

简单多边形检索

1 问题描述

感觉教学网上对于问题的描述并不是很清晰，所以首先确定需要解决的问题：

1. 输入一组线段，找出所有的简单多边形
2. 给出所有简单多边形的坐标点序列、周长、面积及外包矩形等信息，并判断是凹多边形还是凸多边形
3. 尝试：给出多边形之间的拓扑关系

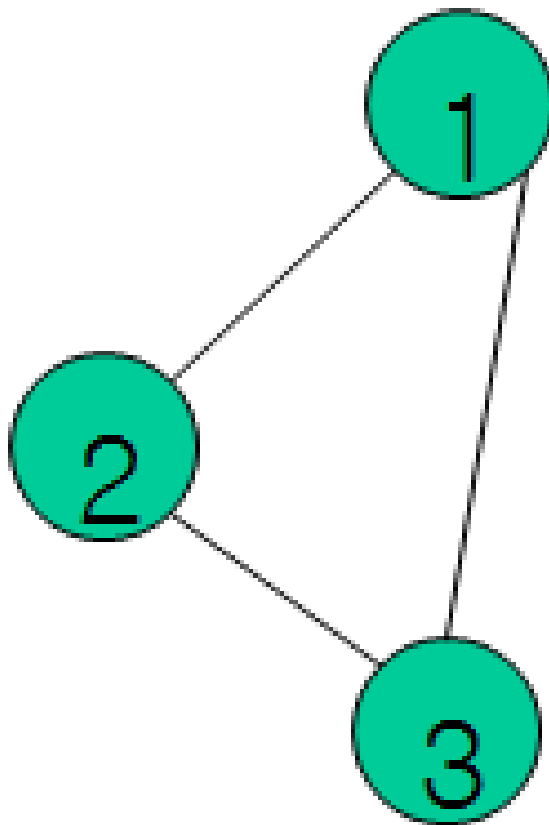
文件的输入是一组线段，相当于是以 直接存边 的方式提供了一个无向图。

解题思路为：首先找出图中所有的无向图中的环，再判断每一个环是否为简单多边形，最后建立简单多边形之间的拓扑关系。

2 DFS 找出无向图中所有的环

深度遍历(DFS)无向图，如果在遍历的过程中，发现某个节点有一条边指向已经访问过的节点，并且这个已访问的节点不是当前节点的父节点（这里的父节点表示 DFS 遍历顺序中的父节点），则表示存在一个环。

但是我们不能仅仅使用一个 bool 数组来标志节点是否访问过。如下图：



从节点 1 开始遍历，接着遍历 2，接着遍历 3，然后发现节点 3 有一条边指向遍历过的节点 1，则存在环。但是完成一次深度遍历回到节点 1 时，它的另一条边指向已访问过的节点 3，又把这个环重复计算了一次。

解决方法是把每个节点分为三种状态，白，灰，黑。开始时所有节点都是白色，当开始访问某个节点时该节点变为灰色，当该节点的所有邻接点都访问完，该节点颜色变为黑色。则算法规则为：如果遍历的过程中发现某个节点有一条边指向颜色为灰的节点，那么存在环。那么上面的例子中，回溯到节点1时，虽然有一条边指向已经访问过的节点3，但是节点3已经是黑色了，所以环不会被重复计算。

在使用 DFS 算法时，无向图最好使用邻接矩阵储存，所以还需要把文件输入的 直接存边 转换为 节点表+邻接矩阵。

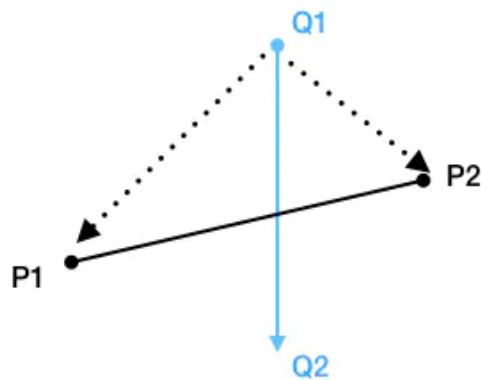
3 简单多边形判定及属性

3.1 暴力解法

简单多边形，即边只能在顶点处相交。

考虑到数据量并不大，这里使用暴力解法：一个 n 边形有 n 条边，每条边只需要和其他的 $n-3$ 条不相邻边判断是否相交即可。

3.2 两条边规范相交



$$\begin{aligned} A &= \vec{P_1P_2} \times \vec{P_1Q_1} \\ B &= \vec{P_1P_2} \times \vec{P_1Q_2} \\ C &= \vec{Q_1Q_2} \times \vec{Q_1P_1} \\ D &= \vec{Q_1Q_2} \times \vec{Q_1P_2} \end{aligned}$$

当 $A \times B < 0$ && $C \times D < 0$ 时，两条线段规范相交。

3.3 凹凸多边形

从第一条边开始，使用相邻边的叉积来测试凸凹性。凸多边形的所有向量叉积均同号。因此，如果某些叉积取正值而另一些为负值，可确定其为凹多边形。

4 数据结构