

2021 Spring 計算機程式設計

期中複習一 Problem7 參考解答

By TA 龔柏森

壹、 題意解析

首先根據題意得知輸入為一個 N 階方陣，輸出該方陣是否為一個幻方 (Magic Square)，並且此題為了簡化問題，已假設使用者輸入之方陣元素均在 1 至 n^2 之間未重複，我們不需要再另外檢查輸入。我們所要做的事情就是檢查輸入是否符合幻方的特性（即：每一行、每一列與兩個對角線的總和是否皆相等）。

貳、 程式規劃與虛擬碼

首先，我們使用題目所給的 `InputArray()` 函式取得輸入陣列，接著再使用一個 `isMagicSquare()` 函式來檢查輸入的方陣，如果是 Magic Square 則回傳 `true`，不是則回傳 `false`。有了上述初步規劃的構想，接下來可以試著將其改寫成簡易的虛擬碼，再轉換成程式碼。

```
static void Main(){
    mat=InputArray();           //第一步先根據題目提供的函式取得輸入
                                //InputArray()函式請參考題目，不另外撰寫
    if(isMagicSquare(mat))      //用 isMagicSquare()函式判斷並回傳布林值
        Write(此方陣是幻方)    //若 return true 輸出就是方陣
    else
        Write(此方陣非幻方)    //若 return false 輸出不是方陣
}

static bool isMagicSquare(){
    n = mat.GetLength(0);       //利用 mat.GetLength(0)取得方陣階數 n
    sum=n*(n*n+1)/2;            //先計算應得總和的值
                                //接著逐一檢查對角線以及每行每列
    for(i=0 to n-1){
        diag1+=mat[i,i];        //計算方陣左上至右下對角線
                                //即 mat[0,0]+[1,1]+.....+[n-1,n-1]的元素
        diag2+=mat[i,n-1-i];    //計算方陣右上至左下對角線
                                //即 mat [0,n-1]+[1,n-2]+[2,n-3]+.....+[n-1,0]
        rowSum=0,colSum=0;      //分別用來計算第 i 行和第 i 列總和
    }
}
```

```

        for(j=0 to n-1){           //在固定 i 的情況下, j 會從 0 跑到 n-1
                                   //利用這一點同時算 rowSum 和 colSum
            rowSum+=mat[i,j]; //第 i 列第 j 行的元素加入列總和
            colSum+=mat[j,i]; //第 j 列第 i 行的元素加入行總和
        }
        //算完第 i 列和第 i 行總和以後再檢查是否等於 sum
        if(rowSum!=sum or colSum!=sum)
            return false; //若不等於 sum 就傳回 false,無須繼續
    }
    if(diag1!=sum or diag2!=sum)//判斷兩對角線和,若其中之一不等於 sum
        return false;           //即 return false,無須再繼續
    //經過前面層層關卡檢查都沒有回傳 false,表示所有總和皆符合限制
    return true; //確認是幻方!
}
}

```

參、 程式實作

根據上述虛擬碼，此時我們就可以將其逐行改成合法的語法，使其完善成為一支程式如下：

```

using System;
namespace Midterm{
    class Program{
        static void Main(){
            int[,] mat = InputArray();
            if (isMagicSquare(mat))
                Console.WriteLine("此方陣是幻方");
            else
                Console.WriteLine("此方陣非幻方");
        }
        static int[,] InputArray(){
            //根據題目提供，不須另外再撰寫
        }
        static bool isMagicSquare(int[,] mat){
            int N = mat.GetLength(0); //取得方陣階數
            int sum = N*(N*N+1)/2;
            int diag1Sum=0,diag2Sum=0;

```

```

for (int i = 0; i < N; i++){
    diag1Sum=diag1Sum+mat[i,i]; //左上至右下對角線總和
    diag2Sum=diag2Sum+mat[i,N-1-i]; //右上至左下對角線總和
    int rowSum=0,colSum=0;
    for(int j=0;j<N;j++){
        rowSum+=mat[i,j]; //第 i 列總和
        colSum+=mat[j,i]; //第 i 行總和
    }
    if(rowSum!=sum || colSum!=sum)
        return false;
}
if(diag1Sum!=sum || diag2Sum!=sum)
    return false;
return true;
}
}
}

```

肆、 考前提醒事項

再次提醒同學：**考試不能攜帶任何電子通訊產品，僅能攜帶紙本資料**。因此屆時並**不會有電腦**幫忙 debug，希望同學能夠練習 **trace 程式碼** 的能力，同時頭腦順著程式碼的邏輯思維走一遍，並且在必要的時候加以紙筆輔助思考。在撰寫這類程式題的時候不妨**先畫出程式的流程圖或手寫簡略的虛擬碼**幫助思考（這題助教寫的虛擬碼較為冗長，主要是為了帶領同學一行一行 trace，考試時也可以再更簡化，虛擬碼只要自己看得懂就好），最後再**將虛擬碼逐行轉化為正確的程式碼**。若是遇到忘記或不熟的語法再回頭去查閱紙本資料，還有務必注意型別的轉換以及初始值的宣告，另外也請記得在程式碼中**適當加入註解**，有助老師及助教閱卷。