火车司机出秦川 解题报告

安师大附中 吴作凡

1 试题来源

2016年国家集训队互测

2 试题大意

如果一个无向连通图的任意一条边最多属于一个简单环，我们就称之为仙人掌。所谓简单环即不经过重复的结点的环。

给定一棵n个点m条边的仙人掌，每条边都有边权，有q次操作，每个操作分为两步：

1. 给定仙人掌上的k_i条路径（经过点最多或者最少的简单路），求它们并集的边权之和。

2. 修改某条边的边权。

令$S=\sum_{i=1}^{q} k_i$，数据范围如下表所示：

测试点编号	n	m	q	S	是否有修改
1	=2	$n-1 \le m \le 2n-2$	≤ 10	≤ 10	有
2	=100	$n-1 \le m \le 2n-2$	≤ 100	≤ 100	有
3	=100000	$m=n-1$	≤ 100000	≤ 100000	无
4	=100000	$m=n-1$	≤ 100000	≤ 100000	有
5	=300000	$m=n-1$	≤ 300000	≤ 300000	无
6	=300000	$m=n-1$	≤ 300000	≤ 300000	有
7	=100000	$n-1 \le m \le 2n-2$	≤ 100000	≤ 100000	无
8	=100000	$n-1 \le m \le 2n-2$	≤ 100000	≤ 100000	有
9	=300000	$n-1 \le m \le 2n-2$	≤ 300000	≤ 300000	无
10	=300000	$n-1 \le m \le 2n-2$	≤ 300000	≤ 300000	有

为了保证路径的唯一性，保证仙人掌所有环长度均为奇数，任意时刻边权总和在int范围内。

3 算法介绍

3.1 算法一

因为$n = 2$而且所有环长均为奇数，所以不存在环，只会有一条边，那么只要k不为0就输出边长就好了，时间复杂度$O(n + q)$，可以通过第一个测试点，期望得分10分。

3.2 算法二

第二个点S和n都很小，可以暴力找出每条询问的路径，求出并集再求和。

这里需要用到仙人掌的一些性质——任意两个简单环都不共边，于是可以先dfs出一棵生成树，每个简单环都可以表示为树上一条路径和连接两个端点的一条非树边。对于询问的路径(u, v)，先走树上的路径，然后枚举每个环，显然每个环都有两种走法，如果从另一边走更优就更改路径。因为没有两个简单环共边，时间复杂度$O(Sm)$，可以通过前两个测试点，期望得分20分。

3.3 算法三

接下来4个点都不是仙人掌，而是我们熟悉的树，那么这个问题就变成经典的树链的并了，解决这个问题的方法有很多。

可以对树进行轻重链剖分，将树上的问题转化到序列上，那么每次就是$O(k \log n)$个区间求并，我们可以排序后合并区间再询问，利用线段树或者树状数组维护序列就好了，复杂度$O(S \log^2 n)$，期望得分40~60分。

对于没有修改操作的测试点，可以用前缀和替换线段树，常数会减小很多，复杂度$O(S \log^2 n)$，期望得分50~60分。

当然也可以用动态树替换轻重链剖分，打上时间戳标记来求并集，复杂度$O(S \log n)$，期望得分60分。

3.4 算法四

仙人掌和树很类似，可以用我在2016年冬令营上营员交流的方法将仙人掌转化为树并轻重链剖分（这里就不再赘述），然后用线段树维护时间戳标记来求并集，复杂度$O(S\log^2 n)$，期望得分80~100分。

3.5 算法五

本题的范围是$n \le 300000$，这提示我们存在$O(S\log n)$的算法，而且仙人掌的算法几乎都是由树上算法修改而来，但是之前树上的最优秀算法是动态树，而动态仙人掌的复杂度是$O(S\log^2 n)$（由于我的水平有限，不是很清楚能不能使用该算法解决这个问题，如果可以解决请联系我），所以需要想出树上的其它方法。

如果询问次数很少，我们可以对于每次询问遍历一遍整棵树，对于一条路径(u, v)，在u, v和$LCA(u, v)$（即u, v的最近公共祖先）上打上标记，就可以知道每条边是否会被计算了。

我们没有必要对整棵树经行遍历，可以发现关键点不多，可以对于这些关键点重新构建一棵树，称为虚树，一个最简单的构造方法是先将所有关键点按dfs序排序，然后维护一个栈，依次加入点，如果栈顶是当前点的祖先就连边，否则弹栈。

然后我们遍历这棵虚树，可以发现虚树的大小是$O(k)$的，于是只要求$O(k)$条链的长度就好了，那么就需要支持$O(\log n)$询问一个点到根的距离，这可以用线段树维护dfs 序解决。复杂度$O(S\log n)$，期望得分60分。

3.6 算法六

接下来就让我们考虑如何将虚树算法拓展到仙人掌上。

3.6.1 建树

仙人掌和树非常类似，我们考虑如何将仙人掌转化为树。

容易想到将每个环缩为一个点。我们可以选择一个点当根进行dfs，将环缩为点，注意到环中除了深度最浅的点（称为环的根），其它点的最长路最短路都会覆盖到环中点，所以我们不将环的根和其他点缩起来，这样一个点会被缩到

它到父亲那条边的环上，每个点只会对应树中的一个节点。我们可以称这棵树为缩环树[1]。

3.6.2 询问

那么我们可以对缩环树建出虚树，对点打上标记然后扫描虚树，为了计算虚树上边的贡献，我们仍然需要维护一个点到根的最长路、最短路、最长路和最短路的并集（最长路和最短路是有相同边的，所以也需要维护并集）。

不同于树，缩环树的节点内部存在边，也会对答案产生贡献。考虑什么时候会产生这些贡献，有两种来源——环上点通过环的根继续向上走，或者一条路径的LCA是这个环。容易发现这两种贡献都是一些区间，而区间总数是$O(k)$的，那么我们可以排序后合并相交区间，所以还需要维护每个环内部的边权。

3.6.3 维护

修改只会修改一条边的权值，那么环内部的边权用树状数组就很好维护。

一个点到根的最长路、最短路、最长路和最短路的并集维护也和树一样，可以用线段树维护dfs序解决，但是细节较多需要讨论。

3.6.4 一些实现上细节

注意到我们需要求一个点是一个环的哪个点的孩子，这里可以二分并用dfs序判定，所以我们建立dfs序的时候要按顺序遍历环上每个点及其孩子，这样可以保证每个点的子仙人掌是一个区间。

另一个需要注意的地方是缩环树上每个点都有两种可能的含义——一个点或者一个环，那么几乎所有的操作都要对这两者进行讨论。

因为是多组询问，还要注意标记清空。

虚树的标记要打在仙人掌的点上，而不是树的点上，仙人掌和缩环树的点不要搞混。

[1]这是我起的名字

3.6.5 时间复杂度

对于单次询问，建立虚树、计算所有环内询问和路径询问的时间复杂度都是$O(k \log n)$，那么总复杂度就是$O(S \log n)$，期望得分100分。

4 总结

本题改编于一个经典的数据结构问题——树链的并，将其推广到仙人掌上，主要考察了选手对经典算法的掌握以及对仙人掌性质的分析能力。

可以发现直接使用我在冬令营营员交流上介绍的算法能够得到很高的分数（如果常数小甚至可以AC），这个题的出题思路就来源于此，是我在思考缩环树这个结构除了剖分还能做什么中产生的，所以如果理解了那个算法很容易想出标算。

可能一些选手能想出标算或者相当优的算法，但是实现起来可能有些困难，因为本题代码量比较大而且细节相当多，需要选手有较强的代码能力和调试能力。实际上我不并建议选手在正式比赛中花费过多时间去写这种题目，因为很可能得不偿失（我在清华集训2015的第二天就是因为这个原因只得了38分的），特别是部分分相当高的时候。

5 得分估计

因为存在$n = 2$以及100的点，所以所有的选手都应该至少得到10至20分。

而树部分是一个经典问题，所以绝大部分选手都会做，我估计至少有四分之三的选手是40至60分。

仙人掌部分虽然有较大的思维难度，但是$O(S \log^2 n)$的剖分算法已经在冬令营上介绍，降低了很多的难度，可惜代码实现较为复杂，所以我估计会有1到2人得到70 至100分。