T1

离线+并查集

删点是永久性的，考虑离线询问倒序做，用并查集维护连通块，并记录每个连通块的点权和。

每轮加入点合并连通块时，新增的文明价值即为两个连通块点权和的乘积(乘法分配律)。

小清新送分题。

T2

本质上就是求所有两端点跨过当前连线覆盖区域的直线数(一端在区间内，一端在区间外)：

考虑某次连边操作为L, R(定义$L < R$，如果输入中$L > R$，交换L, R是等价点)，现存的直线有E条，求的就是$\sum\limits_{i=1}^{E}([L < l_i < R]! = [L < r_i < R])$

于是有一个显然的$O(nm)$暴力：维护现存的直线，每次$O(m)$回答询问&删边(需要遍历所有边找到这条并删去，边的位置是动态的)。可以得到$70pts$(良心)

这道题不好用差分，而考虑到每次求的是$[L < l_i < R]! = [L < r_i < R]$，等价于$[L < l_i < R]\ xor\ [L < r_i < R]$，且各点最多只有一条连边，于是将每条边转化为2进制数上的一位，每个点的值即为2^{tag_i}，tag_i表示点i所连编号(点i没有连边时，$tag_i = 0$)

拆环为链，维护从1~i点点权异或的前缀和sum_i，每次加边操作的答案即为$sum_L\ xor\ sum_R$二进制表示中值为1的位数，直接用$__builtin_popcount O(1)$计算。

设E为同时存在的最多的边的数量(显然$E \leq \dfrac{n}{2}$)，sum_i实际上为一个二进制有E位的数，考虑建立$\lceil\dfrac{E}{64}\rceil$/$\lceil\dfrac{E}{32}\rceil$($unsigned\ long\ long/unsigned\ int$)颗大小为$n$的树状数组维护$sum$。std复杂度$O(\lceil\dfrac{E}{64}\rceil m\log n)$。

$p.s.$需要建一个内存池动态维护边的标号(保证边的标号$\leq E$)

T3

对于$1-6$号测试点，挑选你喜欢的最短路算法，获得相应的分数。

对于$7-8$号测试点，注意到如果把$1-n$构成的环拿出来，所有环外的边没有相交/包含关系。

那么每次的决策就是走非环边或者走环边，可以直接考虑dp一下。

对于$9-12$号测试点，对每次询问的最短路进行分析，发现只有两种情况：

1、经过环外边。

2、不经过环外边。

不经过环外边好说，我们可以直接从环上计算。但是经过环外边的情况怎么统计呢？

这里我们引入一个概念，我们把环外边的所有端点称作关键点。

想到可以对每一个关键点预处理出到所有其他点的最短路长度。

考虑到m的值很小。每次询问，枚举经不经过第k条环外边。设i到j的最短路为$dis(i,j)$，每一次查询x到y答案就是：

$$min(dis(u_i,x)+dis(v_i,y)+w_i)$$

因为是双向边，同时要统计：

$$min(dis(v_i,x)+dis(u_i,y)+w_i)$$

设n个点m条边最短路的复杂度为$S(n,m)$

总的复杂度$O(mS(n,E)+qm)$

对于100%的数据：

发现上面的做法并没有充分利用好复杂度。

首先我们可以拆环为链，因为实际上可以把$1-n$的这条边看做一个链外边，这样就是一个链上的情况了。

再次考虑所有关键点，我们做这样一个连边：

1、在链上每一个关键点向左右两边最近的关键点连边。

2、关键点之间连接链外边。

这样我们成功优化了图的大小，每一次询问把待询问的点加入图中，直接在新图上跑最短路。跑完之后在前向星链表中删除对应边就行了。

总的复杂度为：$O(n+qS(m,m))$

更多的惊喜？

在本场考试的题面中隐藏着一个彩蛋，聪明的你发现了吗？