**《软件工程》实验报告二**

姓名：吴怀文

学号：U201516993

班级：软工1501班

**Introduction：**

在一个边界为[-1, 1]的盒子里，给定m个半径r和位置mu可变的气球（它们不能重叠），找到最优的r和mu使得r^2之和最大。

**Algorithm：** 首先确定圆心遍历的步长（例如0.01），半径增加的步长（0.00001）和气球个数m。且将给定点视作半径为0的圆事先列入所求圆的链表中。这样在题目1的基础上几乎不需做出任何改变，便可以自动在找寻最优解的时候回避掉这些点。依次确定第一个半径最大的圆，而后在剩下的空间中确定第二个半径最大的圆，直至所有m个圆填入区域中，此时m个圆的半径平方之和最大。

**Structure and Test**

typedef struct ball

{

double x; //圆心x坐标

double y; //圆心y坐标

double r; //圆半径

}Ball;

//用来储存已经放置的气球

typedef struct ballList

{

struct ballList \* next;

Ball ball;

}BallList;

**void putPoint(double x, double y) //将给定的点当作球，方便利用已有代码**

**{**

**Ball ball = {x, y, 0};**

**insert(ball);**

**}**

void putBall() //关键函数，找到所剩空间里最大的球

{

Ball ball = {-1 + step, -1 + step, 0};

Ball maxBall = ball;

while (ball.x < 1)

{

ball.x += step;

ball.y = -1 + step;

while(ball.y < 1)

{

ball.y += step;

ball.r = 0;

double rstep = 0.1;

while(rstep > 0.00001)

{

if(ball.r > maxBall.r)

{

maxBall = ball;

}

ball.r += rstep;

if(!judge(ball))

{

ball.r -= rstep;

rstep /= 10;

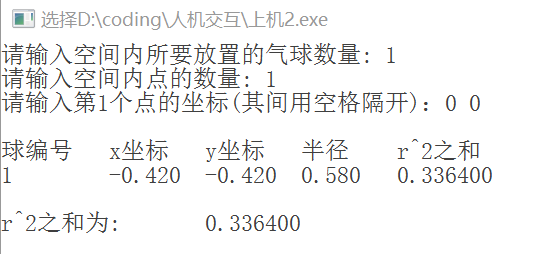
}

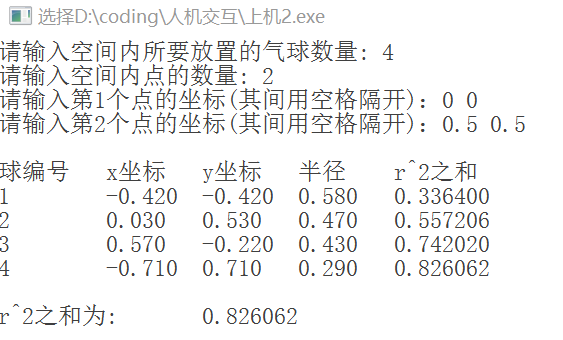
}

}

}

**Result**





**Conclusion** 本题在第一题的基础上只需将需要排除的点视为半径为0的球即可，进而可利用1的代码中排除不同球之间相互重叠的方式，在寻找解答时将这些点排除掉。