

判断线段是否相交



Jayce_SYSU

关注

判断线段是否相交 原创

Jayce_SYSU 2019-09-18 18:23:57 博文文章分类: 数据结构算法

©著作权

文章标签 计算几何 文章分类 Python 编程语言 阅读数 1305



Jayce_SYSU

84

原创

12.6万

人气

0

粉丝

3

评论

0

翻译

0

转载

0

关注

1

收藏

关注

私信

数据结构算法分类的近期文章

leetcode 409: 给定字符串, 求能重组...

leetcode 5: 最长回文子串

最长公共子串

leetcode: 72. 最小编辑距离

约瑟夫环

近期评论

U盘数据恢复

我丢失过两次数据。一次是不小心格式化, 一...

ubuntu 下 virt-manager 的 ssh认证问题

你好, 我也碰到了这个问题, 但是我已经安装...

Git学习心得

临近春节, 你是不是也想参加完公司年会领完...

近期文章

1.leetcode 409: 给定字符串, 求能重组...

2.leetcode 5: 最长回文子串

3.最长公共子串

4.leetcode: 72. 最小编辑距离

5.约瑟夫环

2019年 67篇

2016年 14篇

2015年 3篇

热评好文

U盘数据恢复

七日热门

No Python at 'E:\Python\python.exe'

[Python] Python Libs

Python (01) : Python简介与Python...

python乱码,python编码,python中文编...

[Python] Format Strings in Python

【python学习-python语法】

【python】install python 2.7.9

Python: Opening Python Classes

[Python] Understand Scope in Python

[python]python元类

分类列表

```
1.  # -*- coding: utf-8 -*-
2.  # @Time      : 2019-09-18 16:55
3.  # @Author    : Jayce Wong
4.  # @ProjectName : job
5.  # @FileName   : segment_cross.py
6.  # @Blog      : http://blog.51cto.com/jayce1111
7.  # @Github    : https://github.com/SysuJayce
8.
9.  """
10. Q: 给定两个线段的坐标(也就是四个点的直角坐标系坐标), 判断这两个线段是否相交
11.
12. ### 判断线段是否相交可以利用向量的叉乘 ###
13. 假定输入为P1、P2、Q1、Q2四个点的坐标, P1P2为一条线段, Q1Q2为另一条线段
14.
15. 两条线段相交只有两种情况
16. 1. 其中一条线段的某一段点在另一条线段上:
17. 2. 两条线段形成X形。
18.
19. 首先判断这四个点是否在另一条线段上, 也就是说, 判断P1是否在线段Q1Q2上, P2是否在线段Q1Q2上...
20. 如果上述判断为真, 那么这两条线段相交。【解决了第一种情况】
21.
22. 如果没有点在另一条线段上, 那么进行叉乘判断。
23. 先固定线段Q1Q2, 然后以Q1为轴, 计算Q1P1和Q1Q2、Q1P2和Q1Q2的叉乘是否异号:
24. 然后固定线段P1P2, 然后以P1为轴, 计算P1Q1和P1P2、P1Q2和P1P2的叉乘是否异号。
25. 当上述的叉乘都异号的时候, 两条线段相交。
26. 【解决了第二种情况】
27. """
28.
29.
30. class Point:
31.     def __init__(self, x, y):
32.         self.x = x
33.         self.y = y
34.
35.     def __sub__(self, other):
36.         return Point(self.x - other.x, self.y - other.y)
37.
38.
39. class Segment:
40.     def __init__(self, point1, point2):
41.         self.point1 = point1
42.         self.point2 = point2
43.         self.x = point1.x - point2.x
44.         self.y = point1.y - point2.y
45.
46.
47.     def crossProduct(v1, v2):
48.         return v1.x * v2.y - v2.x * v1.y
49.
50.
51.     def onSegment(p, seg):
52.         """
53.         判断点p不在一条线段上, 关键在于:
54.         1. 三点是否共线
55.         2. 点p是否在线段的延长线上。
56.
57.         只要满足了三点共线, 且点p不在延长线上, 那么点p就在线段上。
58.
59.         判断三点共线可以用向量的叉乘, 三点共线即两个向量平行, 也就是叉乘结果为零向量(对应到二维就是零)
60.         当点p的横纵坐标都在线段端点之间的时候, 点p不在延长线上。
61.         :param p:
62.         :param seg:
63.         :return:
64.         """
65.         # 先确保点p不在延长线上
66.         if min(seg.point1.x, seg.point2.x) <= p.x <= max(seg.point1.x, seg.point2.x)\
67.            and min(seg.point1.y, seg.point2.y) <= p.y <= max(seg.point1.y, seg.point2.y):
68.             # 然后确保这三个点形成的向量两两平行, 这里只要这三个向量中任意两个平行, 第三个一定也平行
69.             if crossProduct(p - seg.point1, p - seg.point2) == 0:
70.                 return True
71.             else:
72.                 return False
73.         else:
74.             return False
```



赞



收藏



评论



分享



返回顶部

判断线段是否相交

```
78.     p1p2 = Segment(p1, p2)
79.     q1q2 = Segment(q1, q2)
80.
81.     p1q1 = Segment(p1, q1)
82.     p1q2 = Segment(p1, q2)
83.
84.     q1p1 = Segment(q1, p1)
85.     q1p2 = Segment(q1, p2)
86.
87.     # 判断是否存在端点位于另一条线段上，是的话则两条线段相交
88.     if any([onSegment(p1, q1q2), onSegment(p2, q1q2),
89.             onSegment(q1, p1p2), onSegment(q2, p1p2)]):
90.         return True
91.
92.     # 否则固定线段P1P2，判断Q1和Q2是否在P1P2的两侧(计算叉乘)
93.     # 然后固定线段Q1Q2，判断P1和P2是否在Q1Q2的两侧
94.     # 如果上面的判断均为真，那么这两条线段形成一个X
95.     return (crossProduct(p1p2, p1q1) * crossProduct(p1p2, p1q2) < 0)\
96.           and (crossProduct(q1q2, q1p1) * crossProduct(q1q2, q1p2) < 0)
97.
98.
99. def main():
100.     p1 = Point(0, 0)
101.     p2 = Point(2, 2)
102.     q1 = Point(1, 1)
103.     q2 = Point(0, 2)
104.     if isCross(p1, p2, q1, q2):
105.         print('Yes')
106.     else:
107.         print('No')
108.
109.
110. if __name__ == '__main__':
111.     main()
112.
```

相关标签

全部

java 判断线段相交

java 判断线程是否等

待

java 判断线程是否结束

java 求线段相

交

java 线段相交

java两线段是否相交

java判断两条线段是否相交

java判断线程是

否启动

java判断线程是否运行

java判断

线程是否阻塞

赞

收藏

评论

分享

赞

收藏


评论

分享

举报

上一篇：剑指offer：在排序数组中查找数字

下一篇：比较两个连分数的大小



提问和评论都可以，用心的回复会被更多人看到

评论

相关文章

判断线段和矩形是否相交

判断线段和矩形是否相交 package { import flash.display.Sprite; import flash.events.MouseEvent; import flash.text.TextField; [S...

计算几何-判断线段是否相交

计算几何-判断线段相交 判断两线段是否相交：快速排斥 跨立实验（这两个词也是我看博客的时候看到的，觉得...



判断线段相交

规范相交：两条线段恰有一个不是端点的公共点。即如果一条线段的一个端点恰在另一条线段上则不视为相交；如...



判断线段是否与圆相交模板

判断线段和圆是否相交 判断圆和线段相交，分两种情况： 1. 如图A所示，当圆心与线段的距离大于圆的半径时，...



平面中判断线段与矩形是否相交

简单介绍了平面中判断线段与矩形是否相交的算法。 目录 1. 原理 2. 实现 3. 参考 1. 原理 这个问题的算法思路挺简单的。分成...

判断两线段是否相交



返回顶部

判断线段是否相交

C - Segments POJ - 3304 (判断线段相交)

题目链接: <https://vjudge.net/contest/276358#problem/C> 题目大意: 给你n条线段, 问你是否存在一条线段使得所有的线段在这...

POJ 1066 - Treasure Hunt - [枚举+判断线段相交]

题目链接: <http://poj.org/problem?id=1066>Time Limit: 1000MS Memory Limit: 10000KDescriptionArcheologists fr...



POJ 2653 Pick-up sticks (判断线段相交)

Pick-up sticks Time Limit: 3000MS Memory Limit: 65536K Total Submissions: 7699 Accepted: 2843 Descriptio...



hdu1558--并查集+判断线段相交

简单的计算几何题, 判断两线段是否相交。将相交的两线段使用并查集归到一类中。查询时输出线段对应集合中元素的个数。 #...

线段是否相交

<http://www.acm.cs.ecnu.edu.cn/problem.php?problemid=1062> 判断线段是否相交 1. 线段组成的矩阵是否相交 2.若不相交, 则两...

【BZOJ 1033】[ZJOI2008]杀蚂蚁antbuster(判断线段是否和圆相交)

【题目链接】: <http://www.lydsy.com/JudgeOnline/problem.php?id=1033> 【题意】 【题解】... 模拟题。要用到计算几何知...

判断两条线段是否相交

Hdoj1086 题意, 给你n条线段然后让你判断有几个交点。 所以思路就是判断两条线段是否相交, 然后for一下就...



POJ 2653 - Pick-up sticks - [枚举+判断线段相交]

题目链接: <http://poj.org/problem?id=2653>Time Limit: 3000MS Memory Limit: 65536KDescriptionStan has n stick...



判断两线段相交

```
#include <iostream> #include <cstdio> #include <algorithm> #include <string.h> #include <stdlib.h> using namespace std; int n;...
```

两线段是否相交模板

```
1 #include<bits/stdc++.h> 2 using namespace std; 3 struct point 4 { 5 int x,y; 6 }p1,p2,p3,p4; 7 int cross(const poi...
```



HDU 1086 You can Solve a Geometry Problem too (水题, 判断线段相交)

You can Solve a Geometry Problem too Time Limit: 2000/1000 MS (Java/Others) Memory Limit: 65536/32768 K (Java/Others...

判断两矩形是否相交

```
1 //判断矩形是否相交 2 bool FMath::IsRectIntersect(const FRect& rect1, const FRect& rect2) 3 { 4 bool bResult =...
```



You can Solve a Geometry Problem too (判断两线段是否相交)

You can Solve a Geometry Problem too Time Limit: 2000/1000 MS (Java/Others) Memory Limit: 65536/32768 K (Java/Others...

- 赞
- 收藏
- 评论
- 分享

友情链接

- 51CTO鸿蒙社区
- 51CTO学堂
- 51CTO

关于我们

- 官方微博
- 意见反馈
- 了解我们
- 热门文章
- 在线客服
- 网站地图