

บทที่ 8

โครงสร้างข้อมูลแบบอะเรย์หลายมิติ (Multi Dimension Array)

บทนำ จากความรู้ในบทที่ 7 นั้นเราได้ทราบถึงโครงสร้างข้อมูลแบบซัซซัน โดยการใช้อะเรย์ประเภทมิติเดียวมาแล้ว ถึงการจัดการพื้นที่ และการนำไปดำเนินการ ซึ่งจะเห็นประโยชน์ในการนำไปประยุกต์ใช้งานที่ชัดเจน นอกจากโครงสร้างอะเรย์ประเภทมิติเดียวแล้ว ภาษาคอมพิวเตอร์เกือบทุกภาษาก็สามารถนิยามการสร้างอะเรย์ที่มีโครงสร้างเป็นแบบหลายมิติ เพื่อนำมาดำเนินการได้ในสถานการณ์ที่มีการจัดเก็บข้อมูลที่ซัซซันยิ่งขึ้นในสมองคอมพิวเตอร์ ในทางปฏิบัติแล้วการที่เราจะนิยามว่ามีข้อมูลประเภทอะเรย์มิติเดียวในการใช้งานก็คงจะพอในการใช้งาน โดยการกำหนดรูปแบบอะเรย์ 2 มิติว่าเป็นอะเรย์ซัซซันอะเรย์ และในทำนองเดียวกัน ถ้าเป็นอะเรย์ 3 มิติ ก็ให้เป็นอะเรย์ซัซซันอะเรย์ซัซซันอะเรย์ ดังภาพที่ปรากฏนี้

$$X_{4 \times 4} = \begin{pmatrix} 1 & 3 & 5 & 7 \\ 2 & 0 & 5 & 1 \\ 8 & 4 & 4 & 9 \\ 1 & 0 & 2 & 4 \end{pmatrix}$$

ดู X ในรูปของ Column Vector

$$A = \begin{pmatrix} 1 \\ 2 \\ 8 \\ 1 \end{pmatrix} \quad B = \begin{pmatrix} 3 \\ 0 \\ 4 \\ 0 \end{pmatrix} \quad C = \begin{pmatrix} 5 \\ 5 \\ 4 \\ 2 \end{pmatrix} \quad D = \begin{pmatrix} 7 \\ 1 \\ 9 \\ 4 \end{pmatrix}$$

ดู X ในรูปของ Row Vector

$$A = (1 \ 3 \ 5 \ 7) \quad B = (2 \ 0 \ 5 \ 1) \\ c = (8 \ 4 \ 4 \ 9) \quad D = (1 \ 0 \ 2 \ 4)$$

$$\text{ดังนั้น } x = (A \ B \ c \ D)$$

การมองอะเรย์ X ที่เป็น 2 มิติในรูปของอะเรย์มิติเดียว ก็คือ เป็นรูปแบบของอะเรย์ซ้อนอะเรย์ ดังนั้น เมตริก X จะประกอบด้วย Column Vector จำนวน 4 Vector คือ Vector A, B, C และ D ดังรูป หรืออาจจะมองในรูปของ Row Vector ดังนั้น เมตริก X จะประกอบด้วย Row Vector จำนวน 4 Vector คือ Vector A, B, C และ D ก็ได้ ถึงแม้ว่ารูปแบบในการจัดการข้อมูล 2 มิติกลายเป็นมิติเดียวได้ก็จริง แต่ก็จะประสบปัญหาในทางปฏิบัติว่าการอ้างถึงอะเรย์แต่ละอะเรย์ย่อยนั้นจะไม่ค่อยสัมพันธ์กัน ทำให้ผู้ดำเนินการยุ่งยากในการนำไปประยุกต์ใช้งาน นอกจากนี้ในกรณีที่เป็นอะเรย์ 3 มิติหรือมากกว่า จะยิ่งประสบปัญหายุ่งยากมากขึ้นเป็นลำดับ ดังนั้นถ้าหากว่าภาษาคอมพิวเตอร์ที่ใช้งานนั้นสามารถกำหนดเป็นอะเรย์เป็นหลายมิติได้ก็ควรจะกำหนดให้เป็นหลายมิติด้วยเพื่อความสะดวกในการจัดการกับของข้อมูล ภายในอะเรย์นั้น

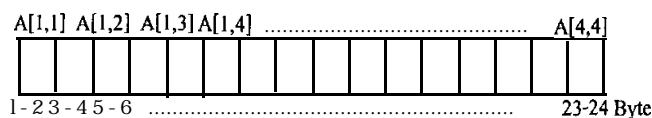
ตัวอย่างการประกาศเพื่อจัดสรรพื้นที่ให้กับอะเรย์ ชนิด 2 มิติเพื่อเก็บข้อมูลที่เป็นเลขจำนวนเต็ม โดยใช้ภาษาปาสคาล

```
const   row = 4 ;
        col = 4 ;

type
        aset = array [1..row,1..col] of integer ;

var
        A : aset ;
```

รูปแบบการจัดสรรพื้นที่ในลักษณะทางกายภาพ (Physical Address)



การใช้รูปแบบการจัดสรรพื้นที่โดยใช้รูปแบบของ Physical Address เหมาะสำหรับการทำความเข้าใจในการจัดเก็บเพื่อใช้ติดต่อในการส่งข้อมูลกับโปรแกรมย่อย แต่สำหรับการใช้งานในลักษณะของการนำไปประมวลผลข้อมูลอาจจะดูค่อนข้างเข้าใจยาก ดังนั้นเราอาจจะใช้รูปแบบที่เป็น Logical View จะดีกว่า การนำเสนอสำหรับ อะเรย์ A ประเภท 2 มิติจะปรากฏเป็นเมตริก ในเชิงคณิตศาสตร์ดังนี้

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

เมตริก นับเป็นโครงสร้างที่มีบทบาทและการใช้งานมากมาย และโดยที่การจะดำเนินงานเกี่ยวกับเมตริก จะจัดในโครงสร้างที่เป็นอะเรย์ 2 มิติ ดังนั้นการศึกษาเรื่องการจัดการข้อมูลที่มีลักษณะเป็นเมตริก จึงเป็นเรื่องที่เราจำเป็นต้องให้ความสนใจ เพื่อจะนำไปใช้งานในเชิงของ Programming Method

เมตริกประเภทพิเศษต่างๆ โดยปกติเรามักจะพบเมตริก แบบปกติธรรมดาที่ใช้จัดเก็บข้อมูล ดังนี้

$$A = \begin{pmatrix} 1 & 3 & 5 & 7 \\ 2 & 0 & 5 & 1 \\ 8 & 4 & 4 & 9 \\ 1 & 0 & 2 & 4 \end{pmatrix}$$

ซึ่งสามารถไปประยุกต์ได้หลายๆสถานการณ์ดังนี้

- ใช้แถวของเมตริก A เพื่อบ่งชี้ว่า เป็นที่ตั้งของคลังสินค้า ดังนี้คือ แถวที่ 1 หมายถึงคลังสินค้าที่ ภาคเหนือ แถวที่ 2 หมายถึงคลังสินค้าที่ภาคกลางเป็นต้น ส่วนสดมภ์ก็อาจจะหมายถึงประเภทของสินค้า เช่นสดมภ์ที่ 1 จะหมายถึงสินค้าประเภทข้าวโพด ส่วนสดมภ์ที่ 2 จะหมายถึงสินค้าประเภทละหู่ เป็นต้น
- ใช้แถวของเมตริก A เพื่อบ่งชี้ว่า เป็นที่ตั้งของตึกเรียน ดังนี้คือ แถวที่ 1 หมายถึงตึกเรียนชั้นที่ 1 แถวที่ 2 หมายถึงตึกเรียนชั้นที่ 2 เป็นต้น โดยที่สดมภ์ จะหมายถึงห้องเรียนย่อยของชั้นเรียนแต่ละ ตัวอย่าง เช่น BSS[2,3] จะหมายถึงห้องเรียนที่ตึก BSS เป็นชั้นที่ 2 และห้องที่ 3 เป็นต้น
- ใช้แถวของเมตริก A เป็นตัวบ่งชี้ชื่อของกางเกงยีน ในขณะที่สดมภ์จะหมายถึงเบอร์ของกางเกงยีน (เช่นเบอร์ XL , L , M เป็นต้น)
- ใช้แถวของเมตริก A เป็นตัวบ่งชี้ถึงคนงานในบริษัทว่าคนที่ 1,2,... ในขณะที่สดมภ์จะหมายถึงจำนวนชั่วโมงที่ทำงานในแต่ละวัน เช่นสดมภ์ที่ 1 เป็นจำนวนชั่วโมงที่ทำงานในวันจันทร์ ส่วนสดมภ์ที่ 2 เป็นจำนวนชั่วโมงที่ทำงานในวันอังคาร เป็นต้น

- ใช้แถวของเมตริก A เป็นตัวบ่งชี้ถึงสินค้าที่เข้ามาแข่งขันประกวดราคา และแต่ละสคัมภ์จะหมายถึงคุณสมบัติของสินค้าที่จะมีลักษณะต่างๆกัน ดูจากตัวอย่างต่อไปนี้

สินค้าที่เข้าประมูล	ราคารวมภาษี	ขนาดเครื่องยนต์	จำนวนหมุน(รอบ/นาที)	อัตราค/นาที	กำลังส่ง
1	50000
2	65000		
3	65320			
4	58401			
5	73210

- การใช้งานในบางลักษณะอาจจะต้องกำหนดอะเรย์ให้มีมิติมากกว่า 2 มิติได้ เช่นการจัดเก็บข้อมูลเกี่ยวกับการขาย โดยแยกเป็นรายปี พื้นที่ตามภาค และประเภทสินค้า ซึ่งแสดงด้วยโครงสร้างเมตริกดังนี้

	ยอดขายปีพศ.2535	ยอดขายปีพศ.2536
Type of Product			
Sale Area	$\begin{pmatrix} 1 & 6 & 10 & 20 \\ 2 & 7 & 11 & 8 \\ 3 & 8 & 12 & 14 \\ 4 & 9 & 13 & 15 \end{pmatrix}$	$\begin{pmatrix} 5 & 6 & 10 & 20 \\ 8 & 7 & 11 & 8 \\ 1 & 8 & 12 & 14 \\ 3 & 8 & 12 & 14 \\ 4 & 9 & 13 & 15 \end{pmatrix}$	

วิธีการจัดเก็บสามารถจัดเก็บในโครงสร้างของอะเรย์ 3 มิติคือมิติแรกแทน ปี พ.ศ. มิติ ที่ 2 แทน พื้นที่ของแต่ละภาคในประเทศไทย และมิติที่ 3 แทน สินค้าแต่ละประเภทที่ขาย

ตัวอย่างการใช้ภาษา Pascal กำหนดอะเรย์หลายมิติ

8 x 4 x 3 = 96 Element Element Type

Type \longleftrightarrow \longleftrightarrow

RealArray = array [1..8,1..4,1..3] of real;

Var

 A : RealArray;

ตัวอย่างการกำหนดอะเรย์ 2 มิติที่เป็นอักขระ

var

Name : array[1..8,1..8] of char

ผลที่ได้ก็คือพื้นที่ที่ได้รับการจัดสรรเพื่อนำไปจัดเก็บข้อมูลที่เป็นอักขระดังรูปที่ปรากฏนี้

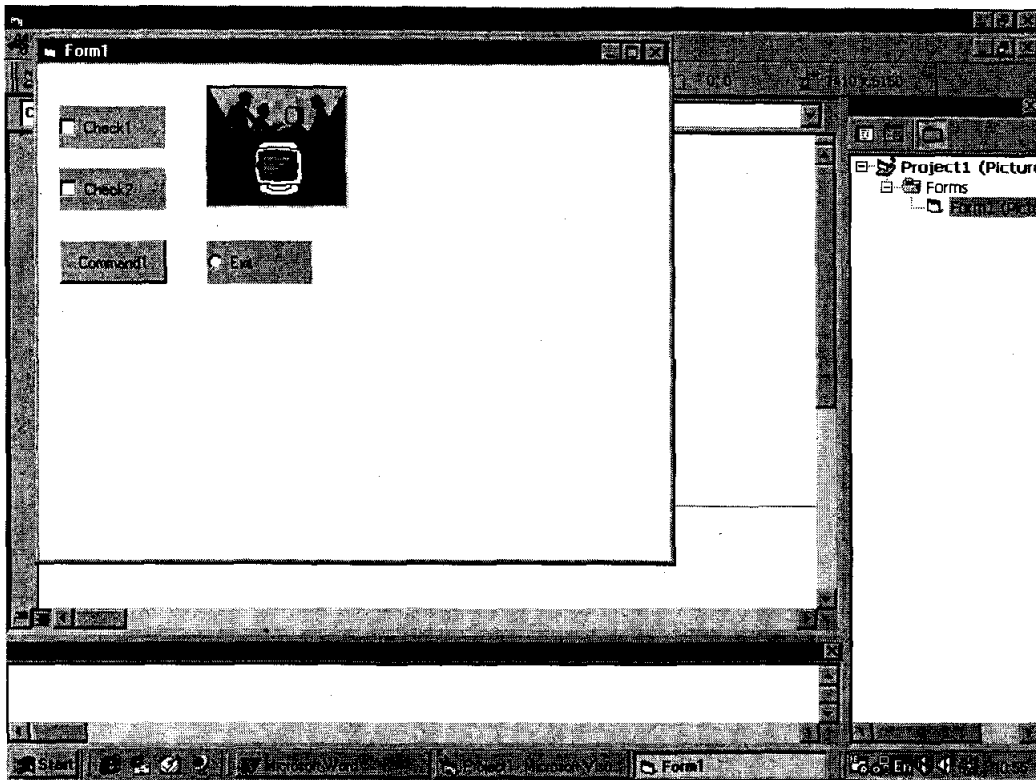
	A[1,1]					A[1,8]		
A[1]	T	H	A	I	L	A	N	D
A[2]								
A[3]								
A[4]								
A[5]								
A[6]								
A[7]								
A[8]								

- การระบุตำแหน่งในการเก็บข้อมูลนั้นจะต้องระบุด้วยการใช้ดัชนี 2 ตัวเพื่อระบุตำแหน่งคือ A[I,J] ; I = 1,2,...,8 ; J = 1,2,...,8 โดยที่เซลล์ที่ระบุนั้นสามารถเก็บอักขระได้เพียง 1 อักขระเพราะมีพื้นที่เพียง 1Byte เท่านั้น

ข้อสังเกตของการสร้างอะเรย์ที่มีเกิน 3 มิติจะไม่ค่อยนิยมใช้กัน เพราะค่อนข้างจะจินตนาการในการจัดการกับข้อมูลลำบาก รวมถึงการอ้างถึงตำแหน่งในการนำไปใช้งานก็จะค่อนข้างยุ่งยากด้วย

ความสามารถในการจัดเก็บข้อมูลของอะเรย์ โดยทั่วไปนั้นการใช้อะเรย์ไม่ว่าจะเป็นมิติเดียวหรือหลายมิติ มักจะใช้กับข้อมูลที่เป็นลักษณะ ของ Numeric Value หรือไม่ก็เป็น Text (Character or String) เท่านั้น แต่ปัจจุบันเมื่อมีระบบหลายสื่อเข้ามาใช้งานในระบบคอมพิวเตอร์ ดังนั้นข้อมูลที่จะจัดเก็บในเซลล์ของอะเรย์ ก็อาจจะเป็น Meta File ได้ เช่น รูป เสียง เป็นต้น หรืออาจจะมีการปรับปรุงให้ไม่ต้องเก็บ Meta File เหล่านั้นในเซลล์ของอะเรย์เพราะจะทำให้เสียพื้นที่มาก เราอาจจะเก็บข้อมูลในเซลล์เป็น ชื่อของ Meta File เหล่านั้นที่จะ Link ไปได้ ซึ่งเป็นแนวปฏิบัติที่เราานิยมจะใช้ในการดำเนินงานกัน

ตัวอย่างต่อไปนี้จะเป็นการใช้อะเรย์เพื่อเก็บตำแหน่งเพิ่มรูปที่จะติดต่อไปเพื่อนำมาใช้งานในการแสดงภาพที่ผู้ใช้ต้องการจะเลือก



ภาพที่ปรากฏนี้เป็นส่วนของ User Interface ที่ใช้ Visual Basic ในการสร้างเพื่อนำมาใช้งาน โดยจะมีโปรแกรมที่กำหนดให้ผู้ใช้งานสามารถเลือกรูปที่ต้องการได้ โดยรูปที่เลือกนั้นจะค้นจากข้อมูลในอะเรย์เพื่อนำไป Link กับแฟ้มข้อมูลที่เป็นรูปภาพต่อไปอันจะนำมาแสดงผลรูปที่ตำแหน่งของ Object ที่ต้องการ โปรแกรมที่ปฏิบัติงานนี้มี 2 ส่วนส่วนหนึ่งเกิดจากการ Generate Code จาก Software VB ส่วนที่สองเกิดจากการเขียนโปรแกรมขึ้นมาใช้งาน ส่วนที่เกิดจากการ Generate Code จาก Software VB นั้นจะเป็นการสร้าง User Interface ซึ่งถ้าเราจะเขียนขึ้นมาใช้งานนั้นจะค่อนข้างเสียเวลาดังนั้นจึงมีการพัฒนาทางด้าน Software Technology เพื่อพัฒนาในส่วนการสร้าง User Interface ที่ใช้แนวทางของการ โปรแกรมเชิงวัตถุ (Object Oriented Programming) เข้าช่วย

รูปแบบการเขียนของโปรแกรมส่วนที่ 1 ที่ถูก Generate Code จาก Software VB

```
Type=Exe
Form=Picture.frm
Reference=*\G{00020430-0000-0000-C000-00000000046}#2.0#0#C:\WINDOWS\SYSTEM\STDOLE2.TLB#OLE Automation
IconForm="Form1"
Startup="Form1"
Command32=""
Name="Project1"
```

HelpContextID="0"
CompatibleMode="0"
MajorVer=1
MinorVer=0
RevisionVer=0
AutoIncrementVer=0
ServerSupportFiles=0
VersionCompanyName="dpu"
CompilationType=0
OptimizationType=0
FavorPentiumPro(tm)=0
CodeViewDebugInfo=0
NoAliasing=0
BoundsCheck=0
OverflowCheck=0
FIPointCheck=0
FDIVCheck=0
UnroundedFP=0
StartMode=0
Unattended=0
ThreadPerObject=0
MaxNumberOfThreads=1

VERSION 5.00

Begin VB.Form Form1

Caption = "Form1"

ClientHeight = 3195

ClientLeft = 60

ClientTop = 345

ClientWidth = 4680

LinkTopic = "Form1"

ScaleHeight = 3195

ScaleWidth = 4680

StartPosition = 3 'Windows Default

WindowState = 2 'Maximized

Begin VB.OptionButton Option1

Caption = "Exit"
Height = 495
Left = 1920
Picture = "Picture.frx":0000
TabIndex = 5
Top = 2040
Width = 1215

End

Begin VB.CommandButton Command1

Caption = "Command1"
Height = 495
Left = 240
TabIndex = 4
Top = 2040
Width = 1215

End

Begin VB.CheckBox Check2

Caption = "Check2"
Height = 495
Left = 240
Picture = "Picture.frx":0442
TabIndex = 3
Top = 1200
Width = 1215

End

Begin VB.CheckBox Check1

Caption = "Check1"
Height = 495
Left = 240
TabIndex = 2
Top = 480
Width = 1215

End

Begin VB.PictureBox Picture1


```

AutoSize      = -1 'True
BackColor     = &H8000000D&
Height       = 1425
Left         = 1920
Picture      = "Picture.frx":0884
ScaleHeight  = 1365
ScaleWidth   = 1575
TabIndex    = 0
Top         = 240
Width       = 1635
Begin VB.PictureBox Picture2
    Height      = 15
    Left       = 480
    ScaleHeight = 15
    ScaleWidth  = 135
    TabIndex   = 1
    Top        = 480
    Width      = 135
End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Sub Check1_Click()
    Dim X(1 To 3) As StdPicture
    Dim S As StdPicture
    Set X(1) = LoadPicture("A:Bird.wmf")
    Set X(2) = LoadPicture("A:Dancers.wmf")
    Set X(3) = LoadPicture("A:Heart.wmf")
    n1 = InputBox("Input number of Picture")
    n = Val(n1)
    If Check1.Value = 1 Then

```

```

        Picture1.Visible = True
    Form1.FillColor = QBColor(1)
    Set Picture1.Picture = X(n)
End If
End Sub
Private Sub Check2_Click()
    If Check2.Value = 1 Then
        Picture1.FillColor = RGB(225, 0, 0)
        Picture1.Visible = False
    End If
End Sub
Private Sub Command1_Click()
    Check1.Value = 0
    Check2.Value = 0
End Sub
Private Sub Option1_Click()
    End
End Sub

```

โปรแกรมส่วนที่เราจะเขียนขึ้นมาเองจะปรากฏดังนี้

```

Private Sub Check1_Click()
    Dim X(1 To 3) As StdPicture
    Dim S As StdPicture
    Set X(1) = LoadPicture("A:Bird.wmf")
    Set X(2) = LoadPicture("A:Dancers.wmf")
    Set X(3) = LoadPicture("A:Heart.wmf")
    n1 = InputBox("Input number of Picture")
    n = Val(n1)
    If Check1.Value = 1 Then
        Picture1.Visible = True
        Form1.FillColor = QBColor(1)
        Set Picture1.Picture = X(n)
    End If
End Sub

```

```

End If
End Sub
Private Sub Check2_Click()
    If Check2.Value = 1 Then
        Picture1.FillColor = RGB(225, 0, 0)
        Picture 1 .Visible = False
    End If
End Sub

```

การนำเมตริกไปดำเนินการ งานทางคณิตศาสตร์ที่เกี่ยวข้องกับการใช้เมตริก นั้นมักจะมึงานประมวลผลพื้นฐานประกอบการ บวกเมตริก ลบเมตริก และคูณเมตริก ซึ่งในกรณีที่มีเมตริกซ์ขนาดใหญ่ เราจำเป็นต้องใช้โปรแกรมเข้าช่วยดำเนินการ ดังนั้นในเราจำเป็นต้องทราบถึงขั้นตอนในการประมวลผลงานพื้นฐานของเมตริก

- การบวกเมตริก $A + B$ สองเมตริกจะบวกกันได้ นั้นจะต้องมีขนาดของแถวและสดมภ์เท่ากัน คำสั่งในการบวกกันจะปรากฏดัง Pseudo Program นี้

```

For I = 1 to Row
    For J = 1 to Col
        C[I, J] = A[I, J] + B[I, J]
    End for J
End for I

```

- การลบเมตริก $A - B$ สองเมตริกจะบวกกันได้ นั้นจะต้องมีขนาดของแถวและสดมภ์เท่ากัน คำสั่งในการลบกันจะปรากฏดัง Pseudo Program นี้

```

For I = 1 to Row
    For J = 1 to Col
        C[I, J] = A[I, J] - B[I, J]
    End for J
End for I

```

- การคูณเมตริก A x B สองเมตริกจะคูณกันได้นั้นจะต้องมีขนาดของสครมภ์ของเมตริก A จะต้องเท่ากับแถวของเมตริก B จึงจะสามารถดำเนินการในการคำนวณได้ คำสั่งในการคูณกันจะปรากฏดัง Pseudo Program นี้

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}_{4 \times 4} \quad \bullet \quad B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \\ b_{41} & b_{42} & b_{43} \end{pmatrix}_{4 \times 3}$$

$$C = \begin{pmatrix} a_{11} * b_{11} + a_{12} * b_{21} + a_{13} * b_{31} + a_{14} * b_{41} & \dots & \dots & \dots \\ a_{21} * b_{11} + a_{22} * b_{21} + a_{23} * b_{31} + a_{24} * b_{41} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ a_{41} * b_{11} + a_{42} * b_{21} + a_{43} * b_{31} + a_{44} * b_{41} & \dots & \dots & \dots \end{pmatrix}_{4 \times 3}$$

For I = 1 to Row

For J = 1 to Col

c [I , J] = 0

For K = 1 to Col

C[I , J] = C [I , J] + A[I , K] * B[K , J]

End for K

End for J

End for I

อะเรย์ 2 มิติที่มีรูปแบบพิเศษ

สืบเนื่องมาจากที่เมตริกที่เราดำเนินงานทางคณิตศาสตร์นั้นก็ถือเสมือนว่าเป็นอะเรย์ชนิด 2 มิติเมื่อนำรูปแบบมาจัดเก็บในเครื่องคอมพิวเตอร์ ดังนั้นการศึกษาเมตริกชนิดพิเศษต่างๆ ก็จะมีผลกระทบต่อประสิทธิภาพในการไปดำเนินงานด้วย Programming Method ด้วยวิธีจัดการกับข้อมูลในรูปแบบของอะเรย์ 2 มิติ ด้วย

เมตริกชนิดพิเศษต่างๆที่ปรากฏในการนำไปใช้งาน

● Sparse Matrices

เป็นเมตริก ที่มีลักษณะพิเศษคือจะมีสมาชิกส่วนใหญ่เป็น 0 ดังนั้นในการจัดเก็บมักจะมีการย่อรูป เพื่อให้ประหยัดพื้นที่แต่โดยที่ยังคงมีรูปลักษณะความหมายเหมือนเดิม การจัดเก็บในที่นี้อาจจะหมายถึงการจัดเก็บในสมองเครื่องคอมพิวเตอร์ หรือ อาจจะจัดเก็บบนสื่อก็ได้ ในการจัดเก็บบนสื่อที่เราจะใช้แนวคิดในการบีบอัดแฟ้ม (Compress File) และภายหลังเมื่อจะนำกลับมาใช้ใหม่นั้นก็ใช้วิธีการระเบิดออกมาใช้ (Exploding)

ตัวอย่างของ Sparse Matrices

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}_{4 \times 8}$$

วิธีการจัดเก็บในสมองเครื่องนั้นอาจจะออกแบบได้หลายวิธี เช่นใช้ อะเรย์ 2 มิติ เก็บเป็นเมตริกเต็มรูปดังตัวอย่างที่ปรากฏของเมตริก A ก็เป็นอะเรย์ 2 มิติ มีขนาด 4 แถวและแถวละ 8 สดมภ์ หรืออาจจะเก็บแบบย่อในรูปแบบอะเรย์ชนิดมิติเดียวจัดเก็บ หรือใช้อะเรย์ 2 มิติในการจัดเก็บก็ได้ หรืออาจจะเทคนิคที่ซับซ้อนขึ้นโดยการเก็บใช้เทคนิคการเชื่อมโยงก็ได้ ในที่นี้จะแสดงใช้วิธีการเก็บย่อโดยวิธีของอะเรย์ดังนี้คือ

1. โครงสร้างแบบที่ 1 การใช้อะเรย์มิติเดียวในการเก็บเมตริก A โดยการสร้างที่เก็บในอะเรย์มิติเดียวคือเมตริก AC เพื่อเก็บคุณลักษณะของข้อมูลคือตำแหน่ง (อันประกอบด้วยแถวและสดมภ์) และมูลค่าของข้อมูล ดังนั้นจากเมตริก A เราจะสร้างเป็น Row Vector เพื่อจัดเก็บในคอมพิวเตอร์ในรูปแบบของอะเรย์มิติเดียว AC ได้ดังนี้

$$AC = [(1, 7, 3), (2, 8, 5), (3, 2, 2), (3, 3, 1)]$$

ความหมายตามที่ปรากฏในการจัดเก็บก็คือ (1, 7, 3) คือข้อมูลแถวที่ 1 สดมภ์ที่ 7 จะมีมูลค่าเป็น 3

การจัดเก็บในโครงสร้างนี้จะมีความหมายว่าสมาชิกหนึ่งตัวของเมตริก A จะต้องประกอบด้วยสมาชิกย่อยของ เมตริก AC ถึง 3 สมาชิกด้วยกัน ดังเช่น ข้อมูลที่ปรากฏ (1,7,3) จะหมายความว่าสมาชิกตำแหน่งที่ แถวเป็น 1 และสดมภ์เป็น 7 จะมีมูลค่าเป็น 3

2. โครงสร้างแบบที่ 2 การใช้อะเรย์ 2 มิติในการเก็บเมตริก A โดยการสร้างที่เก็บในอะเรย์ 2 มิติคือ AB สำคัญและมูลค่าของข้อมูลของเมตริก A ดังนั้นโครงสร้างของ X จะปรากฏดังนี้

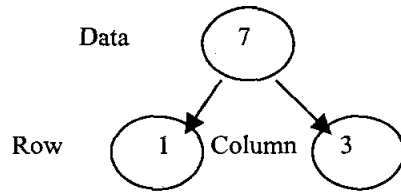
$$X = \begin{pmatrix} 1 & 7 & 3 \\ 2 & 8 & 5 \\ 3 & 2 & 2 \\ 3 & 3 & 1 \end{pmatrix}$$

ความหมายในการนำไปใช้งานก็จะลักษณะคล้ายกับโครงสร้างในแบบที่ 1 โดยที่เมตริกที่ทำการข้อนี้อาจมีลักษณะพิเศษคือเป็น เมตริก ที่มีจำนวน 3 สดมภ์เท่านั้น ส่วนจำนวนแถวนั้นจะขึ้นอยู่กับว่าสมาชิกในเซต A นั้นจะมีจำนวนสมาชิกที่ไม่เป็น 0 อยู่ที่จำนวน ในตัวอย่างนี้ เมตริก A จะมีสมาชิกที่ไม่เป็น 0 อยู่ทั้งสิ้น 4 จำนวน ดังนั้นจำนวนแถวของ เมตริก X จึงมีขนาดเท่ากับ 4 แถวเป็นต้น สมาชิกในสดมภ์ที่ 1 นั้นจะทำหน้าที่ชี้ตำแหน่งของ แถว ในขณะที่สมาชิกสดมภ์ที่ 2 จะทำหน้าที่ชี้สดมภ์ และ สมาชิกสดมภ์ที่ 3 จะเป็นค่าของข้อมูล

3. โครงสร้างแบบที่ 3 การใช้อะเรย์มิติเดียวจำนวน 3 อะเรย์เพื่อทำหน้าที่ในการเก็บสารสนเทศของเมตริก A โดยการสร้างที่เก็บในอะเรย์มิติเดียวขึ้นมา 3 อะเรย์ คือ A1 , A2, A3 โครงสร้างของ A1 , A2 , A3 จะปรากฏดังนี้

A1	1	2	3	2
A2	7	8	2	3
A3	3	5	2	1

ลักษณะของโครงสร้างแบบนี้จะคล้ายกับเก็บข้อมูลด้วยกิ่งต้นไม้ดังตัวอย่างส่วนหนึ่งดังนี้



ข้อสังเกต ในการจัดเก็บทั้ง 3 วิธีที่เสนอมานี้ เราจะต้องกำหนดคุณลักษณะของเมตริกอีก 2 จำนวนคือขนาดของแถวและขนาดสดมภ์ของ Sparse Matrices มิฉะนั้นจะไม่สามารถทำการ Explode ออกมาได้ถูกต้องเหมือนกับ Sparse Matrices เดิมที่เต็มรูปแบบก่อนย่อ

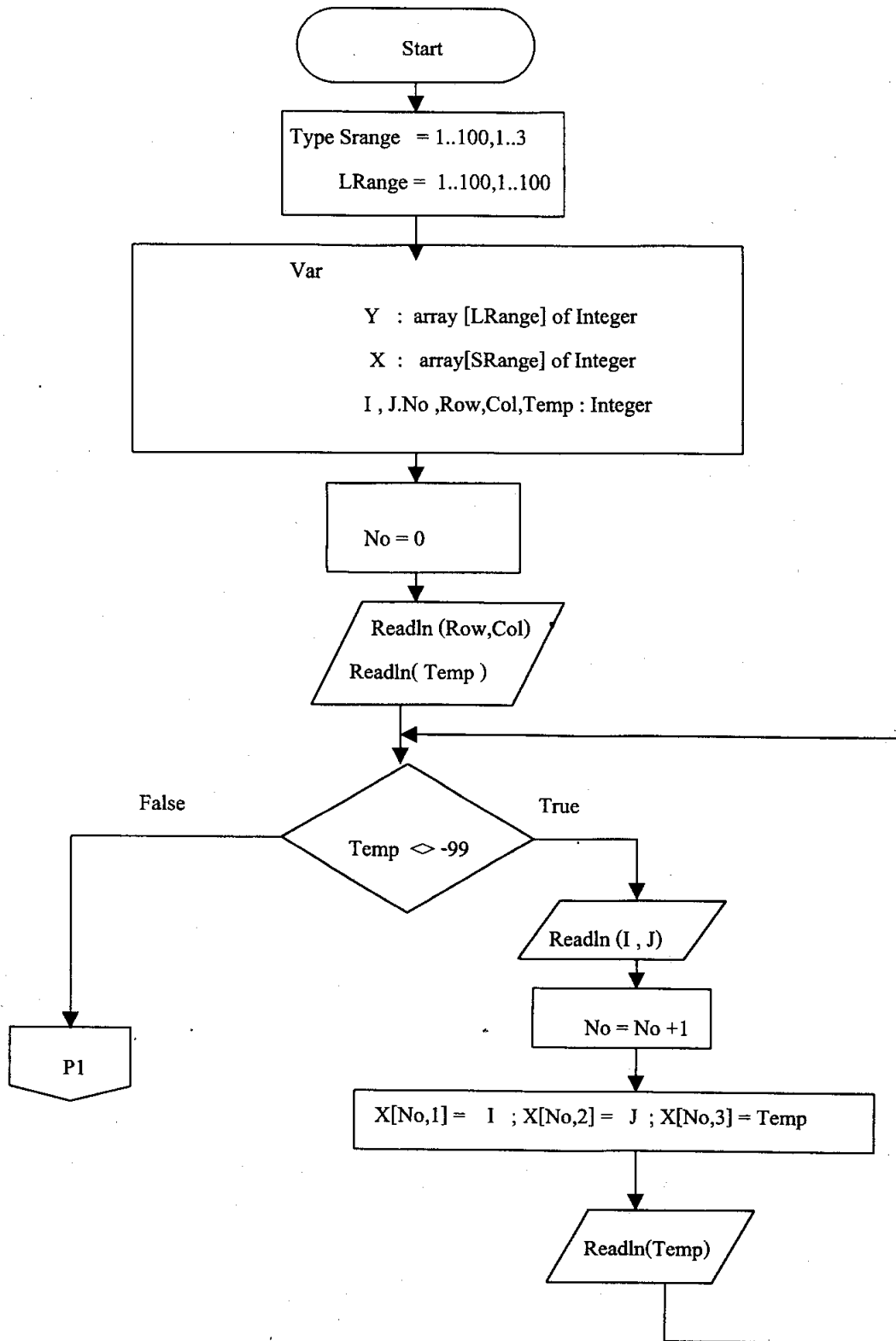
ส่วนวิธีการจัดเก็บข้อมูลบนสื่อที่บันทึกนั้นก็สามารถใช้หลักการใดหลักการหนึ่งก็ได้นำไปประยุกต์ให้เข้ากับการจัดเก็บข้อมูลเข้าแฟ้มต่อไปเพียง แต่จะต้องมีการเพิ่มสารสนเทศบางส่วนเข้าไปเพื่อยังผลให้การ Explode ข้อมูลออกมาสามารถกระทำได้อย่างถูกต้อง ตัวอย่างเช่นต้องเพิ่มขนาดของแถวและสดมภ์ของเมตริกขนาดเต็มรูปเข้าไปในแฟ้มข้อมูลด้วย

การจะ Explode สามารถกระทำได้หลายวิธีเช่นกัน ขึ้นอยู่กับการเลือกรูปแบบของการจัดเก็บ (Compress) ข้อมูลว่าจะใช้รูปแบบใดในตัวอย่าง 3 รูปแบบที่แสดงมาให้ดู ดังตัวอย่างต่อไปนี้จะแสดงขั้นตอนการการ Explode จาก เมตริก X จากอะเรย์ 2 มิติสู่ เมตริก Y ซึ่งเป็นเมตริกขนาด 4 x 8 แต่เพื่อที่จะทำให้โปรแกรมที่ออกแบบสามารถ ทำการ Explode จากอะเรย์ X ที่มีขนาดสูงสุดไม่เกิน 100 เซล เราจะกำหนดให้ X เป็นอะเรย์ที่เป็นไปได้สูงสุดคือ 100 ดังนั้น อะเรย์ Y จะเป็นอะเรย์ 2 มิติที่มีขนาด 100 แถว x 100 สดมภ์ ตัวอย่างที่จะดำเนินการ Explode ต่อไปนี้ จะใช้โครงสร้างแบบที่ 2 ในการเก็บโครงสร้างการเก็บข้อมูลชนิดบีบอัด

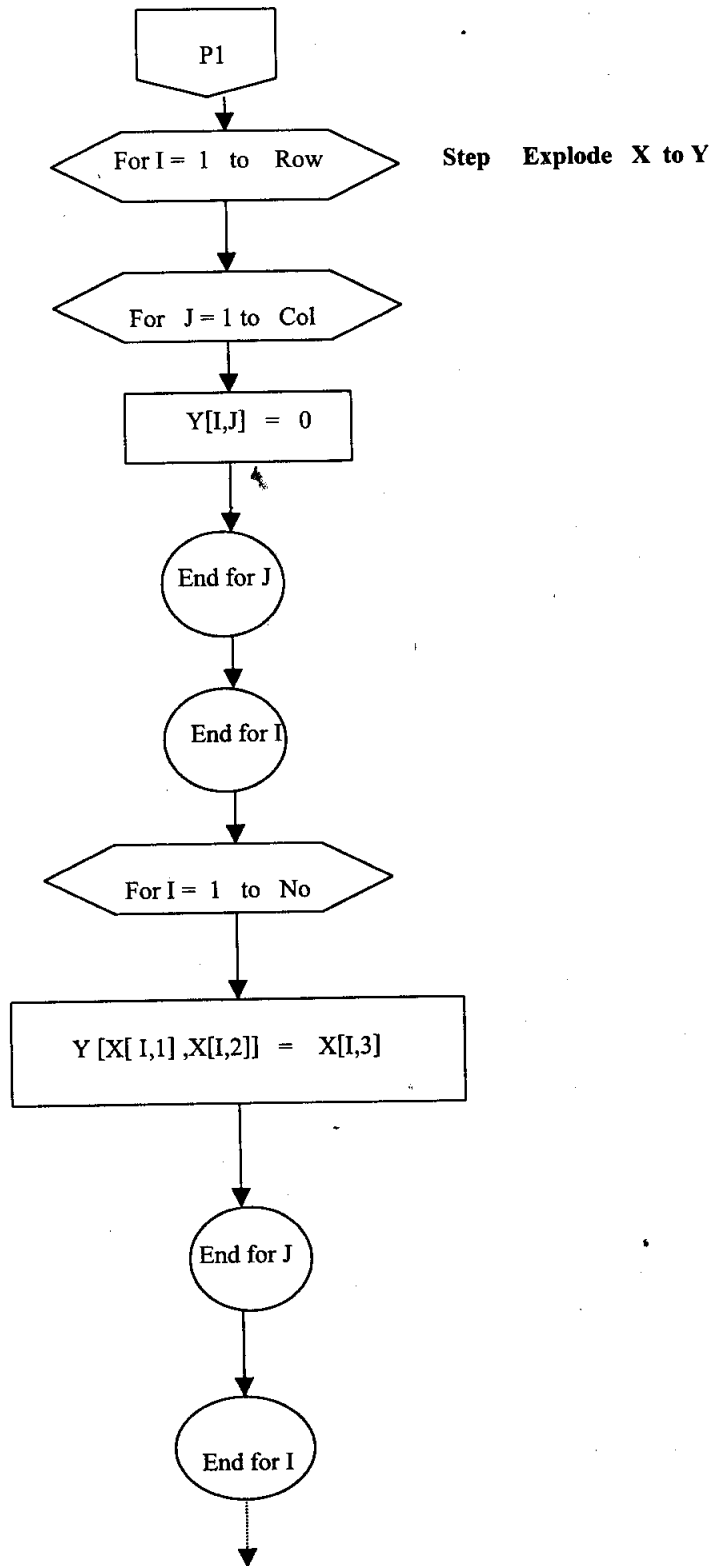
$$X = \begin{vmatrix} 1 & 7 & 3 \\ 2 & 8 & 5 \\ 3 & 2 & 2 \\ 3 & 3 & 1 \end{vmatrix}$$

การวิเคราะห์และข้อจำกัดในการดำเนินงานของโปรแกรม

- จะทำงานในการ Explode กับ อะเรย์ X ที่ Explode แล้วไปสู่อะเรย์ Y ที่มีขนาดไม่เกิน 100 แถว x 100 สดมภ์
- การรับจำนวนสมาชิกของอะเรย์ที่จะนำไป Explode ในที่นี้ก็คืออะเรย์ X จะตรวจสอบว่าสมาชิกใดๆของ X [1,3] จะต้องมีค่าไม่เท่ากับ -99 ดังนั้นเราจะใช้ตัวเลขข้อมูลคือ -99 เป็นเงื่อนไขของการยุติการรับข้อมูลจากอะเรย์ X

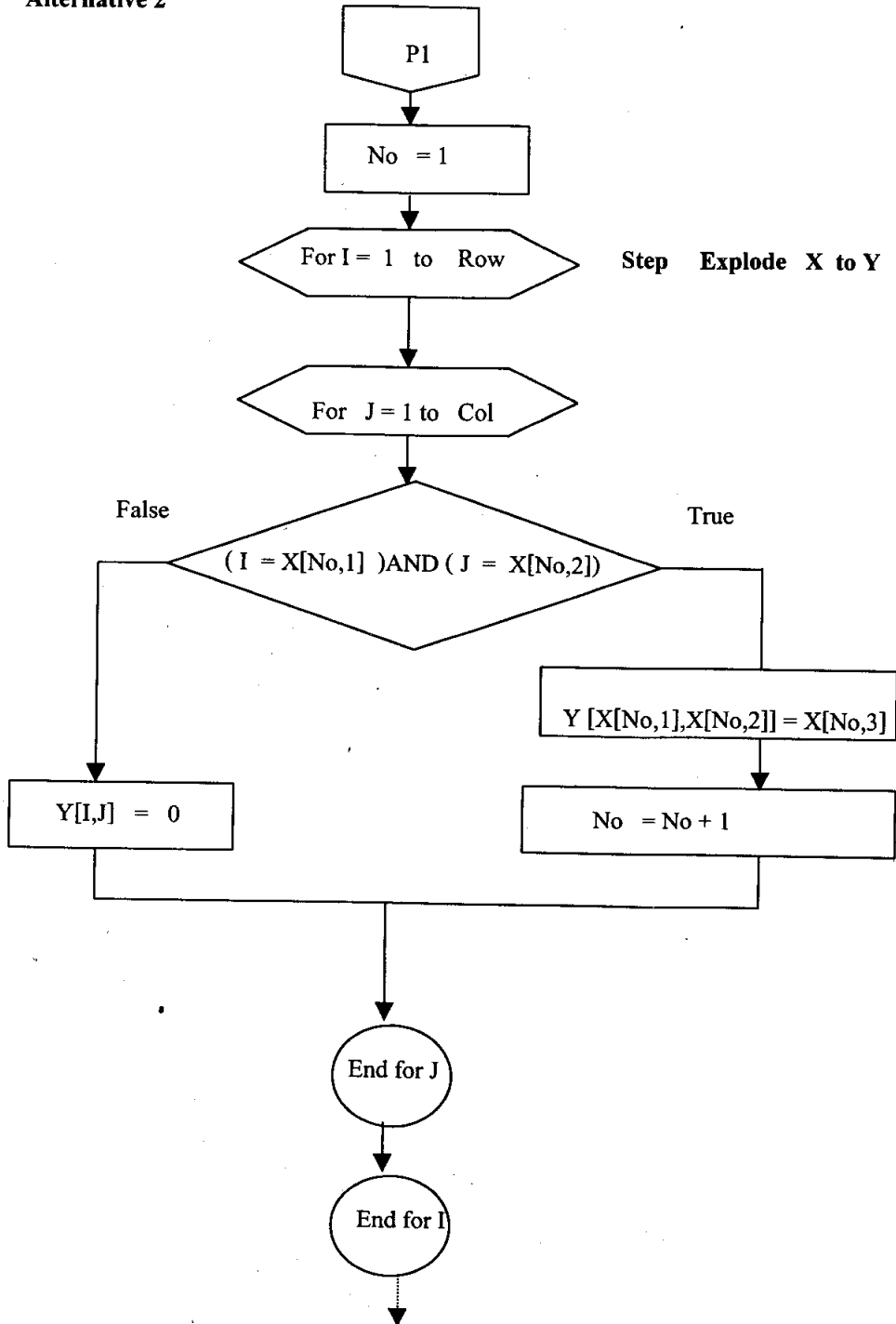


Alternative 1



ในส่วนของการ Explode นี้เราสามารถที่รวมเอา 2 Loop (Loop ซ้อน) เข้าไว้ด้วยกัน เพื่อเป็นการประหยัดเวลาในการวน Loop ได้ดังนี้คือ

Alternative 2



● **Diagonal Matrices** เป็นเมตริกพิเศษที่มีลักษณะที่มีสมาชิกบน Diagonal

ไม่เป็น 0 ส่วนสมาชิกตัวอื่นในส่วนของ Off Diagonal จะเป็น 0 ทั้งหมด ให้ดูจากตัวอย่างต่อไปนี้

$$X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}_{4 \times 4}$$

การจัดเก็บข้อมูลในส่วนของสมองเครื่องหรือการบันทึกบนสื่อสำหรับ เมตริก ชนิดนี้ค่อนข้างจะกระทำได้ง่ายด้วยเหตุผลที่ว่า คุณลักษณะของเมตริกประเภทนี้ค่อนข้างจะชัดเจน เพราะ สมาชิกที่ไม่ปรากฏบนเส้นทแยงมุมจะเป็น 0 ทุกสมาชิก ดังนั้นจำนวนสมาชิกที่อยู่บนเส้นทแยงมุมจึงมีจำนวนเท่ากับขนาดของเมตริกพอดี โดยที่เมตริกดังกล่าวก็จะมีขนาดของแถวกับสดมภ์จะเท่ากัน

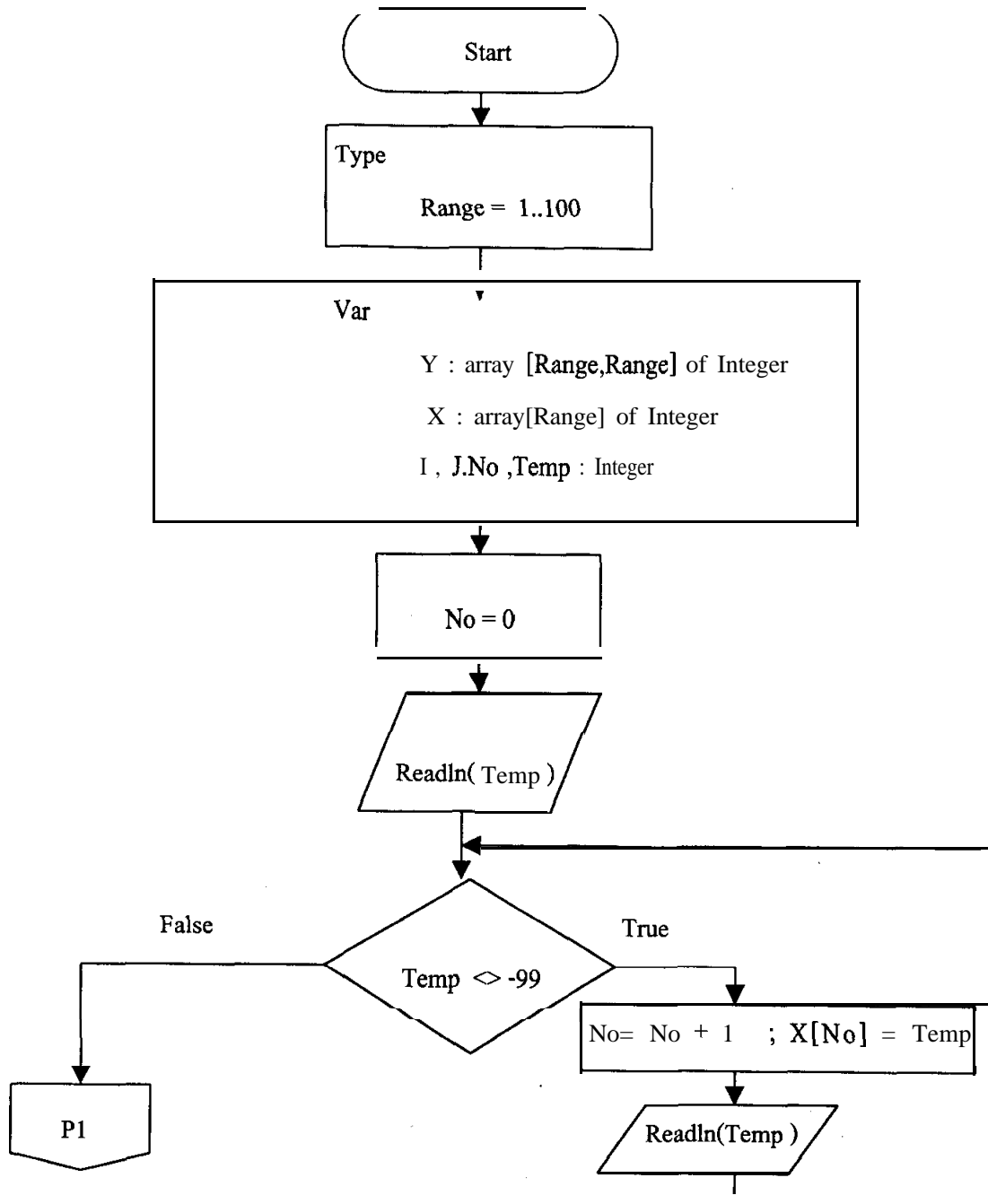
วิธีการจัดเก็บในสมองเครื่องหรือบันทึกบนสื่อ นั้นอาจจะออกแบบได้ดังนี้คือการใช้ อะเรย์มิติเดียว ที่มีจำนวนเซลล์เท่ากับขนาดของแถว (หรือสดมภ์) ของเมตริก X ดังนี้

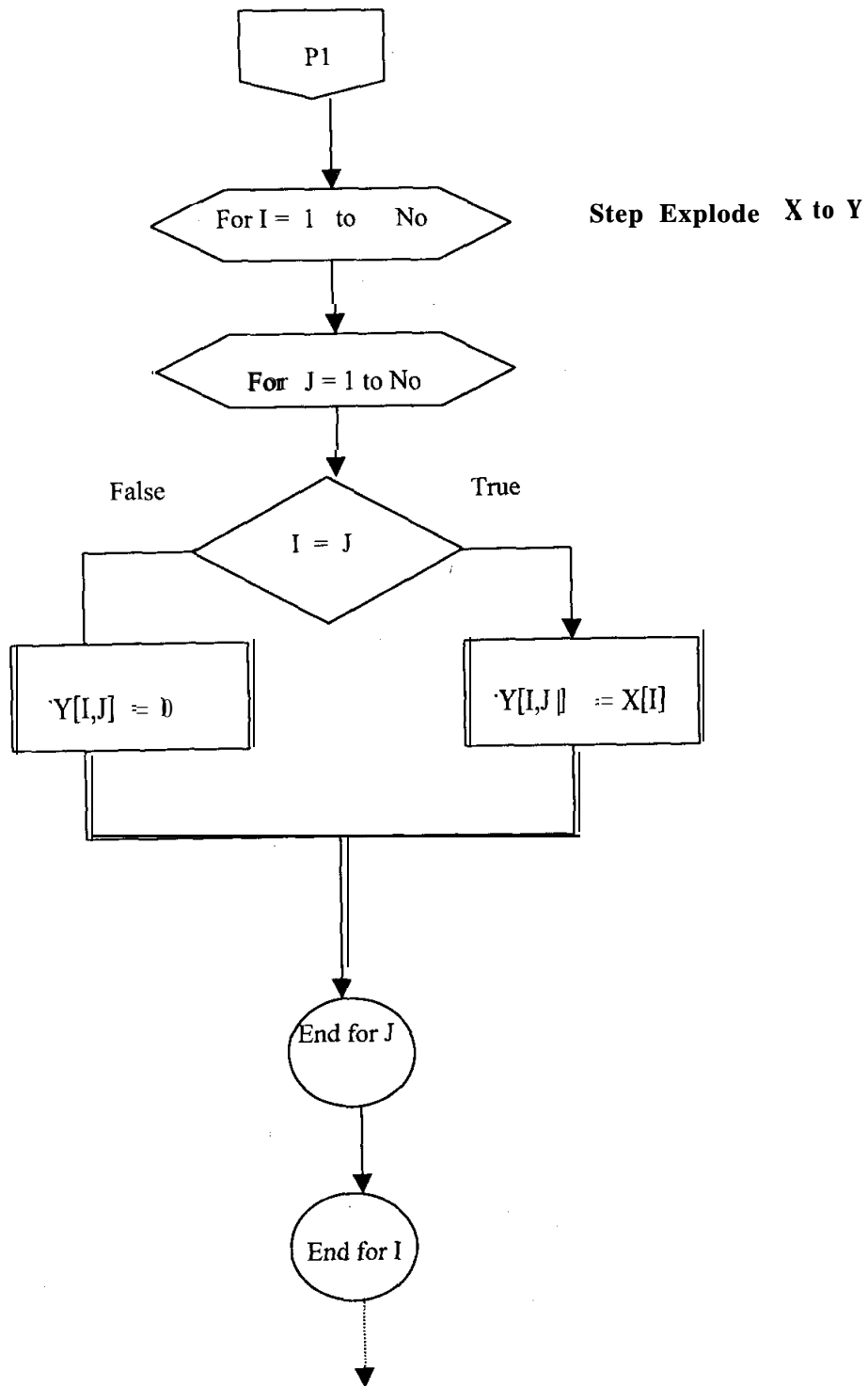
	X[1]	X[2]	X[3]	X[4]
X	1	5	8	2

ส่วนการที่จะ Explode ก็สามารถกระทำได้ง่ายๆเช่นกัน ดังตัวอย่างการแสดงขั้นตอนการการ Explode จาก เมตริก X จากอะเรย์มิติเดียวสู่ เมตริก Y ซึ่งเมตริกขนาด 4 x 4 แต่เพื่อที่จะทำให้โปรแกรมที่ออกแบบสามารถ ทำการ Explode จากอะเรย์ X ที่มีขนาดสูงสุดไม่เกิน 100 เซล เราจะกำหนดให้ X เป็นอะเรย์ที่เป็นไปได้สูงสุดคือ 100 ดังนั้น อะเรย์ Y จะเป็นอะเรย์ 2 มิติที่มีขนาด 100 แถว x 100 สดมภ์

การวิเคราะห์และข้อจำกัดในการดำเนินงานของโปรแกรม

- จะทำงานในการ Explode กับ อะเรย์ที่ Explode แล้วไม่เกิน 100 แถว x 100 สดมภ์
- การรับจำนวนสมาชิกของอะเรย์ที่จะนำไป Explode ในที่นี้ก็คืออะเรย์ X จะตรวจสอบว่าสมาชิกใดๆของ X [I] จะต้องมีค่าไม่เท่ากับ -99 ดังนั้นเราจะใช้ตัวเลขข้อมูลคือ -99 เป็นเงื่อนไขของการยุติการรับข้อมูลจากอะเรย์ X





● **Symmetric Matrices** เป็นเมตริกที่มีลักษณะพิเศษคือ Cell ที่ $X[I, J] = X[J, I]$ หรืออาจจะสรุปได้ว่าเมตริกใดๆที่ เช่นเมตริก $X^T = X$ เราจะเรียกว่าเมตริกนั้นมีคุณสมบัติเป็น Symmetric Matrices ตัวอย่างของเมตริกประเภทนี้

$$X = \begin{pmatrix} 1 & 3 & 5 & 7 \\ 3 & 5 & 2 & 1 \\ 5 & 2 & 9 & 2 \\ 7 & 1 & 2 & 6 \end{pmatrix}$$

วิธีการจัดเก็บเมตริกประเภทนี้ในสมองเครื่องหรือบนแฟ้มข้อมูลนั้นเราสามารถออกแบบได้ 2 วิธีคือ

1. จัดเก็บคุณสมบัติของอะเรย์คือขนาดของอะเรย์ และมูลค่าของข้อมูลที่ปรากฏในส่วนที่เป็นเส้นทแยงมุมขึ้นไปดังภาพนี้

$$A = \begin{pmatrix} 1 & 3 & 5 & 7 \\ 3 & 5 & 2 & 1 \\ 5 & 2 & 9 & 2 \\ 7 & 1 & 2 & 6 \end{pmatrix} \quad 4 \times 4$$

ดังนั้นอะเรย์ที่บีบอัดแล้วจะปรากฏดังนี้

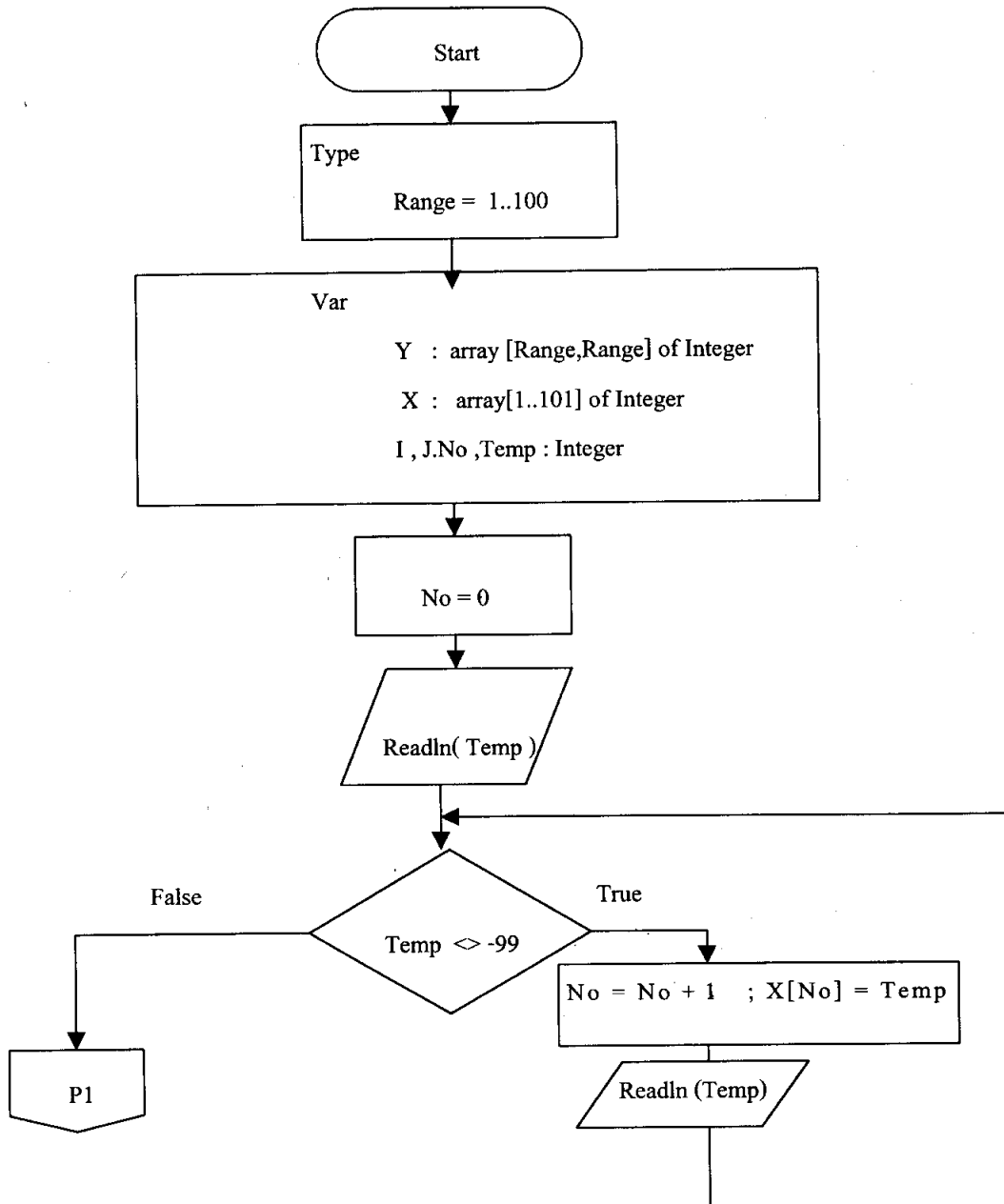
4	1	3	5	7	5	2	1	9	2	6
---	---	---	---	---	---	---	---	---	---	---

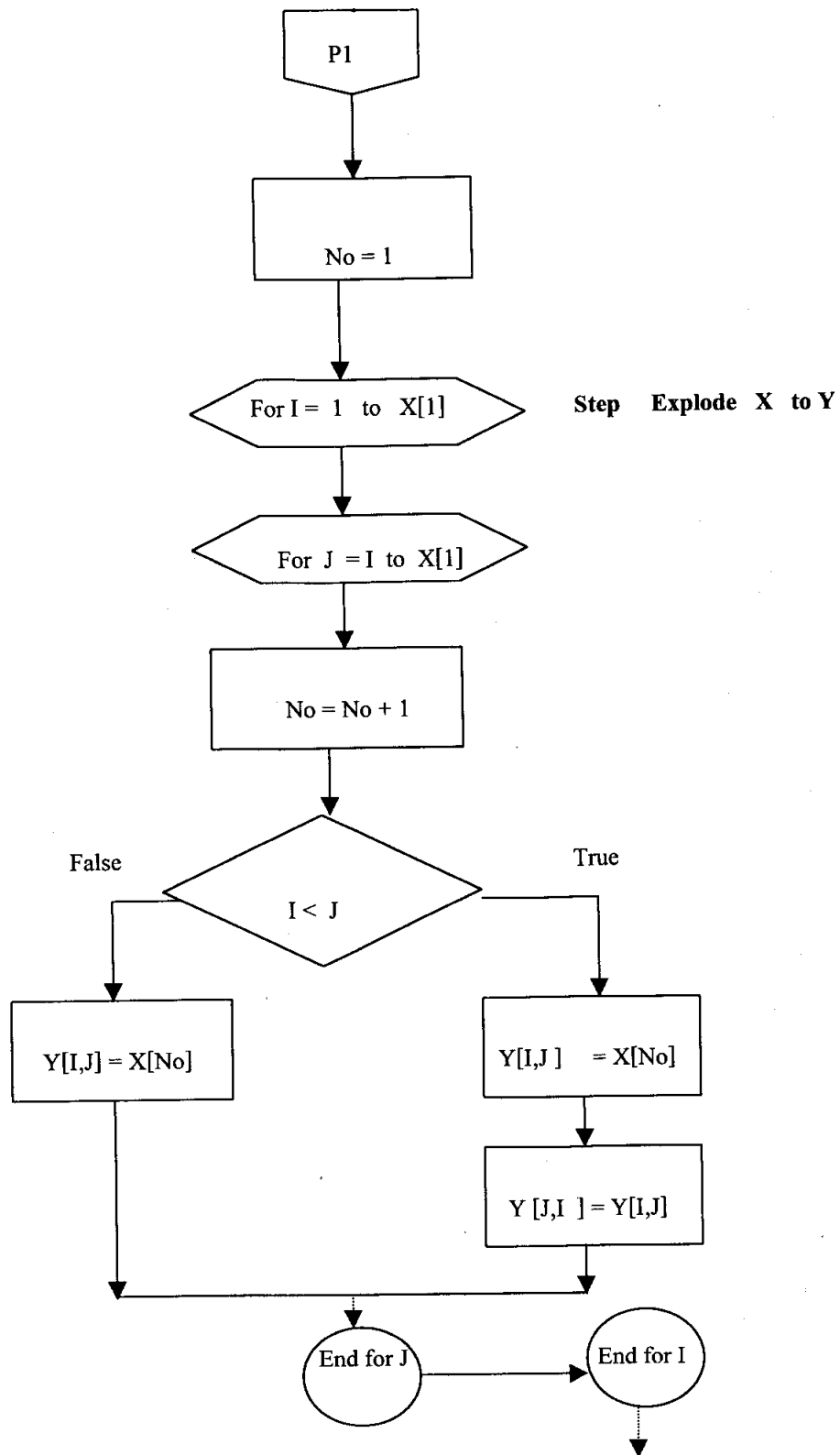
X[1] X[11]

ในที่นี้ X[1] จะหมายถึงจำนวนแถวและสดมภ์ของเมตริก A ส่วน X[I] ; I = 2,3,...,11 จะหมายถึงข้อมูลของเมตริก A วิธีการเก็บสารสนเทศที่ทำการบีบอัด (Compress) เช่นนี้จะส่งผลให้การ Explode นั้นกระทำได้ง่าย เพราะเรามีสารสนเทศขนาดของแถว (สดมภ์) ปรากฏอยู่แล้ว แต่จะมีข้อเสียที่ว่าเราจะต้องเสียพื้นที่ในการจัดเก็บคุณลักษณะนี้ไว้ ผังโปรแกรมต่อไปนี้จะแสดงวิธีการนำข้อมูลจากอะเรย์ X ไปทำการ Explode ให้เป็นอะเรย์ Y โดยมีข้อจำกัดว่าจะใช้สำหรับการ Explode อะเรย์ที่มีขนาดสูงสุด 100 x 100 เท่านั้น

การวิเคราะห์และข้อจำกัดในการดำเนินงานของโปรแกรม

- จะทำงานในการ Explode กับ อะเรย์ที่ Explode แล้วไม่เกิน 100 แถว และ 100 สดมภ์
- การรับจำนวนสมาชิกของอะเรย์ที่จะนำไป Explode ในที่นี้ก็คืออะเรย์ X จะตรวจสอบว่าสมาชิกใดๆของ X [I] จะต้องไม่เท่ากับ -99 ดังนั้นเราจะใช้ตัวเลขข้อมูลคือ -99 เป็นเงื่อนไขของการยุติการรับข้อมูลจากอะเรย์ X





2. การจัดเก็บวิธีนี้จะดำเนินการคล้ายกับวิธีที่ 1 แต่จะต่างกันตรงที่ว่าจะไม่เก็บค่าของแถว(Symmetric Matrices จะมีคุณสมบัติว่า แถวและสทมภ์จะมีขนาดเท่าๆกัน) ไว้ แต่จะเก็บเฉพาะมูลค่าของข้อมูลในตำแหน่งที่กำหนด (ดูรูปวิธีการเก็บประกอบ) เท่านั้นซึ่งจะทำให้เราประหยัดพื้นที่ในการเก็บข้อมูลที่เป็นคุณลักษณะคือขนาดของแตรริก วิธีนี้ก็จะมผลเสียในส่วนที่ว่าเราจะต้องออกแบบในการคำนวณขนาดของ เมตริกที่จะนำมาใช้งานเอง ซึ่งจะทำให้เสียเวลา แนวทางในการคำนวณหาขนาดของเมตริกจะดำเนินการดังนี้คือ

$$X = \begin{pmatrix} 1 & 3 & 5 & 7 \\ 3 & 5 & 2 & 1 \\ 5 & 2 & 9 & 2 \\ 7 & 1 & 2 & 6 \end{pmatrix}_{4 \times 4}$$

การออกแบบการจัดเก็บเพื่อประหยัดพื้นที่ที่จะเก็บเฉพาะข้อมูลดังต่อไปนี้คือ 1,3,5,7,5,2,1,9,2,6 = 10 สมาชิก โดยที่จำนวนสมาชิกทั้งหมดที่จะจัดเก็บนั้นนับได้จากการคำนวณดังนี้คือ

$$\text{Count} = 5 + 4 + 3 + 2 + 1 = 15$$

ในกรณีที่เรามีเมตริกขนาด 100×100 ดังนั้นจำนวนสมาชิกจะคำนวณได้ดังนี้

$$\text{Count} = n + (n-1) + (n-2) + \dots + 4 + 3 + 2 + 1$$

$$\text{Count} = \sum_{i=1}^n X_i = \frac{n(n+1)}{2}$$

จากสูตรที่ได้นี้ ถ้าเรากำหนดขนาดของอะเรย์ให้มีขนาดเท่ากับ 100×100 แล้วเมตริกเต็มรูปจะมีขนาดเท่ากับ $100 \times 100 = 10,000$ เซลล์ ถ้ากำหนดให้ขนาดเซลล์ละ 2 Bytes แล้วจะต้องใช้พื้นที่ในการจัดเก็บภายในสมองเครื่องเท่ากับ $10,000 \times 2 = 20 \times 10^3$ Bytes ประมาณ 20 K Bytes ซึ่งเป็นพื้นที่ที่ค่อนข้างจะมาก ดังนั้นถ้าเราทำการบีบอัดแล้วจะคำนวณพื้นที่ที่ใช้ดังนี้

$$\text{Count} = \frac{100(100+1)}{2} = 5050 \text{ Bytes}$$

พื้นที่ดังกล่าวจะมีค่าประมาณ 5 K Bytes ซึ่งลดจากพื้นที่เต็มรูปไป 3 เท่า การจัดเก็บข้อมูลตามวิธีที่ 2 จะเก็บเหมือนวิธีที่ 1 แต่ไม่มีขนาดของแถว ดังนั้นเราสามารถนำจำนวนสมาชิกที่นับได้แล้วย้อนรอยหาค่าของขนาดของเมตริกได้ดังนี้คือ

สมมติว่า Symmetric Matrices มีขนาด 10×10 ดังนั้นสมาชิกที่นับได้จากอะเรย์ที่บีบอัดแล้วจะมีค่าเท่ากับ 55 เราจะคำนวณหาขนาดของเมตริกได้ดังนี้

$$\text{Count} = 55 = \frac{n(n+1)}{2}$$

$$2(55) = n(n+1)$$

$$100 = n(n+1)$$

$$\therefore n = \text{Int}(\text{Sqrt}(100))$$

$$\therefore n = 10$$

หมายเหตุ $n = \text{Int}(\text{Sqrt}(100))$ หมายถึงนำค่าของ 100 ไปหาค่ารากที่สองแล้วจึงนำค่ารากที่สองที่ได้มาปัดเศษเก็บแต่เลขจำนวนเต็มไว้ ดังนั้น n ก็คือขนาดของเมตริกนั่นเอง การดำเนินการเช่นนี้เราจะต้องทำการนับสมาชิกที่อยู่ในอะเรย์ที่บีบอัดแล้วว่ามีจำนวนจากนั้นนำมาหาค่า n แล้วจึงจะสามารถนำไปดำเนินการ Explode อะเรย์ ได้ จะพิจารณาได้ว่าวิธีดำเนินการออกแบบเช่นนี้โดยประหยัดพื้นที่ไปเพียงไม่กี่ Bytes สำหรับการเก็บคุณลักษณะขนาดของเมตริกดูแล้วไม่คุ้มค่ากับเวลาที่สูญเสียไปในการคำนวณหาค่าของ n เพื่อประกอบการหาพื้นที่สำหรับเมตริกดังกล่าวอีกด้วย สรุปแล้วสำหรับ 2 วิธีนี้ที่นำเสนอมานี้เราจะเลือกวิธีที่ 1 จะดีกว่า

คำถามท้ายบท

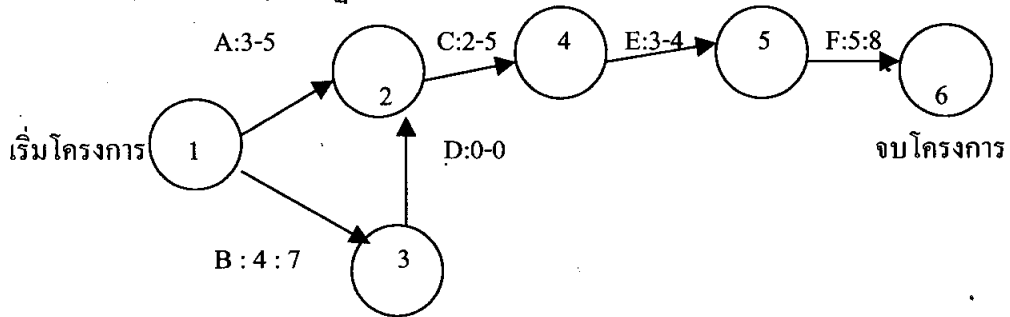
1. กำหนดเมตริกดังนี้คือ
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 \\ 0 & 5 & 7 & 0 \\ 0 & 0 & 0 & 6 \end{pmatrix}$$
 จงเขียนโครงสร้างการจัดเก็บโดยยึดหลัก

ประหยัดที่สุด และภายหลังการออกแบบให้นำโครงสร้างที่ได้ไป Explode ให้เป็นเมตริกเต็มรูปตามที่กำหนดให้

2. กำหนดให้เมตริกพิเศษ $X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix}$ จงเขียนผังโปรแกรมเพื่อรับข้อมูลดัง

กล่าวเข้าไปจัดเก็บในสมองเครื่อง ภายหลังการจัดเก็บให้หาผลรวมของสมาชิกบนเส้นทแยงมุม

3. กำหนดให้โครงการในการก่อสร้างงานหนึ่งปรากฏเป็นเส้นทาง(ระยะเวลา)ในการทำงานและกิจกรรมต่างๆปรากฏดังนี้



โดยที่ A หมายถึงกิจกรรมต่างๆที่ดำเนินการ



C : 2 – 5 หมายถึงกิจกรรม C ที่ดำเนินการ โดยใช้เวลาเร็วที่สุดในการดำเนินการเท่ากับ 2 วัน และระยะเวลาที่ช้าที่สุดในการดำเนินการ C เท่ากับ 5 วัน

จากข้อกำหนดที่ให้นี้ ให้ออกแบบโครงสร้างข้อมูล และค้นหาเส้นทางที่สั้นที่สุด และยาวที่สุดที่โครงการนั้นจะดำเนินการเสร็จ และให้ระบุว่าในเส้นทางที่ยาวที่สุดนั้นมีกิจกรรมใดที่ปรากฏอยู่บนเส้นทางนี้

4. กำหนดเมตริก $X = \begin{pmatrix} 1 & 3 & 0 & 5 \\ 2 & 8 & 7 & 2 \\ 6 & 4 & 3 & 10 \\ 8 & 2 & 0 & 3 \end{pmatrix}$ โดยที่ เมตริก X มีความหมายดังนี้ แถว

หมายถึง Brand Name ของเสื้อคือ A, B, C, D ส่วนสดมภ์หมายถึงขนาดของเสื้อคือ เบอร์

1, 2, 3, 4 โดยที่เมตริก $P = \begin{pmatrix} 250 & 566 & 650 & 780 \\ 485 & 524 & 598 & 600 \\ 323 & 425 & 480 & 580 \\ 402 & 555 & 650 & 750 \end{pmatrix}$ หมายถึงราคาของเสื้อแต่ละ

ยี่ห้อของแต่ละเบอร์

❖ จงเขียนผังโปรแกรมเพื่อทำการหาผลรวมของราคาเสื้อแต่ละยี่ห้อ

❖ สมมุติว่าจะมีการลดราคาลง โดยที่เสื้อแต่ละเบอร์จะลดราคาลง

ดังนี้ $\begin{pmatrix} 40 \\ 20 \\ 56 \\ 25 \end{pmatrix}$ ดังนั้นราคารวมของเสื้อแต่ละยี่ห้อจะลดจาก

เดิมเท่าไร

การดำเนินงานตอบคำถามทั้ง 2 ข้อนี้ให้เขียนด้วยผังโปรแกรม

5. กำหนดให้คนงานของโรงงานแห่งหนึ่งซึ่งมีจำนวน 50 คน แต่ละคนจะทำงานสัปดาห์ละ 5 วัน (จันทร์-ศุกร์) ยกเว้นบางคนที่อาจจะทำวันเสาร์และอาทิตย์ด้วย ในกรณีที่ทำงานในวันเสาร์และอาทิตย์ จะได้ค่าแรงชั่วโมงเพิ่มจากปกติอีก 1 เท่า โดยที่ค่าจ้างชั่วโมงปกติคือ 25 บาท จงออกแบบผังโปรแกรมในงานนี้เพื่อคำนวณค่าจ้างในแต่ละสัปดาห์ให้กับคนงานแต่ละคน