Rain – Solution

试题来源

ACM/ICPC World Finals 2010 H

简要题意

给你一个n个点m条边的平面图，每个区域都是三角形，赋予每个顶点海拔h_i，就形成了一张三维地形图。现在源源不断的大雨从天而降，雨水在该区域的某些低洼除沉积下来，就形成了湖。问总共形成了多少湖以及每个湖湖面的海拔高度。

考察算法

计算几何、最短路

题解

分析题目，我们不难发现以下性质：

性质 1 每个湖至少包含一个顶点。

性质 2 每个顶点至多属于一个湖。

性质 3 水能在某个顶点上积累至海拔h的条件是：$h > h_i$且不存在任意一条从该顶点到边界顶点的路径，满足路径上的点的海拔均小于h。

性质 4 两个顶点属于同一个海拔为h的湖的条件是：水在这两个点上均能积累至海拔h，并且这两点间存在一条路径满足路径上的所有点海拔均小于h。

30%算法

从大到小枚举湖的海拔h，对于每个还未包含在某个湖中并且海拔小于h的顶点，我们检查它是否可能被包含在一个海拔为h的湖中，根据**性质 3**，我们只需从该顶点开始 BFS，只走海拔小于h的顶点，判断是否能到达边界上的某个顶点即可。接下来，根据性质 4，由满足条件的顶点组成的连通块必定属于同一个湖，于是我们统计满足条件的点组成的极大连通块的个数，即为海拔为h的湖的个数。

由于给定的是一个平面图，$m = O(n)$，因此对于每一个高度h的时间消耗是$O(n^2)$，该算法的总时间复杂度是$O(n^2h)$，能获得 30%的分数。

40%算法

注意到 30%算法中，本质不同的h只有$O(n)$种，因此该算法可以在$O(n^3)$时间内实现。可以获得 40%的分数。

100%算法

根据**性质 2** 每个顶点最多属于 1 个湖，又根据 30%算法我们知道，这个湖的海拔是满足**性质 3** 的最大的h，我们考虑如何求h。我们发现"满足**性质 3** 的最大的h"可以等价转化为"不存在任意一条从该顶点到边界顶点的路径，满足路径上的点的海拔最大值小于h，并且存在任意一条从该顶点到边界顶点的路径，满足路径上的点的海拔的最大值小于$h+1$"，也就是说

$$h = \min\left\{ \max\{h_u \mid u \in P\} \mid P\text{是该顶点到边界的一条路径}\right\}$$

这个式子实际上就是短板原理——湖的海拔取决于海拔最低的"挡板"的海拔。记第i个顶点对应的h为d_i，不难得到如下转移方程

$$d_i = \begin{cases} h_i, & i\text{为边界上的顶点} \\ \min\left\{\max\left(d_j, h_i\right) \mid i\text{与}j\text{相邻}\right\}, & i\text{不为边界上的顶点} \end{cases}$$

我们可以使用类似于 Dijkstra 算法的方式，初始时设边界上的点的d值为其海拔，其他点为inf，每次选取d值最小的点来更新周围的点，重复$n-1$次即可计算出所有d_i。

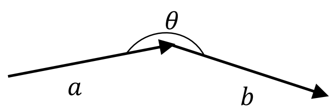
这样一来，对于某个顶点i，如果$d_i > h_i$，则表示存在一个包含i的湖，其湖面海拔为d_i。根据**性质 4**，我们从该点开始 BFS，只走海拔小于d_i的顶点，即可求出这个湖的其他顶点。我们依次确定每个顶点所属的湖，即可求出所有的湖。

求解d时，如果我们用堆来实现，那么该方法的时间复杂度即为$O(n\log n)$。实际上我们也可以对于每种海拔挂链表的方式，从小到大枚举海拔高度来实现优先队列，这种方法的时间复杂度为$O(n+h)$。

寻找边界上的顶点

无论使用上面哪个算法，我们都需要确定哪些顶点在边界上。

首先寻找区域中最左边的顶点（x最小的顶点），由其出发的倾角最大的边一定在边界上，记这条边为 S。



如上图所示，定义两条连续的边a和b（连续表示a的终点就是b的起点）夹角θ为将b绕两条边的公共点逆时针旋转至于a重合所需旋转角度。我们从 S 出发，每次寻找与当前边夹角最小的边扩展下去，最终一定能按顺时针沿图形边界环绕一周，回到 S。这样我们就确定了边界上有哪些边，也就找出了边界上有哪些点。