《算法分析与设计》

实 验 报 告

学 号

姓 名

年 级 2020

专 业 软件工程

院 系 计算机与人工智能学院

二0二二年三月

目 录

实验一 求两个圆相交部分的面积······································1

# 实验一 求两个圆相交部分的面积

1. 实验目的

|  |
| --- |
| 1. 掌握算法的自然语言描述法，流程图绘制方法以及伪代码描述方法。 2. 掌握算法的程序实现方法。 3. 掌握程序的调试方法及测试方法。 |

1. 实验任务

|  |
| --- |
| 已知两个圆的圆心坐标以及半径，求两个圆相交部分的面积，并精确到小数点后3位。要求：   1. 用自然语言描述法描述上述问题的求解算法。 2. 绘制算法程序流程图。 3. 用伪代码描述上述问题的求解算法。 4. 根据相关的描述方法，编写程序。 5. 设计相应的测试数据，上机验证程序执行结果是否正确。 6. 撰写相应的实验报告，实验报告内容包括：实验目的、实验任务、实 验环境、实验步骤、实验结果及其分析以及实验总结等部分内容。 |

1. 实验环境

|  |
| --- |
| * 1. 硬件环境  1. 计算机：拯救者R7000P 2020 2. CPU: AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz 3. RAM：16GB    1. 软件环境 4. 操作系统：Windows11家庭中文版 5. 开发工具：Visual Studio Code |

1. 实验步骤及结果

|  |
| --- |
| * 1. 实验预习      1. 该问题的自然语言描述法   输入：圆o1的圆心坐标（a，b）,半径为c,圆o2的圆心坐标为(x,y),半径为z。  输出：两圆相交部分的面积ans（双精度数）。  算法描述：   1. 设双精度数l为圆心距，l=; 2. 如果两圆相离、外切或至少一圆半径为0时，所求面积为0； 3. 如果两圆内切或内含：在z>c时,则z=c，打印面积z\*z\*PI； 4. 如果两圆相交，由余弦定理分别求出公共弦在圆o1、o2中对应的圆心角的一半d,e；圆o1中扇形面积m=d\*c\*c,圆o2中扇形面积n=e\*z\*z；圆o1中扇形对应三角形面积q=c\*c\*sin（b）\*cos（b），圆o2中扇形对应三角形面积w=z\*z\*sin（y）\*sin（y）；q+w为图中四边形面积，两扇形面积之和与四边形面积之差即为所求面积，即ans=(m+n)-(q+w)。    * 1. 程序流程描述   如图1所示。    图1 程序流程图   * + 1. 伪代码描述   输入：a,b,c,x,y,z;  输出：ans;  算法：  lßsqrt((a-x)\*(a-x)+(b-y)\*(b-y));  if l>=c+z||!c||!z then  print ansß0.000;  else if l<=fabs(z-c) then  if z>c then  z=c;  print ansßz\*z\*PI;  else  dßacos((l\*l+c\*c-z\*z)/(2\*l\*c));  eßacos((l\*l+z\*z-c\*c)/(2\*l\*c));  mßd\*c\*c;  nße\*z\*z;  qßc\*c\*sin（b）\*cos（b）  wßz\*z\*sin（y）\*sin（y）  ansß (m+n)-(q+w);  print ans;  end   * + 1. 程序代码   #include <cmath>  #include <cstdio>  double q, w, m, n, a, b, c, d, x, y, z, e, PI, ans;  int main()  {      PI = 2 \* asin(1.0);      scanf("%lf%lf%lf", &a, &b, &c); *//横坐标，纵坐标，半径*      scanf("%lf%lf%lf", &x, &y, &z); *//横坐标，纵坐标，半径*      double l = sqrt((a - x) \* (a - x) + (b - y) \* (b - y)); *//计算圆心距*      if (l >= c + z || !c || !z) *//如果两圆相离、外切或至少一圆半径为0时，那么所求面积为0*          ans = 0;      else if (l <= fabs(z - c)) *//如果两圆内切或内含，那么所求面积为小圆面积*      {          if (z > c)              z = c;          ans = z \* z \* PI;      }      else *//如果两圆相交，*      {          d = acos((l \* l + c \* c - z \* z) / (2 \* l \* c)); *//由余弦定理求出公共弦在圆o1中对应的圆心角的一半    a^2+b^2-c^2=2\*a\*b\*cos(C)*          e = acos((l \* l + z \* z - c \* c) / (2 \* l \* z)); *//由余弦定理求出公共弦在圆o2中对应的圆心角的一半*          m = d \* c \* c; *//圆o1中扇形面积*          n = e \* z \* z; *//圆o2中扇形面积*         ans = m + n - c \* l \* sin(d);      }      printf("%.3f\n", ans);      system("pause");      return 0;  }  4.2上机实验  4.2.1算法测试   1. 测试数据1 2. 2.0 3.0 4.0 5.0 6.0 3. 测试数据2   0.0 0.0 2.0 1.0 0.0 1.0   1. 测试数据3   0.0 0.0 2.0 3.0 0.0 1.0   1. 测试数据4 2. 0.0 2.0 2.5 0.0 1.0   4.2.2测试结果及其分析   1. 对应测试数据1，其程序执行过程中中间变量的值如图2所示。     图2 程序执行过程中间结果  程序执行结果如图3所示。    图3 测试数据1对应的程序执行结果  结果分析：  测试数据1对应的示意图如图4所示。图中重叠区域的面积=扇形BCD面积m+扇形ACD面积n-四边形CADB的面积S    图4 测试数据1示意图  其中:  M= acos((l \* l + c \* c - z \* z) / (2 \* l \* c))\*c\*c (1)  N= acos((l \* l + z \* z - c \* c) / (2 \* l \* z))\*z\*z (2)  S= c \* l \* sin(d) (3)  ans=23.005 (4)  从上面的计算过程可以看出，其结果与程序运行结果一致，说明针对测试数据1其程序和算法正确。  (2)针对测试数据2，其运行过程及结果如5所示。    图5 数据2的运行过程及结果  结果分析：  测试数据2对应如6所示示意图。    图6 测试数据2示意图  显然这种情况下，其相交部分的面积  ans=PI\*c\*c=3.1416\*1\*1=3.1416≈3.142 （5）  (3)针对测试数据3，其程序执行结果如图7所示。    图7 测试数据3的执行结果  结果分析：  测试数据3针对的是图8的情况。    图8 测试数据3示意图  显然，这种情况下两圆相切，相交部分面积  ans=0 （6）  (4)针对测试数据4，其执行过程如图9所示。      图9 测试数据4执行过程中变量的值  其执行结果如图10所示。    图10 测试数据4执行结果  结果分析：  测试数据4对应的示意图如图11所示。    图11 测试数据4示意图    相交部分的面积ans=扇形ACM’D面积m-+扇形BCH’D的面积n-四边形ABCD的面积S  M= acos((l \* l + c \* c - z \* z) / (2 \* l \* c))\*c\*c (7)  N= acos((l \* l + z \* z - c \* c) / (2 \* l \* z))\*z\*z (8)  S= c \* l \* sin(d) (9)  ans=m+n-s= 0.522 (10)  从上面的计算过程来看，其计算结果与程序执行结果一致，表明在该种情况下的算法和程序正确。  同时分析（1）和（4）的测试数据的计算过程可以发现，其计算的表达式其实是一样的，可以归纳成一种计算表达式。如前面的算法所述。 |
| 5.实验总结  通过本次实验，掌握了算法的自然语言描述方法、使用markdown语法绘制流程图方法、伪代码表示方法等多种方法表示算法流程。学会了如何分析算法的执行过程以及验证算法及程序的正确性。 |