Winter Camp 2013 试题解题报告

南京外国语学校　许昊然

Contents

1 Winter Camp 2013 平面图 **2**

1.1 题目大意 2

1.2 思路分析 2

1.3 针对性算法讨论 3

1.4 通用算法讨论 4

2 Winter Camp 2013 糖果公园 **5**

2.1 题目大意 5

2.2 思路分析 6

2.3 针对性算法讨论 6

2.4 通用算法讨论 7

3 Winter Camp 2013 小Q运动季 **8**

3.1 题目大意 8

3.2 算法讨论 8

4 Special Thanks **10**

1 Winter Camp 2013 平面图

关键词

平面图与对偶图 点定位数据结构 MST 倍增祖先 计算几何

1.1 题目大意

给定一张n个点m条边的连通的平面图。边上有权值。

要求回答q组询问，每组询问给定两个点，要求：以一任意曲线连接这两个点，曲线不得经过平面图的无穷域，且曲线穿过的边的权值的最大值尽可能小。 要求回答这个值最小为多少。

数据规模见下表，时间限制5秒。

<table>
<tr><th>编号</th><th>n、m、q范围</th><th>x、y范围</th><th>其他特征</th></tr>
<tr><td>1</td><td>$n = 10,\ m = 13,\ q = 20$</td><td rowspan="2">$0 \le x \le 1, 0 \le y \le 2000$</td><td rowspan="5">所有直线段长度均为1</td></tr>
<tr><td>2</td><td>$n = 2012,\ m = 3016,\ q \le 1000$</td></tr>
<tr><td>3</td><td>$n, m \le 50\ q \le 200$</td><td rowspan="3">$0 \le x, y \le 1000$</td></tr>
<tr><td>4</td><td rowspan="2">$n, m, q \le 10^5$</td></tr>
<tr><td>5</td></tr>
<tr><td>6</td><td rowspan="2">$n, m, q \le 2000$</td><td rowspan="5">$0 \le x, y \le 10^7$</td><td>每个有界区域都是凸多边形， 且每条直线段权值均等于1</td></tr>
<tr><td>7</td><td>每条直线段权值均等于1</td></tr>
<tr><td>8</td><td rowspan="3">$n, m, q \le 10^5$</td><td rowspan="3">无</td></tr>
<tr><td>9</td></tr>
<tr><td>10</td></tr>
</table>

对所有数据，都有$5 \le n, m, q \le 10^5$；所有直线段权值不会超过10^9； 所有询问坐标都是不超过10^7的实数，且保证为0.5的整数倍； 所有顶点坐标都是整数。

1.2 思路分析

我们不妨考虑一个简化版的问题。

给定一张无向连通图，边上有权。 反复询问： 给定两个顶点，要求找到一条连接这两个顶点的路径，使得路径上最大的边权尽量小，求这个值。

经过观察可以发现，所选择的最优路径必然是该图最小生成树中连接这两点的路径。

证明： 如果存在一条路径的最大边权比这条路径还要小，那么这两条路径必然组成了一个环，我们将环上最大的边（这必然 属于最优路径，在MST上）替换成这条路径的最大的边（不在MST上，因为MST显然无环），必然能得到一个更优的MST，与 最小生成树的最小性矛盾。

因此，我们只需求出该无向图的MST，问题即可转化为：

给定一棵边上带权的树，反复询问某两顶点的路径中最大的边权。

这显然可以倍增祖先并记录最大值即可。 因此，这个问题可以在$O(n \log n) - O(\log n)$内解决。

解决了上述问题，题目任务就很明确了。

我们可以发现，该平面图的对偶图就是上述问题中的图。 而为了将两个给定的点映射到对偶图中的顶点，我们需要建立平面上的点与对偶图的顶点的对应关系，也就是查询一个点属于平面图的哪个域。

我们实际要解决的，是以下两个问题：

- 给定一张平面图，求出它的对偶图。（转对偶）
- 给定一张平面图，支持查询平面上某点属于平面图的哪个区域。（点定位）

1.3 针对性算法讨论

下面给出针对各个数据点的特殊性质而可以设计的算法。

数据点1～2

数据点1、2的规模很特殊，n和m都是给定的，顶点的横坐标要么是0要么是1。

简单计算一下就可以发现，这两个数据给定的图必然是一竖列1*1的方块，显然可以很方便的求出某个点属于哪个方块，而该图的对偶图就是一条链， 连接给定两个点的最优方案，也就是对偶图上这两点对应的顶点间的路径。最优权值就是其所经过的横边的权值的最大值。

于是我们发现，这实际是一个序列问题： 给定一个序列，查询其一个子序列的最大值。因为规模只有2000，完全可以暴力。 也可以选择ST表等数据结构。

期望得分： 20分

数据点1～5

前5个数据点都有一个共性：所有直线段的长度均等于1，且坐标规模不大。 这个性质可以怎么利用呢？

因为所有的顶点都在整点上，可以发现，『所有直线段长度均等于1』等价于『所有的直线段都平行于x轴或y轴，且平面图的边上的每个整点都是顶点』

因此可以发现，这张图实际是一张满网格图删掉若干顶点以及与它们相连的边而形成的“类网格图”，它的基本组成单位依然是1*1的小方块。

因为坐标规模不大（$x \leq 1000, y \leq 2000$），我们可以直接开一个$1000 * 2000$的数组表示所有的网格，并记录每个格子的四条边是否存在。 然后对这个数组进行FloodFill即可标记出每个单位方格属于哪个区域，这样就求出了对偶图，而且因为可以直接找到平面上一点对应的小方块， 于是可以直接得到一个点属于哪个域，于是解决了点定位问题。

期望得分： 50分

数据点6

这个数据点中，n, m, q规模都不大，可以使用每次回答$O(N)$之类的暴力算法。 这个数据点保证了平面图所有的封闭区域都是凸的，且边权都是1.

因此，实际我们需要判定的仅仅是：这两个点是否属于无限域、这两个点是否属于同一个区域中。

因为所有的封闭区域都是凸的，两点属于同一区域的充要条件是两点连线与平面图中的线段均无交点。

而因为整个平面图也是凸的，我们可以对一个点随机往外引若干射线，如果这些射线均不与平面图的边相交，则可以认为该点属于平面图的无限域（因为平面图是凸的，所以无限域中一点随机引射线与平面图相交几率不会大于$\frac{1}{2}$，引多条射线即可得到可以接受的准确率， 而平面图内的点无论怎么引射线都必然和平面图相交）。

期望得分： 结合前面的算法60分

数据点6～7

这两个数据点中，n, m, q规模都不大，可以使用每次回答$O(N)$之类的暴力算法，且保证了平面图所有边权都是1.

因此，实际我们需要判定的仅仅是：这两个点是否属于无限域、这两个点是否属于同一个区域中。 但这次平面图不再是凸的，上一个算法无法使用。

因此，我们需要使用平面图求域的通用算法（见后文）求出平面图所有的区域，然后暴力枚举所有域，用射线法判定该点是否属于这个域即可。

期望得分： 结合前面的算法70分

1.4 通用算法讨论

平面图转对偶图

这个问题可以使用“最左转边”算法。 先把无向边拆成两条有向边，然后把每个顶点引出的有向边按极角排序。

我们枚举所有还没有被选中的有向边，选择其指向结点引的边中，相对于当前边“最左转”的那一条边作为下一条边。 反复进行，直至回到了第一条边，我们就找到了平面图的一个区域。

实际上，“最左转边”就是在该顶点的极角序中，当前边的反向边的下一条边。

那么怎么判定无限区域呢？ 一个比较简单的方法是，求出每个区域后，算出每个区域的面积，面积最大的区域就是无限域。

利用C++ STL库可以极大方便实现，不到50行就能写出这个算法，实现很简便。

点定位数据结构

点定位问题是经典问题，传统解决方法是梯形剖分算法。当然这个算法极为复杂，考场上几乎不可能在给定时间内正确的实现出来。

实际上我们有更简单的算法： 离线处理+扫描线维护。

我们把所有顶点的x坐标作为关键点排序，把平面图切成一条一条的，用平衡树维护当前的一条的区间情况。 我们找到所有指向x轴正方向的有向线段，并记录其下方是什么区域。因为线段不会在端点以外的地方相交，因此各个线段之间的相对高低关系一旦确立，就始终保持不变。 我们只需在关键点处先删除以该关键点为终点的有向线段，再插入以该关键点为起点的有向线段即可。

实际实现时，有个巧妙的办法。 当删除线段时，以上个x区间的中点处各个线段的y坐标为关键字进行判定删除； 当插入线段时，以当前x区间的中点处各个线段的y坐标为关键字进行插入。 因为线段只在端点相交，在一个区间的中点处，所有线段的y坐标 两两不同，是个非常有用的性质，可以有效避免平衡树中的特判。

当然，如果不愿意写离线，也可以写在线算法： 使用可持久化treap作为平衡树维护即可，反正从无旋转treap到可持久化treap只要加几句话就行了。 但注意可持久化数据结构非常耗内存，注意内存使用。 我这道题使用可持久化treap时花了469MB内存，险些就爆内存限制了。 在必要时可以使用引用计数+垃圾回收减少内存消耗。

利用C++ STL库可以有效方便实现，关键代码仅有60行左右。

最小生成树、树上路径最大值

这些问题都比较基础，最小生成树可以使用kruskal或prim算法，树上路径最大值只需在倍增祖先算LCA的时候顺带记下最大值就可以了。 也可以tarjan算法离线$O(n+Q)$直接求出树上路径最大值。

2 Winter Camp 2013 糖果公园

关键词

数据结构 分块

2.1 题目大意

给定一棵n个点的无根树，经过一个点i会得到一颗类型为C_i糖果，总共有m种糖果，第i种糖果有一个美味度V_i。

还有一个长度为n的序列W，表示新奇指数。 第k次吃第i种糖果会增加$W_k * V_i$的愉悦值。

要求支持2种操作：

- 给定两个顶点a,b，查询从顶点a沿简单路径走到b后能收获多少愉悦值。
- 修改某个点i分发的糖果类型C_i。

数据规模见下表，时间限制10秒。

编号	n	m	q	其他限制
1	≤ 20	≤ 20	≤ 20	无
2	≤ 2000	≤ 2000	≤ 2000	
3	≤ 10000	≤ 10000	≤ 10000	
4	≤ 80000	≤ 100	≤ 80000	没有修改操作，给出的图构成一条链
5	≤ 90000	≤ 100	≤ 90000	
6	≤ 80000	≤ 80000	≤ 80000	没有修改操作
7	≤ 90000	≤ 90000	≤ 90000	
8	≤ 80000	≤ 80000	≤ 80000	给出的图构成一条链
9	≤ 90000	≤ 90000	≤ 90000	
10	≤ 100000	≤ 100000	≤ 100000	无

2.2 思路分析

此题看起来十分复杂，要维护的东西也很棘手。

一般来说，数据结构题有四种比较通用的方法： 树形数据结构，分块数据结构，区间分块预处理，转移询问区间。

不难发现，本题要维护的东西不具有任何快速合并性质，因此难以用树形数据结构来维护；

而且，答案也不具有任何快速合并性质，因此，一般的分块数据结构也难以胜任。

于是，剩下可能的可行方法就只有区间分块预处理和转移询问区间了。

2.3 针对性算法讨论

下面给出针对各个数据点的特殊性质而可以设计的算法。

数据点1～3

数据点1～3的规模都很小，完全可以直接暴力处理。

期望得分： 30分

数据点4～5

数据点4～5中，n和q的规模很大，但m的规模很小，而且数据保证是一条链，且没有修改。

因此我们不妨对每个颜色维护其出现的位置，并用树状数组维护部分和，统计时计算每种颜色对答案的贡献即可。 时间复杂度$O(m \log n)$每次查询，$O(\log n)$每次修改，完全可以接受。

期望得分： 结合前面算法50分

数据点6～7

数据点4～5中，n、m、q的规模都很大，但数据保证没有修改操作。

如果数据还保证了这是一条链，那么一个经典的做法就是转移询问区间： 将序列等分成$O(N^{0.5})$块，枚举起始块， 回答所有左端点在起始块，右端点在起始块右侧的询问。 这可以通过将所有左端点在起始块中的询问按右端点排序， 然后右端点从小到大移动，同时更新当前维护区间的答案，并把维护区间左端点移动到询问区间左端点。 每个询问会 移动左端点不超过$O(N^{0.5})$的距离（因为左端点始终在一个$O(N^{0.5})$大小的块里），而每次枚举起始块右端点 会移动总计$O(N)$的距离，总共有$O(N^{0.5})$个起始块要枚举，因此总时间复杂度$O((N+Q)*N^{0.5})$。

在树上的情况做法是相似的。 我们可以通过一个简单的dp把树分解成$O(N^{0.5})$个大小为$O(N^{0.5})$级别，且块内最远点距离 也是$O(N^{0.5})$级别的块（对每个结点递归处理子树，然后把子树未划分结点每凑满了$N^{0.5}$个就分成一块即可）。 然后我们 枚举每个起始块，回答所有询问的一个端点在起始块中的询问。 我们按照树的dfs序来对右端点排序，按dfs序枚举右端点，并移动维护区间 左端点到询问左端点即可。 时间复杂度证明同上，也是$O((N+Q)*N^{0.5})$的。

期望得分： 结合前面算法70分

数据点8～9

数据点8～9中，n、m、q的规模都很大，但数据保证是一条链。

此时转移询问区间已经不可行，因为上述方法无法处理修改。但我们还有另一种方法，区间分块预处理。 我们把整个序列分成$O(N^{\frac{1}{3}})$ 块，并预处理所有连续的块区间（共有$O(N^{\frac{2}{3}})$个）的答案，并记录下每种糖果出现了几次。

询问时，只需找到询问区间包含的最大的已经被预处理过的区间，此时在区间外的元素不超过$O(N^{\frac{1}{3}})$个，直接暴力更新答案即可； 修改时，因为预处理的块区间只有$O(N^{\frac{2}{3}})$个，对包含了修改元素的这些块区间直接更新记录即可。

时间复杂度$O((N+Q)*N^{\frac{2}{3}})$，空间复杂度$O(N^{\frac{5}{3}})$，需要一些常数优化方可通过。

期望得分： 结合前面算法90分

2.4 通用算法讨论

算法一

首先说明一个性质。 在一棵树上随机选择$O(N^{\frac{1}{3}})$个点，那么树上任意一点距离最近的被选择的点的期望是$O(N^{\frac{2}{3}})$级别的。 有兴趣的读者可以自行证明。

利用这个性质，一个想法产生了： 随机在树上标记$O(N^{\frac{1}{3}})$个点，记录下这些点两两之间的路径（共有$O(N^{\frac{2}{3}})$条）的答案， 并记录下每个颜色出现了多少次。 查询时，找到查询的端点距离最近的被标记的点，并把端点转移过去，期望的转移距离是$O(N^{\frac{2}{3}})$，因此 时间复杂度也是$O(N^{\frac{2}{3}})$； 而修改时，找到包含被修改点的路径（最多$O(N^{\frac{2}{3}})$条），直接更新答案即可。

时间复杂度期望$O((N+Q)*N^{\frac{2}{3}})$，尽管理论上不错，但实际常数较大，难以跑进时限；空间复杂度$O(N^{\frac{5}{3}})$，常数也较大，难以卡进内存。

期望得分： 30～100分 （我写的这个算法虽经反复优化，依然只能得到30分，不知道诸位神犇有没有更好的写法）

算法二

我们考虑将针对测试点6～7和针对数据点8～9的算法结合起来。转移询问区间的方法难以处理修改，原因是我们为了按照我们的顺序处理右端点， 不得不改变了右端点的序，从而使得修改无法应用。 因此我们考虑不再按照dfs序处理右端点，而也像左端点一样，做块内移动。

于是，一个想法产生了，我们枚举起始块和终止块，并按时间顺序处理：修改时，如果修改的点在当前路径上，就更新记录； 询问时，把当前路径的左右端点 分别转移到询问的两个端点。 这样的话，不妨假设块的大小是S，那么枚举的复杂度就是$O(\frac{Q*N^2}{S^2})$，询问的总复杂度是$O(QS)$。 为了使总复杂度最低，S应当取$O(N^{\frac{2}{3}})$。 因此，我们将树按照针对数据点6～7的方法分成$O(N^{\frac{1}{3}})$块，然后枚举起始块 和终止块，然后按时间顺序处理即可。

时间复杂度$O((N+Q)*N^{\frac{2}{3}})$，与算法一相同，但常数较小，经过适当的常数优化，便可以在时限内通过。

期望得分： 90～100分

3 Winter Camp 2013 小Q运动季

关键词

中国剩余定理 高斯消元 乘法逆元

3.1 题目大意

给定一线性同余方程组，求一组解使得尽可能多的方程得到满足。

提交答案题，根据你的解的优劣评分。

3.2 算法讨论

数据点1

本数据点只有一个变量，有5000个式子，同余方程模数相等，且不大。

算法显然，暴力枚举这个变量的值即可。

期望得分： 10分

数据点2

本数据点只有一个变量，有50个式子，同余方程模数不同，且两两互质，质因子很小。

首先判定一个式子是否可能可行，如果不可行直接排除。 剩余的式子中，因为模数两两互质，且质因子很小，可以直接暴力中国剩余定理+高精度。

期望得分： 10分

数据点3

本数据点有300个变量，300个式子，同余方程模数相同且为质数。

经过高斯消元试验，可以发现存在一组解满足所有的方程。

期望得分： 10分

数据点4

本数据点有20个变量，845个式子，同余方程模数相同且为质数，其中前800个式子由40个不同的式子组成，每个式子重复20遍。 而40个不同的式子分成20组，每组2个式子系数均相同，仅余数不同。 从每组中任取一个式子，得到的20个式子能组成下三角矩阵。最后45个式子杂乱无章。

可以发现，我们应该让那些重复20组的式子尽可能满足。 我们枚举每组选择哪个式子2^{20}，求出它的解， 然后带入最后45个式子看有多少额外的式子满足，取最多的即可。

期望得分： 10分

数据点5

本数据点有100个变量，100个式子，同余方程模数相同但不是质数。

把模数分解成质因子，对每个质因子模数分别进行高斯消元试验，可以发现存在一组解满足所有的方程。用中国剩余定理合并答案即可。

期望得分： 10分

数据点6

本数据点有50个变量，50个式子，无其他性质。

把模数分解成质因子，对每个质因子模数分别进行高斯消元试验，可以发现存在一组解满足所有的方程。用中国剩余定理合并答案。需要高精度。

期望得分： 10分

数据点7

本数据点有50个变量，200个式子，模数相同且不大。 其中一个式子重复了50遍，剩下150个式子都是“某变量等于某值”的最简单形式， 每个变量恰好出现在3个式子中。

显然我们应该满足那个重复了50遍的“VIP式子”，并且尽量用剩余150个式子中的取值。于是我们用$dp[i][s]$表示当前到了第i个变量， VIP式子当前的值是s时是否可行。 转移枚举第$i+1$个变量的3个取值即可。

经试验发现，所有的变量都使用给定的三个值之一就能满足VIP式子，因此这个解必然最优。

期望得分： 10分

数据点8

本数据点有50个变量，50个式子，无其他性质。

我们如果照搬数据点6的做法，就会发现，此时高斯消元找不到合法解。 于是，一个很自然的想法就是，一旦发现一个式子无法满足，就把它删掉，从头再做一次。

为了充分利用提交答案的性质，我们还可以加入一些随机化成分，比如在高斯消元之前随机打乱式子的顺序，这样就能找到不同的无法满足的式子来进行删除试验， 多次试验取最优解，能取得很好的效果。

事实上，这个数据点只有1～2个式子是无法满足的（视第一次删除的式子而定，如果删的好就只需要删一个，删的不好就要删两个）。 加上随机化后，只要跑几次 就能跑出满足49个式子的最优解了。

期望得分： 10分

数据点9

本数据点只有一个变量，有100个式子，模数相同。

这个数据点是送分点，显然统计一下数据中哪个数出现次数最多就可以了。

期望得分： 10分

数据点10

本数据点有50个变量，90个式子，无其他性质。

这个点如果直接分解质因数+高斯消元，也是找不到解的，而且有大量无法同时满足的方程组。 我们用数据点8的随机化做法来跑，反复执行取最优解。

经实际试验，这个算法几乎总是能找到24以上的解，从而得到5分。 如果花更多时间来跑，可能可以得到更多的分数。 我目前得到的最好的解是30.

期望得分： 5～10分

4 Special Thanks

- 感谢中国计算机学会提供了这个交流的平台。
- 感谢龙浩民同学在LaTeX方面给予我的大量帮助，让我能用LaTeX写出这篇题解。