嵌入式程式設計

1071 電機系

期中考報告

點到直線最小垂直距離和
與 Linear Regression 方法之程式

電機三 B

王靖嫻

B1055157

蕭培墉老師

目錄

一、 題目解析

二、 解題原理與數學

三、 演算法與流程圖

四、 其他設計文件

五、 心得與討論

六、 原始程式列印

五、心得與討論

六、原始程式列印

- Headerfile.h

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

typedef struct{
    int* point_x;
    int* point_y;
}Point;
typedef struct{
    float Line1_m;
    float SumDis1;
}Line1;
typedef struct{
    float Line2_m;
    float Line2_b;
    float SumDis2;
}Line2;
typedef struct{
    Point *Point_ptr;
    Line1 *Line1_ptr;
    Line2 *Line2_ptr;
}Datastruct;
void ReadPoints(char IPtxtfname[],char OPtxtfname[],Point *Point_ptr);
void Line1Function(Point *Point_ptr,Line1 *Line1_ptr);
void Line2Function(Point *Point_ptr,Line2 *Line2_ptr);
void WriteResult(char OPtxtfname[],Line1 *Line1_ptr,Line2 *Line2_ptr);
```

- main.c

```c
#include <stdio.h>
#include <stdlib.h>
#include "Headerfile.h"
int main() {
    char IPtxt_Filename[350]="InputPoint.txt";
    char OPtxt_Filename[350];
    Datastruct *Datastruct_DS=(Datastruct*)malloc(sizeof(Datastruct));
    Datastruct_DS->Point_ptr=(Point*)malloc(sizeof(Point));
    Datastruct_DS->Line1_ptr=(Line1*)malloc(sizeof(Line1));
    Datastruct_DS->Line2_ptr=(Line2*)malloc(sizeof(Line2));

    ReadPoints(IPtxt_Filename,OPtxt_Filename,Datastruct_DS->Point_ptr);
    printf("--輸入座標點文字檔: %s\n",IPtxt_Filename) ;
    printf("--輸出運算結果文字檔: %s\n",OPtxt_Filename);
    printf("--輸入50個座標點:\n");
    int i;
    for(i=0;i<50;i++){
        printf("(%3d ,%4d ) ",Datastruct_DS->Point_ptr->point_x[i],
            Datastruct_DS->Point_ptr->point_y[i]);
        if(i%5==4&&i!=0)
            printf("\n");
    }

    Line1Function(Datastruct_DS->Point_ptr,Datastruct_DS->Line1_ptr);
    printf("Line1: y=%.1fx\n",Datastruct_DS->Line1_ptr->Line1_m);
    printf("SumDis1 = %.2f\n",Datastruct_DS->Line1_ptr->SumDis1);
    Line2Function(Datastruct_DS->Point_ptr,Datastruct_DS->Line2_ptr);
    printf("Line2: y=%.1fx+%.1f\n",Datastruct_DS->Line2_ptr->Line2_m,
        Datastruct_DS->Line2_ptr->Line2_b);
    printf("SumDis2 = %.2f\n",Datastruct_DS->Line2_ptr->SumDis2);

    WriteResult(OPtxt_Filename,Datastruct_DS->Line1_ptr,
        Datastruct_DS->Line2_ptr);
```

```
    free(Datastruct_DS->Point_ptr);
    free(Datastruct_DS->Line1_ptr);
    free(Datastruct_DS->Line2_ptr);
    free(Datastruct_DS);
    return 0;
}
```

- ReadPoints.c

```
#include <stdio.h>
#include <stdlib.h>
#include "Headerfile.h"
void ReadPoints(char IPtxtfname[],char OPtxtfname[],Point *Point_ptr)
 {
    FILE *fpoint;
    fpoint=fopen(IPtxtfname,"r");
    while(fpoint==NULL){
        printf("FAULT\n");
        printf("Input File Name:\n");
        scanf("%s\n", &IPtxtfname);
        fpoint = fopen(IPtxtfname, "r");
    }
    char buffer[350];
    fgets(buffer,350,fpoint);
    fscanf(fpoint,"%s\n",OPtxtfname);
    fgets(buffer,350,fpoint);
    Point_ptr->point_x=(int*)malloc(sizeof(int)*50);
    Point_ptr->point_y=(int*)malloc(sizeof(int)*50);
    int *X=(int*)Point_ptr->point_x;
    int *Y=(int*)Point_ptr->point_y;
```

```
    int i;
    for(i=0;i<50;i++)
        fscanf(fpoint,"(%d,%d)\n",&X[i],&Y[i]);
    fclose(fpoint);
}
```

- Line1Function.c

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "Headerfile.h"
void Line1Function(Point *Point_ptr,Line1 *Line1_ptr)
{
    float d,m;
    float Total_d;
    float i;
    int j;
    int *X=(int*)Point_ptr->point_x;
    int *Y=(int*)Point_ptr->point_y;
    Line1_ptr->SumDis1=0;
```

```
    for(i=1;i<=20;i++){
            Total_d=0;
            m=pow(2,i);
        for(j=0;j<50;j++){
            d=fabs(-m*(float)X[j]+(float)Y[j])/pow(pow(m,2.0)+1,0.5);
            Total_d+=d;
            }

        if((Line1_ptr->SumDis1)==0||Total_d<(Line1_ptr->SumDis1)){
            Line1_ptr->SumDis1=Total_d;
            Line1_ptr->Line1_m=m;
            }
    }
}
```

- Line2Function.c

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "Headerfile.h"
void Line2Function(Point *Point_ptr,Line2 *Line2_ptr)
{
    float Diff_Xi_Xavg,Diff_Yi_Yavg;
    float Sum_MultiDxDy=0,Sum_DxSqua=0;
    float Xavg,Yavg,Sum_X=0,Sum_Y=0;
    int i;
    float temp;
    int *X=(int*)Point_ptr->point_x;
    int *Y=(int*)Point_ptr->point_y;

    for(i=0;i<50;i++){
        Sum_X+=(float)X[i];
        Sum_Y+=(float)Y[i];
    }
    Xavg=Sum_X/50;
    Yavg=Sum_Y/50;

    for(i=0;i<50;i++){
        Diff_Xi_Xavg=(float)X[i]-Xavg;
        Diff_Yi_Yavg=(float)Y[i]-Yavg;
        temp=Diff_Xi_Xavg*Diff_Yi_Yavg;
        Sum_MultiDxDy+=temp;
        temp=pow(Diff_Xi_Xavg,2);
        Sum_DxSqua+=temp;
    }
    Line2_ptr->Line2_m= Sum_MultiDxDy/Sum_DxSqua;
    Line2_ptr->Line2_b=Yavg-(Line2_ptr->Line2_m)*Xavg;

    float y_line;
    Line2_ptr->SumDis2=0;
    for(i=0;i<50;i++){
        y_line=(Line2_ptr->Line2_m)*(float)X[i]+(Line2_ptr->Line2_b);
        temp=fabs((float)Y[i]-y_line);
        Line2_ptr->SumDis2+=temp;
    }
}
```

■ WriteResult.c

```c
#include <stdio.h>
#include <stdlib.h>
#include "Headerfile.h"
void WriteResult(char OPtxtfname[],Line1 *Line1_ptr,Line2 *Line2_ptr)
{
    FILE *fwrite;
    fwrite=fopen(OPtxtfname,"w");
    while(fwrite==NULL){
        printf("FAULT\n");
        printf("Output File Name:\n");
        scanf("%s\n", &OPtxtfname);
        fwrite = fopen(OPtxtfname, "w");
    }
    float Diff;
    fprintf(fwrite,"==================輸出結果==================\n");
    Diff=(Line2_ptr->SumDis2)-(Line1_ptr->SumDis1);
    fprintf(fwrite,"Line1: y = %.2fx\n",Line1_ptr->Line1_m);
    fprintf(fwrite,"SumDis1 = %.2f\n",Line1_ptr->SumDis1);
    fprintf(fwrite,"Line2: y = %.2fx + %.2f\n",
    Line2_ptr->Line2_m,Line2_ptr->Line2_b);
    fprintf(fwrite,"SumDis2 = %.2f\n",Line2_ptr->SumDis2);
    fprintf(fwrite,"Diff = SumDis2 - SumDis1\n     = %.2f - %.2f = %.2f",
    Line2_ptr->SumDis2,Line1_ptr->SumDis1,Diff);
    printf("Diff = SumDis2 - SumDis1\n     = %.2f - %.2f = %.2f",
    Line2_ptr->SumDis2,Line1_ptr->SumDis1,Diff);

    fclose(fwrite);
}
```

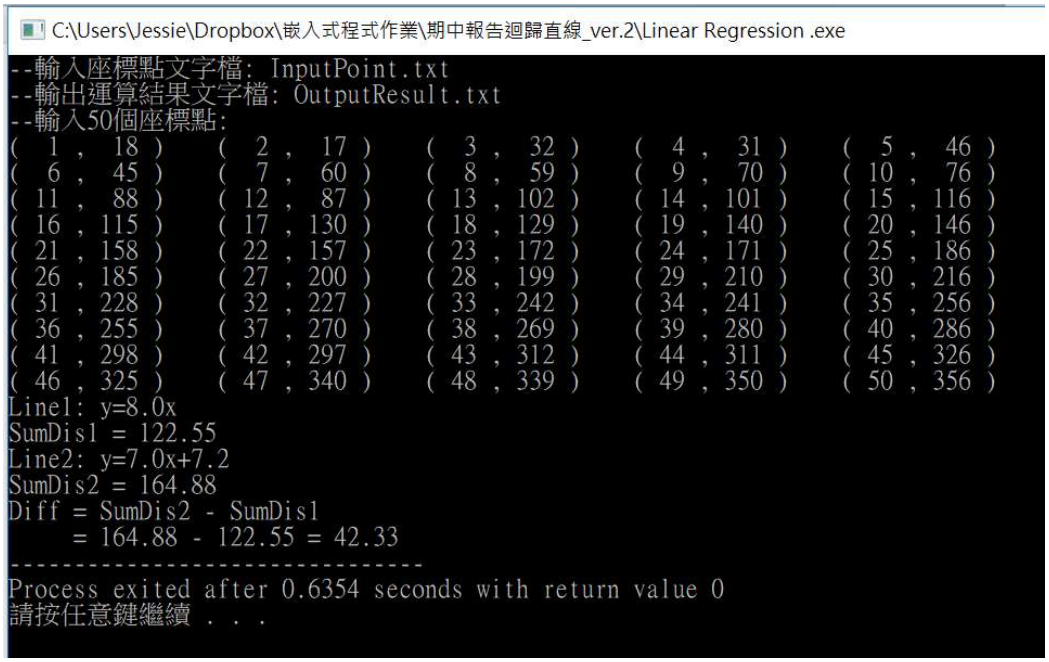
□ 輸入文字檔 InputPoint.txt

InputPoint.txt - 記事本

檔案(F) 編輯(E) 格式(O) 檢視

```
--輸出運算結果文字檔
OutputResult.txt
--輸入50個座標點:
(1,18)    (11,88)   (21,158)  (31,228)  (41,298)
(2,17)    (12,87)   (22,157)  (32,227)  (42,297)
(3,32)    (13,102)  (23,172)  (33,242)  (43,312)
(4,31)    (14,101)  (24,171)  (34,241)  (44,311)
(5,46)    (15,116)  (25,186)  (35,256)  (45,326)
(6,45)    (16,115)  (26,185)  (36,255)  (46,325)
(7,60)    (17,130)  (27,200)  (37,270)  (47,340)
(8,59)    (18,129)  (28,199)  (38,269)  (48,339)
(9,70)    (19,140)  (29,210)  (39,280)  (49,350)
(10,76)   (20,146)  (30,216)  (40,286)  (50,356)
```

- 執行結果



- 輸出文字檔 OutputResult.txt