

平面图 解题报告

山东省胜利第一中学 王子昱

2013 年 4 月 4 日

Contents

1	题目大意	2
2	数据规模	2
3	算法分析	2
3.1	算法1	2
3.1.1	算法1-1	2
3.1.2	算法1-2	2
3.1.3	算法1-3	3
3.2	算法2	3
3.3	算法3	4
4	参考文献	5

1 题目大意

给定一个边带权的平面图，包含 N 个点和 M 条边，并给定顶点坐标。

Q 组询问，每次给定两个点，询问两点所在区域在对偶图中对应的顶点间瓶颈最短路。其中瓶颈最短路的定义是，两点间最大边权最小的路径。

2 数据规模

对于50%的数据，给定的图是网格图。其中20%的数据， $N \leq 2000, M \leq 3000, q \leq 1000$ 。

对于另外20%的数据， $N, M, Q \leq 2000$ ，边权均为1。其中10%的数据，每个区域是凸多边形。

所有数据满足 $N, M, Q \leq 10^6$ 。

3 算法分析

我们将问题分为三部分：找出每个点在平面图中对应的区域（点定位问题），构建对偶图，以及求瓶颈最短路。

3.1 算法1

对于前50%的数据，由于给定的图是网格图，问题的前两部分可以方便地得到解决。

接下来，我们只需要考虑问题的第三部分。

3.1.1 算法1-1

对其中20%的数据，对每次询问，我们二分答案 w ，在对偶图中删去权值超过 w 的边，使用BFS或者并查集确定两点是否连通。

时间复杂度 $O(N + Q(N + M)\log M)$ 。期望得分20分。

3.1.2 算法1-2

首先我们给出瓶颈最短路的一个性质。

引理 在图的最小生成树上连接两点的路径一定是图中这两点间的瓶颈最短路。

证明 假设结论对点对 (u, v) 不成立。设 (u, v) 间的瓶颈最短路权值为 W ，于是最小生成树上连接 uv 的路径中，存在一条权值大于 W 的边。删去这条边后生成树分为两部分，易知瓶颈最短路存在一条边连接这两部分。加入该边，我们得到了一棵权值和更小的生成树，这与原树为最小生成树矛盾。证毕。

由这一性质，我们可以求出对偶图的最小生成树，每次询问转化为树上的两点间权值最大的边。这一问题可以用倍增法或者Link-Cut Tree解决，时间复杂度 $O(\log N)$ 。

总时间复杂度 $O((N + M)\log N)$ 。期望得分50分。

3.1.3 算法1-3

对瓶颈最短路问题，我们还有其他的做法。将边按权值排序，维护权值不超过 W 的边构成的图的连通性。这一问题可以使用块状链表维护的持久化并查集解决。

使用路径压缩和按秩合并实现的并查集，每次操作对前驱数组的修改次数是均摊 $O(\alpha(N))$ 的。块状链表中，每次修改的时空代价均为 $O(\sqrt{N})$ ，因此，预处理的复杂度是 $O(M\alpha(N)\sqrt{N})$ 。询问时二分答案，在对应的并查集中 $O(\alpha(N))$ 验证。算法的时间复杂度 $O((M\sqrt{N} + Q\log M)\alpha(N))$ ，空间复杂度 $O(M\alpha(N)\sqrt{N})$ 。

3.2 算法2

对于20%的数据， $Q, N, M \leq 2000$ ，因此每次询问我们可以在 $O(N + M)$ 的时间内完成点定位。只需要枚举每个区域，判断点是否在对应的多边形之内即可。

在凸多边形中，一点在多边形中等价于该点与每条边的两个端点构成的矢量的叉积的符号恒定；在一般多边形中，可以用射线法判定。具体细节可以参考《算法艺术与信息学竞赛》中第三章的内容。

构建对偶图时，我们使用如下算法：将每个点出发的边按极角序排序，从每个点出发，每次选择未被选择过的极角序最小的边走下去，直到返回出发时的边，此时我们就确定了一个区域。这一算法的复杂度是 $O(N + M\log M)$ 的。

问题的第三部分，求瓶颈最短路，可以使用算法1-1中提到的方法。

总复杂度 $O(M\log M + N + Q(N + M)\log M)$ 。期望得分20分。与算法1-2或1-3结合，可以得到70分。

3.3 算法3

在之前的讨论中，问题的第二部分和第三部分已被圆满解决。现在考虑问题的第一部分，点定位。

这一问题的传统做法是梯形剖分，但是其实现难度较大，很难在考场上完成。这一做法可以参见冬令营时钱桥的讲义。

注意到需要定位的点的坐标在开始时全部给定，所以我们考虑离线解决。用扫描线解决这一问题。按照x坐标扫描，用平衡树维护x坐标与当前值有交集的边的集合。查询时在每个点上方的最下方的一条边对应的区域即为所求。

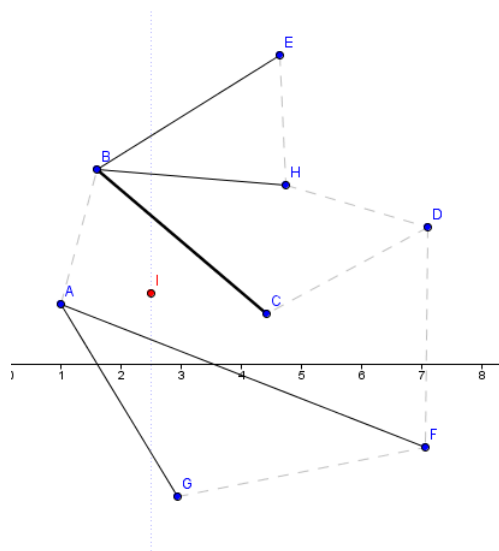


Figure 1: An example of the scanning line algorithm

如图，扫描线在虚线所在位置时，平衡树中包含的边用实线标明，查询点I时查到的边加粗表示。

至此，问题得到解决。算法的时间复杂度 $O((M + Q)\log N)$ 。期望得分100分。

注意这一算法可以用在在线询问中，只需要将扫描线算法中所用的平衡树持久化即可。这是点定位问题的一个性价比较高的解决方案。

4 参考文献

1. 刘汝佳, 黄亮, 《算法艺术与信息学竞赛》, 清华大学出版社。