出佇列內容
                showQueue(q)
            Case 4 '離開程式
                Exit While
        End Select
    End While
End Sub
```

Ch11. 資料結構 – 佇列(Queue)

▶ 參考程式：

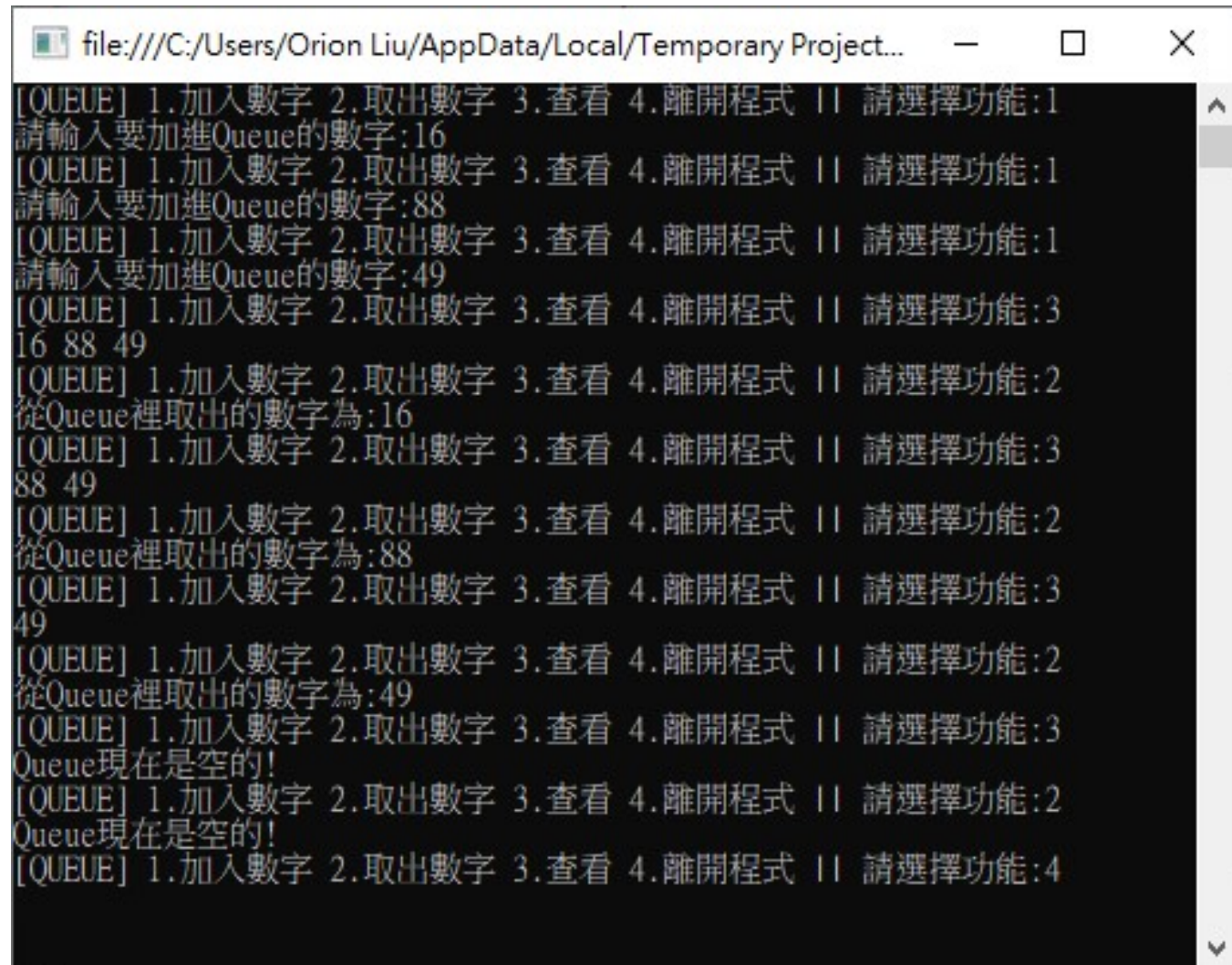
```
Sub addNumberToQueue(q) '加入一個資料到佇列
    Dim n As Integer
    Console.Write("請輸入要加進Queue的數字:")
    n = Console.ReadLine
    q.Enqueue(n)
End Sub

Sub popNumberFromQueue(q) '從佇列取出一個資料
    Dim y As Integer
    If q.count > 0 Then
        y = q.Dequeue()
        Console.WriteLine("從Queue裡取出的數字為:" & y)
    Else
        Console.WriteLine("Queue現在是空的!")
    End If
End Sub

Sub showQueue(q) '印出佇列內容
    If q.count > 0 Then
        For Each element As Integer In q
            Console.Write(element & " ")
        Next
        Console.WriteLine()
    Else
        Console.WriteLine("Queue現在是空的!")
    End If
End Sub
```

Ch11.資料結構 – 佇列(Queue)

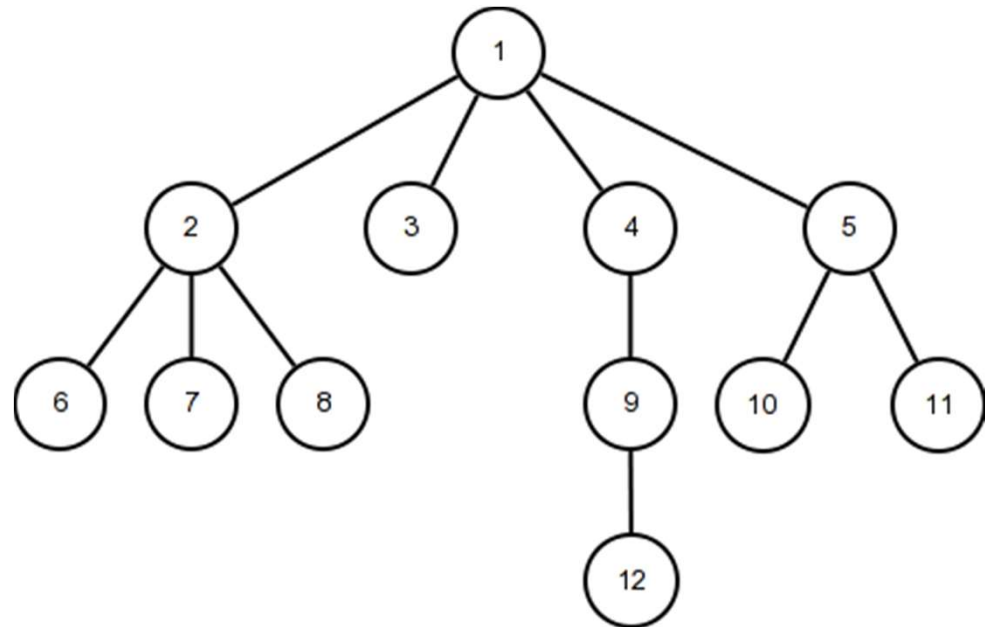
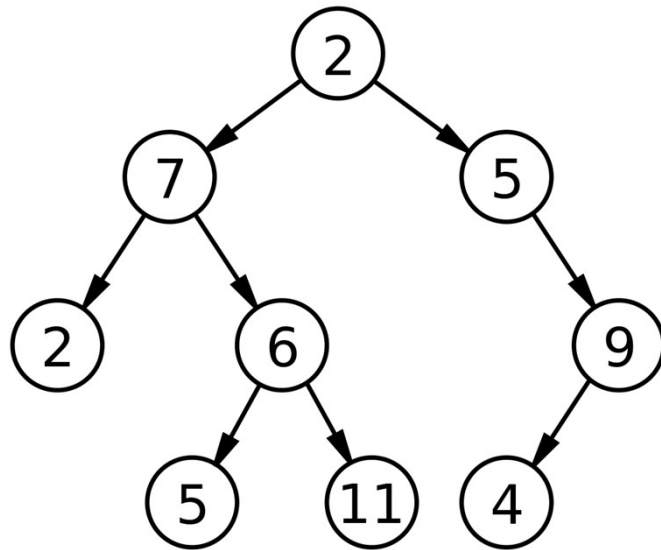
▶ 執行結果：



```
file:///C:/Users/Orion Liu/AppData/Local/Temporary Project...  
[QUEUE] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:1  
請輸入要加進Queue的數字:16  
[QUEUE] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:1  
請輸入要加進Queue的數字:88  
[QUEUE] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:1  
請輸入要加進Queue的數字:49  
[QUEUE] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:3  
16 88 49  
[QUEUE] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:2  
從Queue裡取出的數字為:16  
[QUEUE] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:3  
88 49  
[QUEUE] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:2  
從Queue裡取出的數字為:88  
[QUEUE] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:3  
49  
[QUEUE] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:2  
從Queue裡取出的數字為:49  
[QUEUE] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:3  
Queue現在是空的!  
[QUEUE] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:2  
Queue現在是空的!  
[QUEUE] 1.加入數字 2.取出數字 3.查看 4.離開程式 || 請選擇功能:4  
Queue現在是空的!
```

Ch11. 資料結構 – 二元搜尋樹(Binary Search Tree)

- ▶ 樹是一種資料結構方式，使用連結串列(Link List)來組成資料，因為連接的方式像一棵樹，故稱之。
- ▶ 這都是「樹」：



Ch11. 資料結構 – 二元搜尋樹(Binary Search Tree)

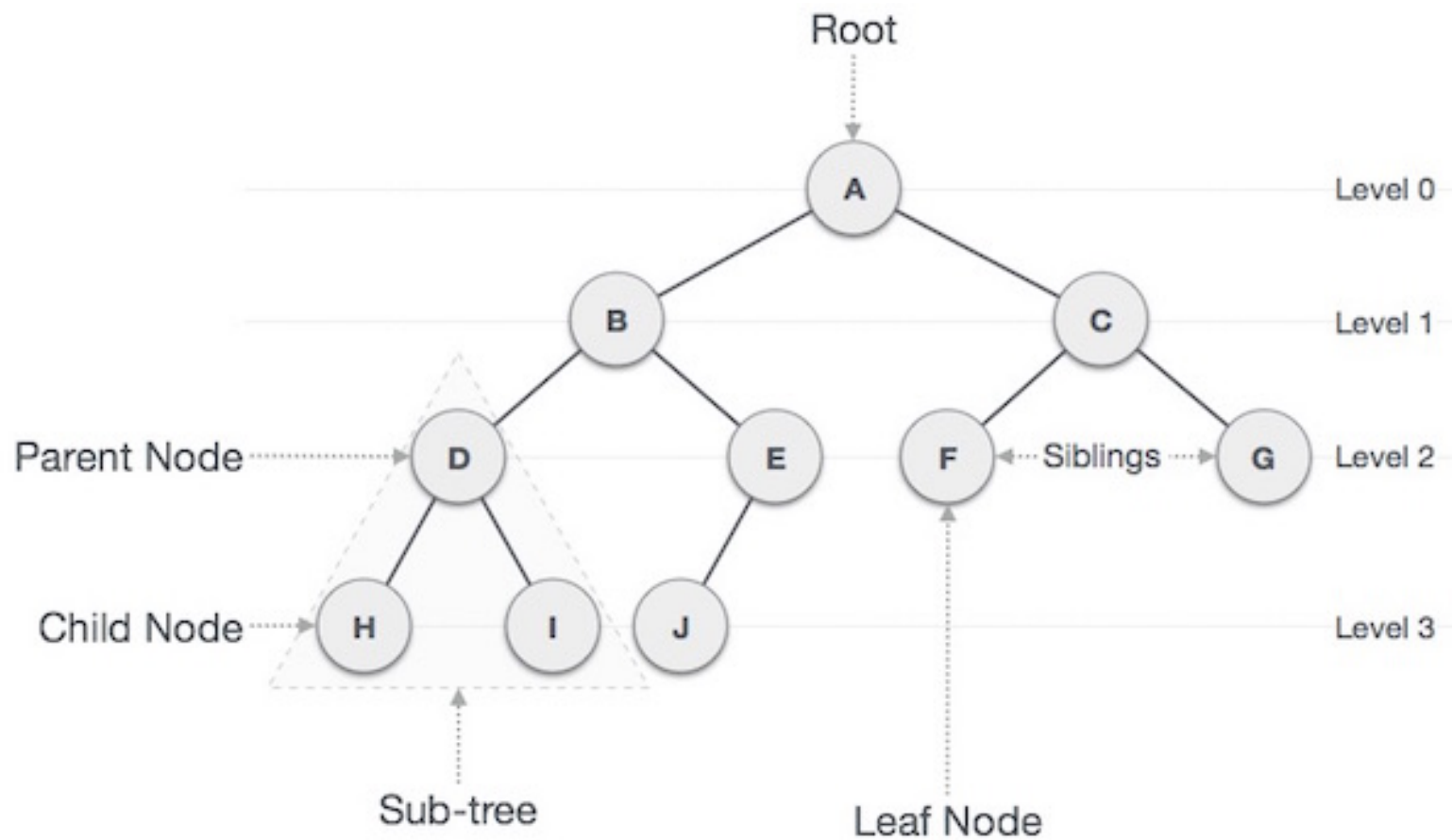
▶ 樹的相關定義：

- ▶ 1. 一個樹包含了一組有限的元素，稱為節點(Node)。
- ▶ 2. 一組有限的方向線段，稱分支(Branch)，連接到結點上。
- ▶ 3. 和這個節點有關的分支數目稱為此節點的degree。
- ▶ 4. 當一個分支指向一個節點時，稱此分支為indegree。
- ▶ 5. 當一個分支離開一個節點時，稱此分支為outdegree。
- ▶ 6. indegree和outdegree的總和等於該節點的degree。
- ▶ 7. 第一個節點稱為根(Root)，除了根節點外，任何節點都必須至少有一個indegree，但outdegree則沒有限制。



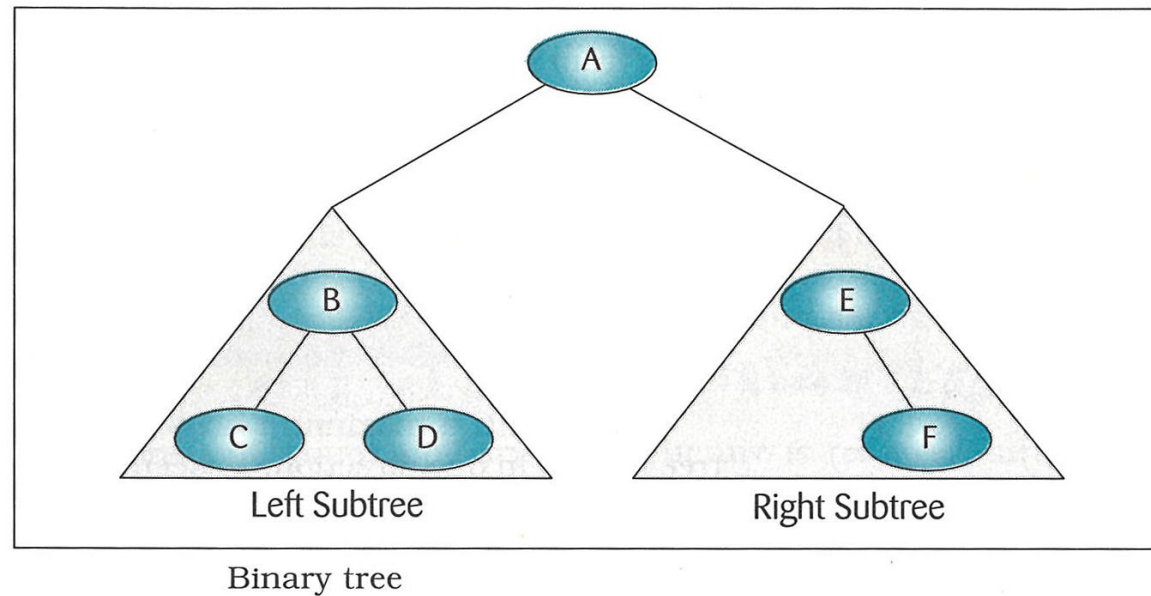
Ch11. 資料結構 – 二元搜尋樹(Binary Search Tree)

► 樹的一些專有名詞：



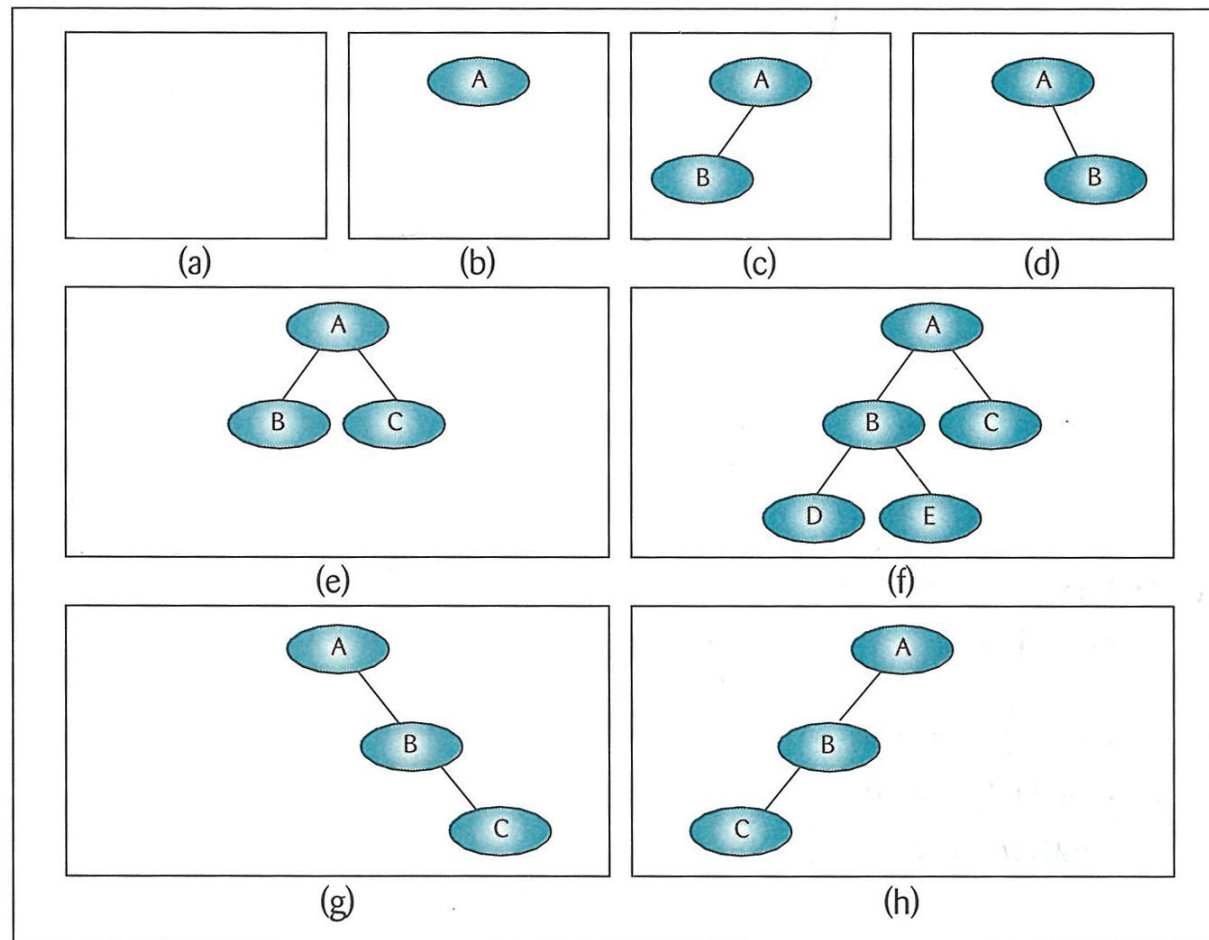
Ch11. 資料結構 – 二元搜尋樹(Binary Search Tree)

- ▶ 二元樹定義：
- ▶ 1. 所有的節點(node)至多只能有兩個子樹，稱二元樹。
- ▶ 2. 左邊稱為左子樹，右邊稱為右子樹。
- ▶ 3. 樹不一定要對稱。



Ch11. 資料結構 – 二元搜尋樹(Binary Search Tree)

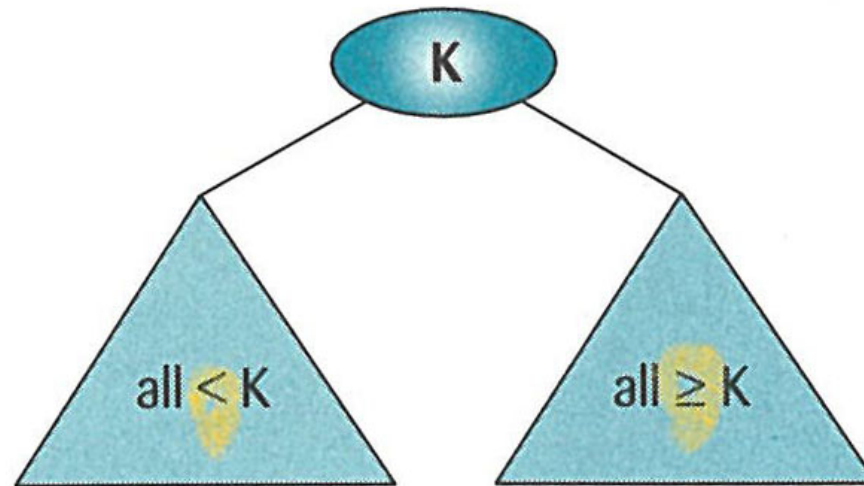
► 一些二元樹的例子：



A collection of binary trees

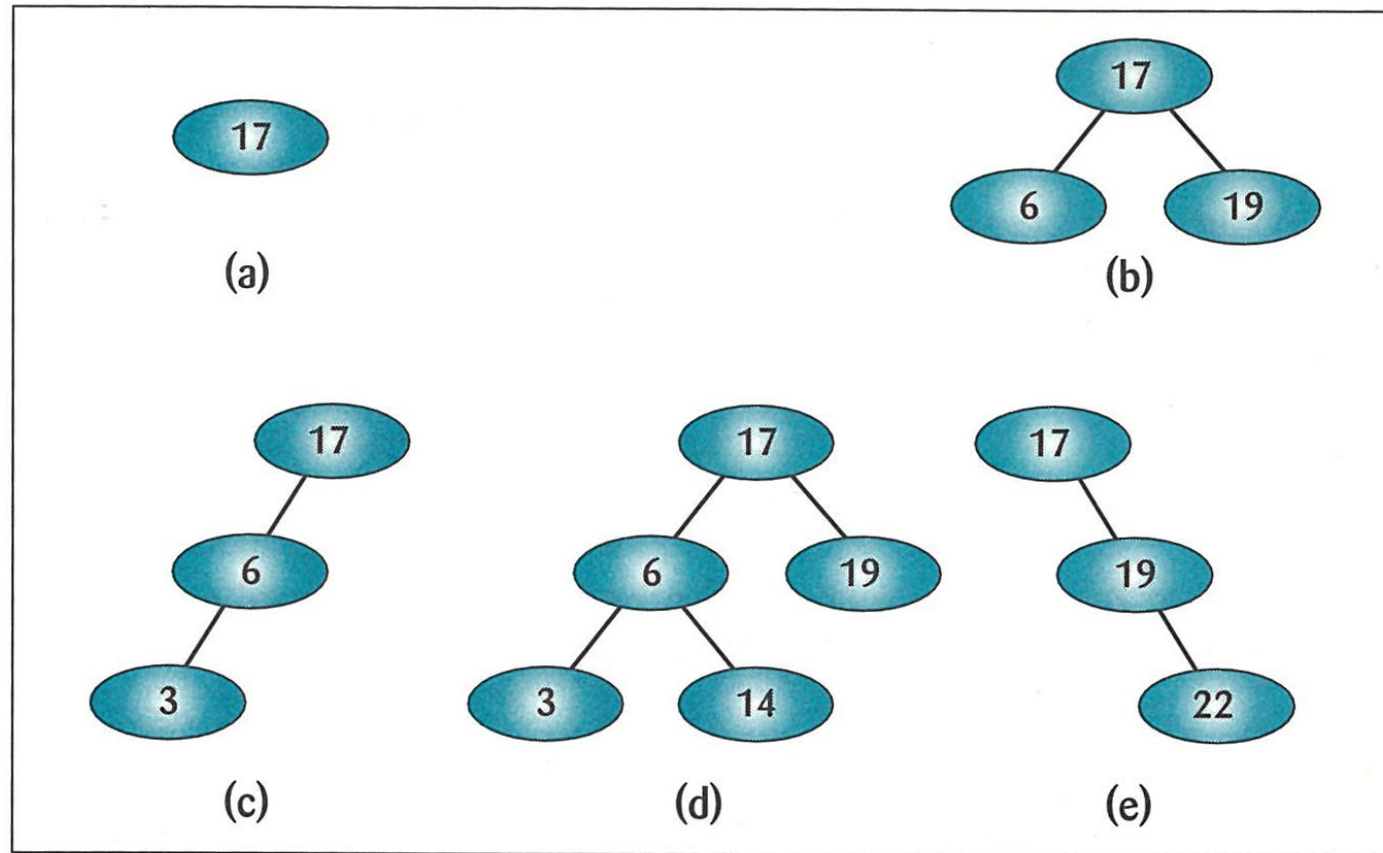
Ch11. 資料結構 – 二元搜尋樹(Binary Search Tree)

- ▶ 二元搜尋樹(Binary Search Tree, BST)是一個二元樹，並具有下列性質：
 - ▶ 1. 任一左子樹都比它的根來的小。
 - ▶ 2. 任一右子樹都大於或等於它的根。
 - ▶ 3. 每一個子樹亦是一個二元搜尋樹。



Ch11. 資料結構 – 二元搜尋樹(Binary Search Tree)

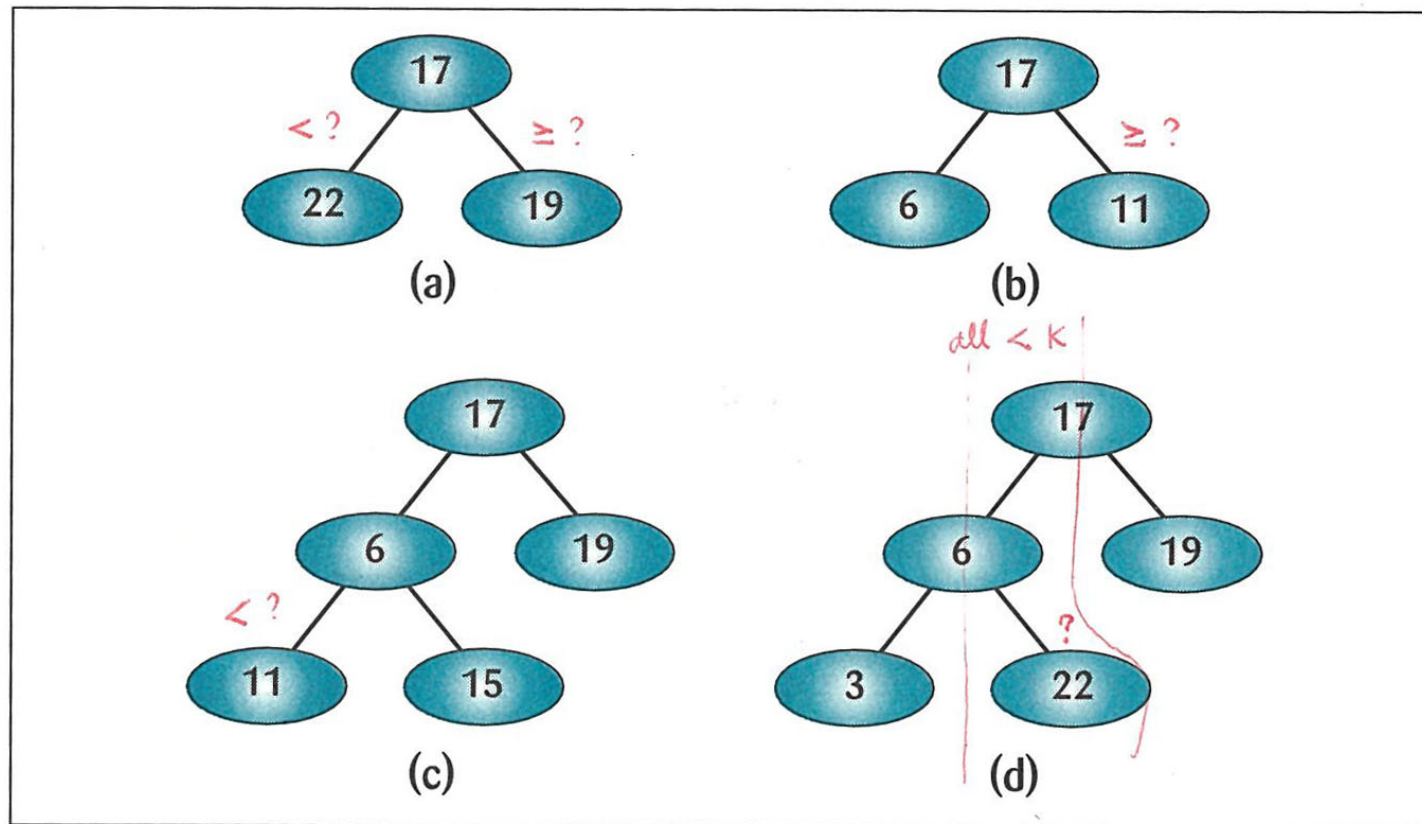
► 一些二元搜尋樹的例子：



Binary search trees

Ch11. 資料結構 – 二元搜尋樹(Binary Search Tree)

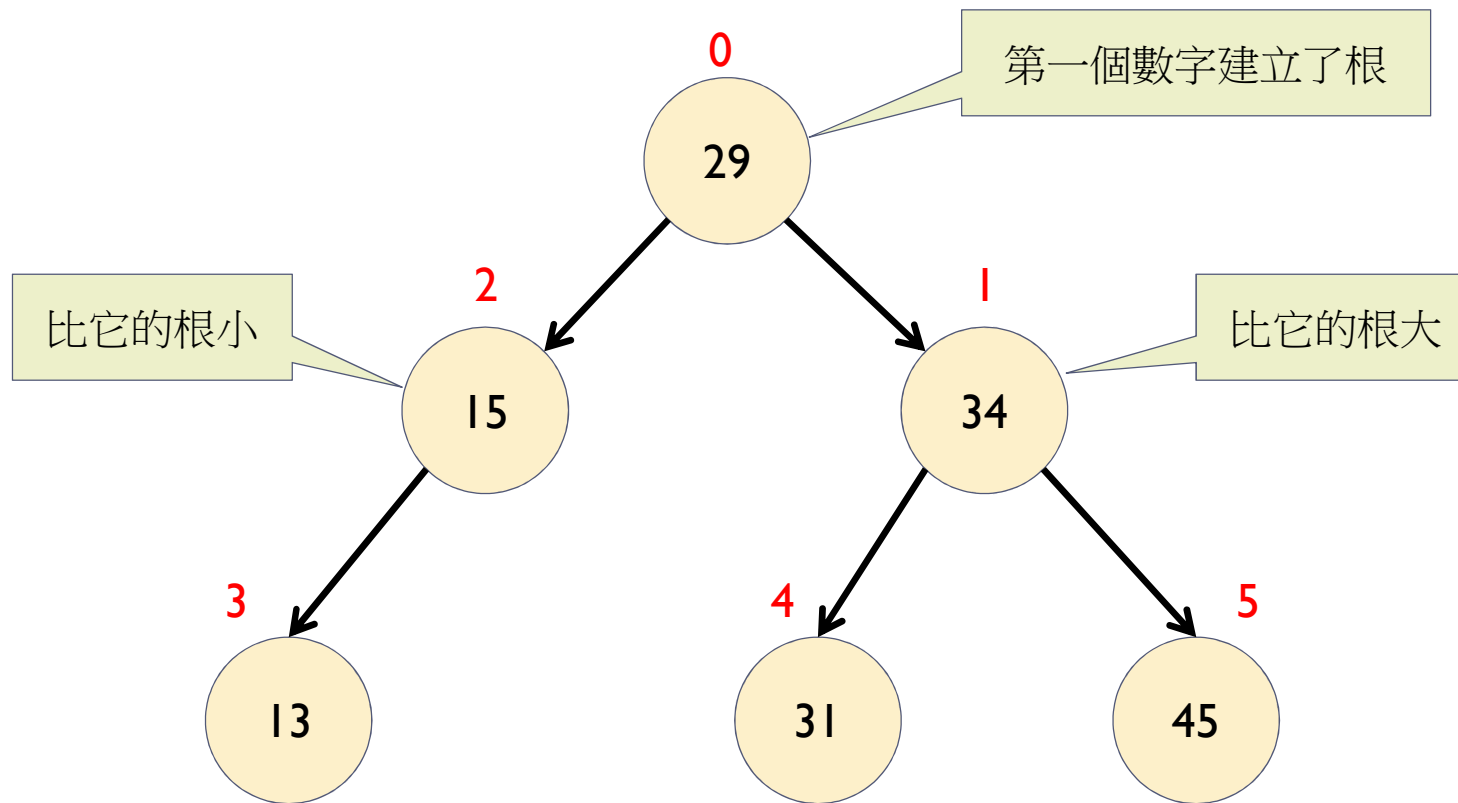
- ▶ 一些無效的二元搜尋樹的例子：



Invalid binary search trees

Ch11. 資料結構 – 二元搜尋樹(Binary Search Tree)

- ▶ 如何建立一個二元搜尋樹
- ▶ 假設輸入數字順序如下：29、34、15、13、31、45



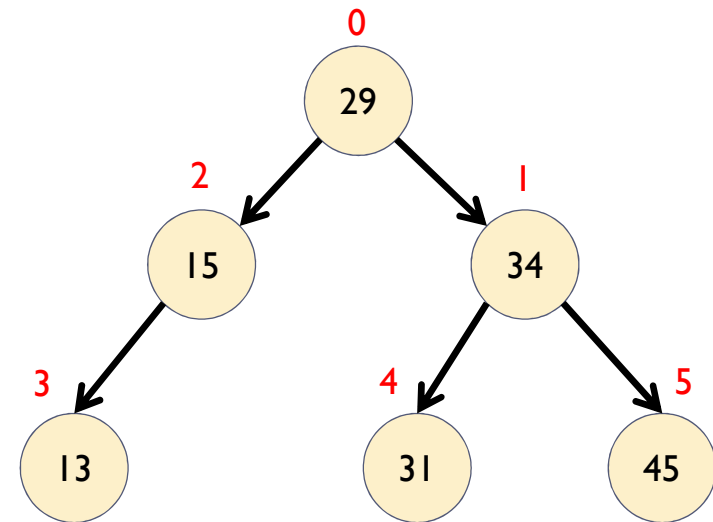
Ch11. 資料結構 – 二元搜尋樹(Binary Search Tree)

- ▶ 在這裡我們使用結構來表示一個節點：

```
Structure node
  Dim value As Integer '節點的值
  Dim num As Integer   '節點的編號
  Dim L As Integer     '節點的左子節點編號
  Dim R As Integer     '節點的右子節點編號
End Structure
```

- ▶ 所以一棵樹可用一個結構陣列來表示：

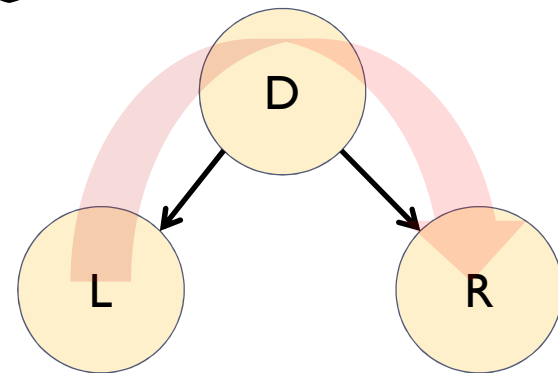
Tree()	value	num	L	R
T(0)	29	0	2	1
T(1)	34	1	4	5
T(2)	15	2	3	-1
T(3)	13	3	-1	-1
T(4)	31	4	-1	-1
T(5)	45	5	-1	-1



- ▶ -1表示無子節點。

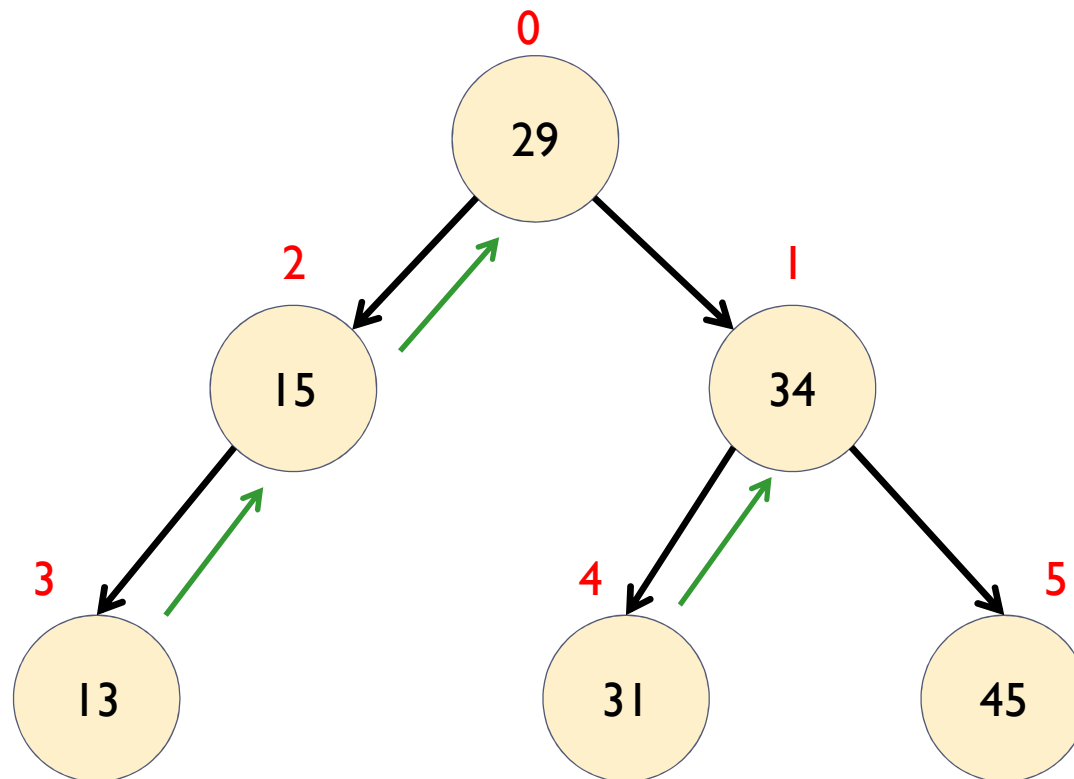
Ch11. 資料結構 – 二元搜尋樹(Binary Search Tree)

- ▶ 二元搜尋樹走訪(列出所有資料)，有三種方式：
 - ▶ DLR: 前序走訪 (Preorder Traversal)
 - ▶ **LDR: 中序走訪(Inorder Traversal)**
 - ▶ LRD: 後序走訪(Postorder Traversal)
 - ▶ (D: root, L: 左子樹, R: 右子樹)
- ▶ 以LDR為例，是先走訪左子樹(直到沒有左子樹為止)，根結點，然後右子樹。如果右子樹底下還有左子樹，擇期左子樹要優先訪，這是一個遞迴。



Ch11. 資料結構 – 二元搜尋樹(Binary Search Tree)

► LDR 中序走訪 (Inorder Traversal) :



輸出結果： 13 15 29 31 34 45



Ch11. 資料結構 – 二元搜尋樹(Binary Search Tree)

▶ 範例程式，結構宣告：

```
Structure node
```

```
    Dim value As Integer '節點的值  
    Dim num As Integer   '節點的編號  
    Dim L As Integer     '節點的左子節點編號  
    Dim R As Integer     '節點的右子節點編號
```

```
End Structure
```

```
Dim T(0) As node ' "節點"的陣列，一棵"樹"
```

Ch11. 資料結構 – 二元搜尋樹(Binary Search Tree)

▶ 主程式：

```
Sub Main() '主程式
    Dim x, y As Integer
    T(0).num = -1 '先指定成-1，表示尚未建立root
    While True
        Console.WriteLine("[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:")
        x = Val(Console.ReadLine)
        Select Case x
            Case 1 '加入一個節點到樹
                Console.WriteLine("請輸入要加入的數字:")
                y = Console.ReadLine
                If T(0).num = -1 Then '空樹，此為第一個節點root，將此node加入binary tree T中
                    T(0).value = y
                    T(0).num = 0
                    T(0).L = -1 '用-1代表左無子樹
                    T(0).R = -1 '用-1代表右無子樹
                Else
                    insert(0, y) '試著找到適合的地方，將輸入的數字插入
                End If
            Case 2 '從樹中搜尋一個資料
                Console.WriteLine("請輸入要搜尋的數字:")
                y = Console.ReadLine
                Console.WriteLine("從NODE(0)開始尋找...")
                searchValueFromTree(0, y)
            Case 3 '中序查看Tree
                inorderShowTree(0)
            Case 4 '離開程式
                Exit While
        End Select
    End While
End Sub
```

Ch11. 資料結構 – 二元搜尋樹(Binary Search Tree)

- ▶ 二元搜尋樹的輸入：將一個數字input_value輸入到某一節點index。

```
Sub insert(index, input_value)    '加入一個節點到樹
    Dim U As Integer
    '若輸入的數字比此node所儲存的值還小，則針對left chile tree
    If input_value < T(index).value Then
        '若left chile tree存在，試著在left sub-tree中找到適合的位置插入輸入的數字
        If T(index).L > -1 Then
            insert(T(index).L, input_value)
        Else '若left chile tree不存在，將此node產生一個新的left chile node並儲存此輸入的數字
            U = UBound(T) + 1
            ReDim Preserve T(U)    '產生一個新的node
            T(U).value = input_value : T(U).num = U
            T(U).L = -1 : T(U).R = -1
            T(index).L = U
            Console.WriteLine("加入新node:NODE(" & U & ")(此node為NODE(" &
                               T(index).num & ")的left chile node).")
        End If
    Else '若輸入的數字比此node所儲存的值還大，則針對right chile tree
        If T(index).R > -1 Then
            insert(T(index).R, input_value)
        Else '若right chile tree不存在，將此node產生一個新的right chile node並儲存此輸入的數字
            U = UBound(T) + 1
            ReDim Preserve T(U)    '產生一個新的node
            T(U).value = input_value : T(U).num = U
            T(U).L = -1 : T(U).R = -1
            T(index).R = U
            Console.WriteLine("加入新node:NODE(" & U & ")(此node為NODE(" &
                               T(index).num & ")的right chile node).")
        End If
    End If
End Sub
```

Ch11. 資料結構 – 二元搜尋樹(Binary Search Tree)

▶ 二元搜尋樹的搜尋：

```
Sub searchValueFromTree(index, input_value) '從樹中搜尋一個資料
    Console.WriteLine("NODE(" & T(index).num & ")所儲存的數字為" & T(index).value)
    If input_value < T(index).value Then '若比此node所儲存的數字還小，則找left sub-tree
        Console.WriteLine("尋找NODE(" & T(index).num & ")的left child node:NODE(" &
            T(index).L & ")")
        If T(index).L = -1 Then
            Console.WriteLine("NODE(" & T(index).num & ")的left child node不存在，找不到" &
                input_value & "這個數字!")
        Else
            searchValueFromTree(T(index).L, input_value)
        End If
    ElseIf input_value > T(index).value Then '若比此node所儲存的數字還大，則找right sub-tree
        Console.WriteLine("尋找NODE(" & T(index).num & ")的right child node:NODE(" &
            T(index).R & ")")
        If T(index).R = -1 Then
            Console.WriteLine("NODE(" & T(index).num & ")的right child node不存在，找不到" &
                input_value & "這個數字!")
        Else
            searchValueFromTree(T(index).R, input_value)
        End If
    Else '此node所儲存的數字與input_value相同
        Console.WriteLine(input_value & "這個數字存在tree中(位置在NODE(" & T(index).num & "))!")
    End If
End Sub
```



Ch11. 資料結構 – 二元搜尋樹(Binary Search Tree)

- ▶ 二元搜尋樹的讀取資料：

```
Sub inorderShowTree(index) '中序查看Tree
    If T(index).L > -1 Then '追左樹
        inorderShowTree(T(index).L)
    End If
    Console.WriteLine("NODE(" & T(index).num & ")所儲存的數字為" &
        T(index).value & ",left chile node為NODE(" &
        T(index).L & "),right chile node為NODE(" &
        T(index).R & ")") '中節點
    If T(index).R > -1 Then '追右樹
        inorderShowTree(T(index).R)
    End If
End Sub
```

- ▶ 要讀出整棵樹裡面各節點的數字，我們用採中序式走訪。



Ch11. 資料結構

▶ 執行結果：

```
file:///C:/Users/Orion Liu/documents/visual studio 2010/Projects/ConsoleApplication...
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:1
請輸入要加入的數字:16
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:1
請輸入要加入的數字:88
加入新node:NODE(1)(此node為NODE(0)的right chile node).
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:1
請輸入要加入的數字:49
加入新node:NODE(2)(此node為NODE(1)的left chile node).
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:1
請輸入要加入的數字:8
加入新node:NODE(3)(此node為NODE(0)的left chile node).
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:1
請輸入要加入的數字:65
加入新node:NODE(4)(此node為NODE(2)的right chile node).
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:1
請輸入要加入的數字:99
加入新node:NODE(5)(此node為NODE(1)的right chile node).
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:1
請輸入要加入的數字:4
加入新node:NODE(6)(此node為NODE(3)的left chile node).
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:2
請輸入要搜尋的數字:10
從NODE(0)開始尋找...
NODE(0)所儲存的數字為16
尋找NODE(0)的left child node:NODE(3)
NODE(3)所儲存的數字為8
尋找NODE(3)的right child node:NODE(-1)
NODE(3)的right child node不存在，找不到10這個數字!
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:2
請輸入要搜尋的數字:8
從NODE(0)開始尋找...
NODE(0)所儲存的數字為16
尋找NODE(0)的left child node:NODE(3)
NODE(3)所儲存的數字為8
8這個數字存在tree中(位置在NODE(3))!
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:3
NODE(6)所儲存的數字為4,left chile node為NODE(-1),right chile node為NODE(-1)
NODE(3)所儲存的數字為8,left chile node為NODE(6),right chile node為NODE(-1)
NODE(0)所儲存的數字為16,left chile node為NODE(3),right chile node為NODE(1)
NODE(2)所儲存的數字為49,left chile node為NODE(-1),right chile node為NODE(4)
NODE(4)所儲存的數字為65,left chile node為NODE(-1),right chile node為NODE(-1)
NODE(1)所儲存的數字為88,left chile node為NODE(2),right chile node為NODE(5)
NODE(5)所儲存的數字為99,left chile node為NODE(-1),right chile node為NODE(-1)
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:4
```


下課~

