資結作業

姓名:楊燿寧

班級:資訊三丙

學號:D0745765

作業一說明

要用一個陣列做出多項式相乘，結果全部都要傳進去 那個陣列。排序也要在那裏做。

用類似以下的方法寫出來。

coef	startA	finishA	startB	finishB	
exp					

作業二說明

用 M*N 的矩陣來限定蟑螂走的範圍，初始要把矩陣 所有值歸零，每走一格記一次，直到所有格數都走過即結束。而我們要用隨機亂數來決定他要往哪個方位走。

作業一程式碼

```c
#include<stdio.h>
#include<stdlib.h>

struct data{
 int nth;    //係數
 int fac;    //次方
}A[1000],temp;

int main(){
    int i,j,cot=0,cot1=0,cot2=0;
    // ,  ,A的記數,B的記數,C的記數
    int startA=0,finishA,startB,finishB,startall,finishall;
    printf("輸入多項式A: \n");
    while(1){
        printf("輸入係數和次方: ");
        scanf("%d %d",&A[cot].nth,&A[cot].fac);
        if(A[cot].nth==0&&A[cot].fac==0){
            break;
        }
        else{
            cot++;   //A的記數來記A有幾項
        }
    }
    finishA=cot;
    startB=cot+1;
```

```c
    cot1=cot+1;
    printf("輸入多項式B: \n");
    while(1){

        printf("輸入係數和次方: ");
        scanf("%d %d",&A[cot1].nth,&A[cot1].fac);
        if(A[cot1].nth==0&&A[cot1].fac==0){
            break;

        }
        else{
            cot1++;      //用B的記數來記B有幾項
        }

    }
    finishB=cot1;
    startall=cot1+1;
    cot2=cot1+1;
    for(i=0;i<finishA;i++){
        for(j=startB;j<finishB;j++){
            A[cot2].fac=A[i].fac+A[j].fac;
            A[cot2].nth=A[i].nth*A[j].nth;
            cot2++; ////用ALL的記數來記ALL有幾項
        }
    }
```

```
    finishall=cot2;
    for(i=startall;i<finishall;i++){    //判斷次方是不是一樣
        for(j=i+1;j<finishall;j++){
            if(A[i].fac==A[j].fac){     //要是一樣把下一位的數值加到上一個，在把下一個值歸零避免被記到
                A[i].nth=A[i].nth+A[j].nth;
                A[j].nth=0;     // 係數歸零
                A[j].fac=0;     //次方歸零
            }
        }
    }
```

```
    for(i=startall;i<finishall;i++){     //排序
        for(j=startall;j<finishall-1;j++){
            if(A[j].fac<A[j+1].fac){
                temp.fac=A[j].fac;
                A[j].fac=A[j+1].fac;
                A[j+1].fac=temp.fac;

                temp.nth=A[j].nth;
                A[j].nth=A[j+1].nth;
                A[j+1].nth=temp.nth;
            }
        }
    }
    printf("相乘結果:");
    for(j=startall;j<finishall;j++){
```

```
        if(A[j].fac!=0){
            if(A[j].nth!=0)
                printf("%dx^%d",A[j].nth,A[j].fac);
            if(j+1<finishall&&A[j+1].nth>0)
                printf("+");
        }
        else if(A[j].fac==0&&A[j].nth!=0)
            printf("%d",A[j].nth);

    }
    printf("\n");
    return 0;

}
```

作業二程式碼

```c
#include<stdio.h>
#include<stdlib.h>
#include<time.h>


int main(){
    srand(time(NULL));
    int a,b,c,d,e,i,j;
    //n,m,row,column,

    printf("m,n,row,column: ");
    scanf("%d %d %d %d",&a,&b,&c,&d);
    int A[a][b];
    //矩陣大小
    for(i=0;i<a;i++){
        for(j=0;j<b;j++){
            A[i][j]=0;
        }
    }
    A[c][d]=1;
    //哪裡為起始

    if(a<=2||a>40||b<2||b>20){
        printf("ERROR\n");
    }
```

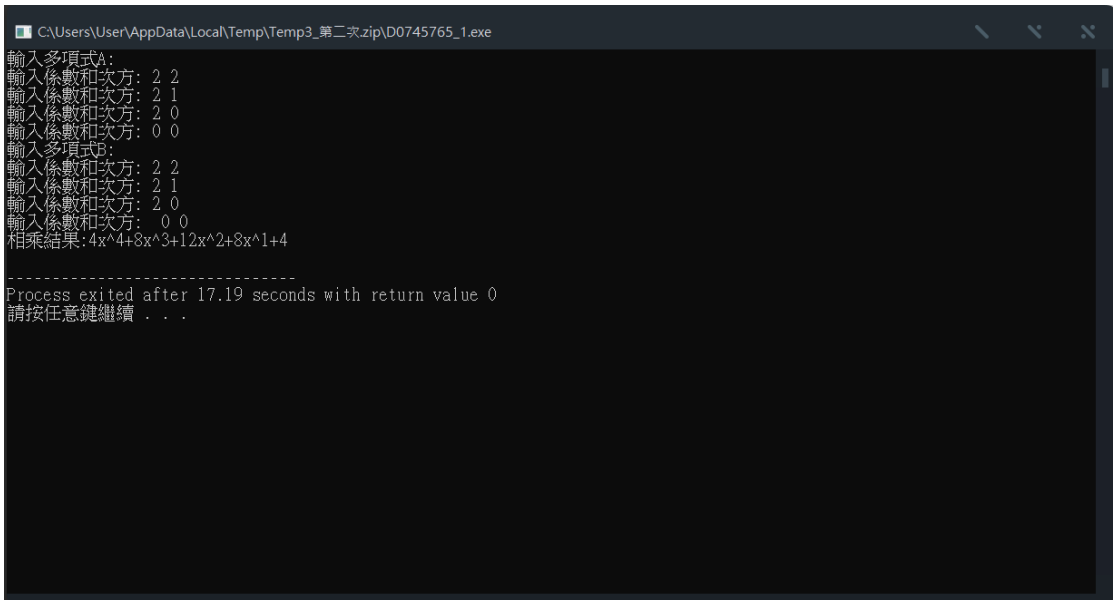
```c
    else{

        int con=0,temp=0;
        while(1){




            e=rand()%8;

            if(e==0&&(c-1>=0&&d-1>=0)){ //左上
                c--;
                d--;
                A[c][d]++;
                temp++;

            }
            else if(e==1&&d-1>=0){  //左
                d--;
                A[c][d]++;
                temp++;

            }
            else if(e==2&&(c+1<b&&d-1>=0)){ //左下
                c++;
```

```
51                  d--;
52                  A[c][d]++;
53                  temp++;
54
55              }
56              else if(e==3&&c+1<b){   //下
57                  c++;
58                  A[c][d]++;
59                  temp++;
60
61              }
62              else if(e==4&&(c+1<b&&d+1<a)){  //右下
63                  c++;
64                  d++;
65                  A[c][d]++;
66                  temp++;
67
68              }
69              else if(e==5&&d+1<a){   //右
70                  d++;
71                  A[c][d]++;
72                  temp++;
73
74              }
75              else if(e==6&&(c-1>=0&&d+1<a)){ //右上
```

```
76                  c--;
77                  d++;
78                  A[c][d]++;
79                  temp++;
80
81              }
82              else if(e==7&&c-1>=0){  //上
83                  c--;
84                  A[c][d]++;
85                  temp++;
86
87              }
88
89
90              for(i=0;i<a;i++){   //記總數
91                  for(j=0;j<b;j++){
92                      if(A[i][j]!=0){
93                          con++;
94                      }
95                  }
96              }
97              //printf("%d %d\n",con,a*b);
98
99              if(con==a*b){
100                 printf("總步數: %d\n",temp);
```

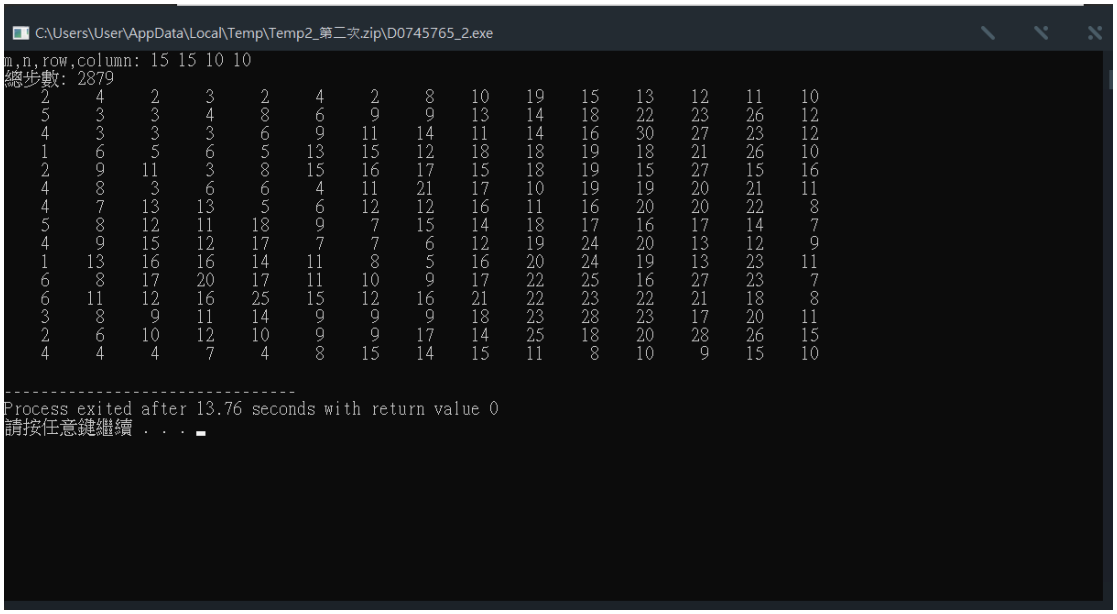
```
                    for(i=0;i<a;i++){
                        for(j=0;j<b;j++){
                            printf("%5d ",A[i][j]);
                        }
                        printf("\n");
                    }
                    break;
                }
                con=0;
                if(temp>50000){
                    printf("超過上限\n");
                    break;
                }
            }

        }

        return 0;
}
```

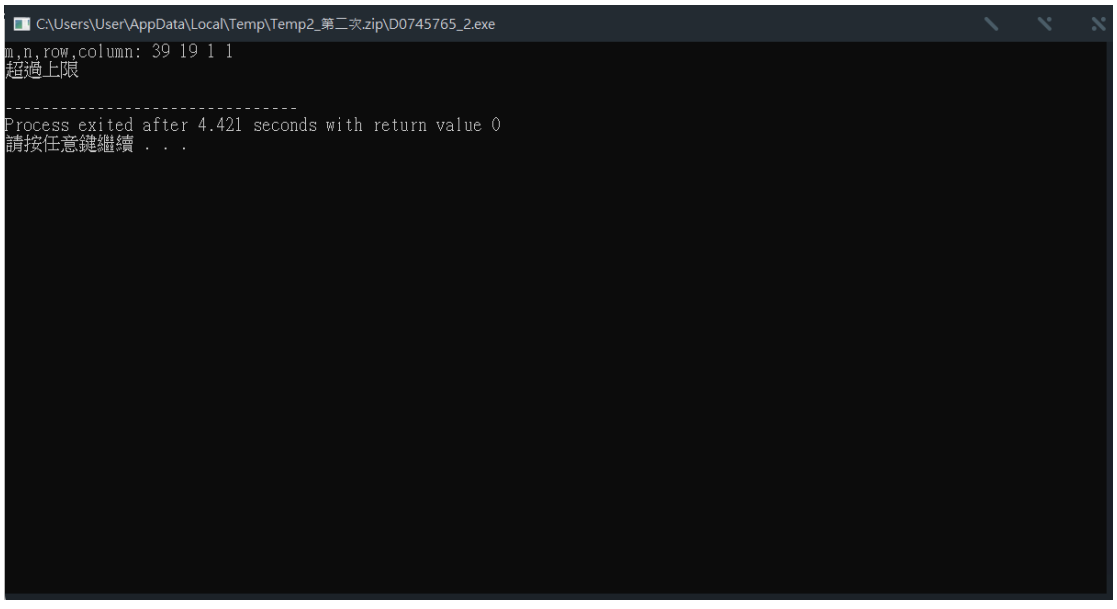
作業一輸出結果


C:\Users\User\AppData\Local\Temp\Temp3_第二次.zip\D0745765_1.exe
輸入多項式A:
輸入係數和次方: 2 2
輸入係數和次方: 2 1
輸入係數和次方: 2 0
輸入係數和次方: 0 0
輸入多項式B:
輸入係數和次方: 2 2
輸入係數和次方: 2 1
輸入係數和次方: 2 0
輸入係數和次方: 0 0
相乘結果:4x^4+8x^3+12x^2+8x^1+4

Process exited after 17.19 seconds with return value 0
請按任意鍵繼續 . . .

作業二輸出結果


C:\Users\User\AppData\Local\Temp\Temp2_第二次.zip\D0745765_2.exe
m,n,row,column: 15 15 10 10
總步數: 2879
2 4 2 3 2 4 2 8 10 19 15 13 12 11 10
5 3 3 4 8 6 9 9 13 14 18 22 23 26 12
4 3 3 3 6 9 11 14 11 14 16 30 27 23 12
1 6 5 6 5 13 15 12 18 18 19 18 21 26 10
2 9 11 3 8 15 16 17 15 18 19 15 27 15 16
4 8 3 6 6 4 11 21 17 10 19 19 20 21 11
4 7 13 13 5 6 12 12 16 11 16 20 20 22 8
5 8 12 11 18 9 7 15 14 18 17 16 17 14 7
4 9 15 12 17 7 7 6 12 19 24 20 13 12 9
1 13 16 16 14 11 8 5 16 20 24 19 13 23 11
6 8 17 20 17 11 10 9 17 22 25 16 27 23 7
6 11 12 16 25 15 12 16 21 22 23 22 21 18 8
3 8 9 11 14 9 9 9 18 23 28 23 17 20 11
2 6 10 12 10 9 9 17 14 25 18 20 28 26 15
4 4 4 7 4 8 15 14 15 11 8 10 9 15 10
Process exited after 13.76 seconds with return value 0
請按任意鍵繼續 . . .


C:\Users\User\AppData\Local\Temp\Temp2_第二次.zip\D0745765_2.exe
m,n,row,column: 39 19 1 1
超過上限
Process exited after 4.421 seconds with return value 0
請按任意鍵繼續 . . .

作業一時間複雜度

程式中有兩個時間複雜度明顯較高的演算法：

設多項式 A 大小為 n，多項式 B 大小為 m

1. 相乘計算時，巢狀迴圈所帶來的時間複雜度為 O(n*m)。

2. 進行泡沫排序法時，最壞情況為「全部的項次都無法進 行相加合併」，也就是說在「結果多項式」中會有 n*m 項，並且在最壞情況下為由小到大排列，因此在排序的 部分會有最壞時間複雜度 O((n*m)^2) = O(n^2*m^2)。

上述的時間複雜度相加： O(n*m) + O(n^2*m^2) = O(n^2*m^2)

因此，我們可以得到，此程式的時間複雜度為： O(n^2*m^2)

作業二時間複雜度

設使用者輸入的地圖大小長寬為 n、m 有三個時間複雜度相對較高的地方：

1. 生成地圖邊界的迴圈 => O((n+2)*(m+2))

2. 生成地圖可走區域迴圈 => O(n*m)

3. 列印出完整地圖結果 => O(n*m) 由上面三個複雜度相加，我們可以得到：

   O((n+2)*(m+2)) + O(n*m) + O(n*m) = O(n*m)

因此，本程式的時間複雜度為： O(n*m)

心得

在一開始做多項式乘法時我原本以為很簡單，後來才發現 要用一個陣列做完就變得非常複雜，因為在記數時很容易 亂掉，也因為這樣浪費了許多時間在想如何記數，以及如 何相乘排序，所以才會遲交，因為很多東丟還沒有釐清，整個超困難的。但在蟑螂走迷宮那題就相對容易多了，因 為那題想法蠻直觀的，在解決上也沒遇到甚麼困難，唯一 遇到的麻煩是，我把 d++打成 d=d++時就會跑爆，我覺得這 兩個是一樣的怎麼會有差，後來聽老師講解後才知道可能 是先後順序出了問題。也因此正確地完成了程式。