Summer Tranning Week 4

比赛简记

Max.D.

专题

本周暂无

比赛

本周暂无

题目

本周暂无

Hardict

专题

比赛

题目

MountVoom

专题

无

比赛

求求来点cf div1

题目

无

个人总结

陈铭煊 Max.D.

多练习，跟上队友思路。

龙鹏宇 Hardict

肖思炀 MountVoom

感觉有些浮躁，希望能静下心来，做题的时候更加集中一些。

本周推荐

陈铭煊 Max.D.

来源：

牛客2020多校第七场 C. A National Pandemic

一道动态点分治题，后来又学到了一种树链剖分的做法，两个$\log$居然跑得比前者还要快。

标签：

动态点分治 树链剖分 树上距离的化简

题意：

T组数据，每组给出n个点组成的树，每个点有初始权值$F(x)=0$以及m个操作，操作分三种：

1. 给出点x和w为所有点增加权值$w-dist(x,i)$$dist(x,i)$表示$x$和$i$的树上距离；
2. 更新点x处的权值为$\min(F(x),0)$;

3. 查询某个$F(x)$

$1\le T\le 5,1\le n,m\le 5 \times 10^4,0\le w\le 10000$需要对操作$3$进行回答。

题解：

看到这种树上距离有关的更新，很容易想到了动态点分治，可惜没学到家，比赛时还仔细回忆了半天细节，最后还没写出来。

基本思路是，构造点分树，点分树上每个点实际上代表了一个树上连通区域，记录三个值$S,dp,ddp$$S[i]$代表点分树上节点$i$的子树中，操作$1$被执行的总次数$dp[i]$表示点分树上子树$i$中每个点到$i$父亲$pa[i]$的距离之和（如果$i$为根，取$pa[i]=i$$ddp[i]$表示点分树上子树$i$中每个点到$i$的距离之和。这三个值都可以在点分树上不断走根的过程中求出。

这个时候我们如果想知道操作1为$F(x)$带来的总的$dist$之和，可以从x开始，向上这样累加：

```
int x=u;
while (pa[u]) {
    ans += ddp[pa[u]] - dp[u] + Dis(pa[u], x) * (S[pa[u]] - S[u]);
    u = pa[u];
}
```

每次累加的值实际上是$pa[u]$的非u点分子树到x的总贡献，这个值由两部分组成，一部分是$pa[u]$在u这棵子树外的所有点，到达$pa[u]$产生的贡献和，也就是$ddp[pa[u]]-dp[u]$另一部分就是x到$pa[u]$这段路径长度产生的贡献——所有这些点都要走过这一段路。

注意一下ddp的值不一定等于儿子dp值的和哦。另外Dis函数可以用RMQ的方法求，这样就是真正$O(m\log n)$效率了。

现在既然距离的贡献都解决了，其他就很容易了，操作2额外记忆一个$delta$即可。

其实要是明白透彻了，实现起来也不难。

第二种方法就显得很巧妙了：

对于操作1，我们考虑一次修改对y来说会增加$w-dis(x,y)$$W-dis(x,y)=w-(dep(x)+dep(y)-2*dep(lca))=w-dep(x)-dep(y)+2*dep(lca)$所以，对于每次1操作，我们将其到根上所有点的$cnt+=2$询问的时候那部分就是求它到根的权值和。所以，树上路径加，路径查询，用树链剖分，再用数据结构来区间加，区间求和即可。

不得不说，这个式子很是传神。

评论：

好题

龙鹏宇 Hardict

来源：

标签：

题意：

题解：

评论：

肖思炀 MountVoom

来源：

牛客第八场 A. All-Star Game

标签：

线段树分治，并查集

题意：

球迷j愿意观看球员i的比赛需要满足□j是i的粉丝或者j和j'都是i'的粉丝且j'是i的粉丝。

选择最少的球员进全明星赛，使得所有球迷都愿意观看。且有q次粉丝关系的修改□x是y的粉丝变为不是□x不是y的粉丝变为是，每次修改都会询问。

题解：

把球员和粉丝相连，答案即为(n个球员的连通分量个数)- (孤立球员个数)，如果有球迷的度为0，则答案为-1。

把每个关系存在的时间划分成线段树上的log个区间，线段树每个节点维护了在这段时间存活的所有边，最后遍历一次线段树即可，因为回溯时需要删边，所以需要用启发式合并的并查集。

最终复杂度为$O(n \log^2 n)$

评论：

比赛做得太慢了以至于没有时间写这道题…