Snow

dyp

subtask1

应该没有人不会NOIP2018铺设道路吧?

subtask2

考虑一个点作为端点会被统计到多少条路径内，由于每一条边的覆盖次数是固定的，因此假设一个点连的边的边权分别为z_1, z_2，那么这个点的作为端点的最小贡献就是$|z_1 - z_2| \cdot w_x$，$O(n)$即可解决。

subtask3~7

考虑将上面的方法扩展到树上，依旧是单独考虑每一个点x作为路径的端点的贡献，那么相当于是将x连着的边尽可能匹配，最后剩下的不能匹配的边的边权就是x要伸出去的路径的条数。

不难发现一个结论，假设x度数du_x为偶数，那么如果$2z_{max} \le du_x$一定能够全部匹配完，其中z_{max}表示x连向的边中最大的边权。

证明:

> 考虑贪心，每一次选择的肯定是最大和次大来匹配。
>
> 设$a = z_{max}$，如果有$2a > du_x$，那么肯定没有完美匹配，贡献显然是$2a - du_x$。
>
> 否则有$2a \le du_x$，接下来只需要证明，每一次删去最大和次大还能满足$2a \le du_x$，就能够保证一直删到$a \le 1$，就一定满足条件了。
>
> 首先最大边和次大边不可能成为新晋的$2a > du_x$，考虑第三大的边v，如果$2v \le du_x$，但是$2v > du_x - 2$，即成为了"过半边"。由于它是第三大，所以$du_x \ge 3v$，即$2v > 3v - 2$，则$v = 1$，与上面的假设矛盾。
>
> 因此如果有$2a \le du_x$，一定能够删完。

即$2a \le du_x$时无贡献，$2a > du_x$时贡献$(2a - du_x)w_x$

再考虑如果度数为奇数，那么至少有一条链，因此我们可以先把这条链分配给权值最大边，然后再做上面的过程，时间复杂度$O(nq)$，在随机树、菊花等特殊的树上由于链短，所以也可以很快处理完。

subtask8

上面的判定相当于：$2(a - du_x\&1) > du_x - du_x\&1$，即$2a > du_x + du_x\&1$，而贡献为$(2a - du_x)w_x$。

对于一条链的区间加，对于中间的点，首先$du_x\&1$不变，$2a - du_x - du_x\&1$也不变，因此中间的点是不会有影响的。

所以只需要$O(1)$修改端点即可。

subtask9

同理我们可以考虑树上的链加，对于端点暴力处理。

中间的点，会有两条边的权值$+d$，那么如果有一条边是最大边，那么设$p=2a-du_x-du_x\&1$,修改之后$p'=2(a+d)-du_x-2d-(du_x+2d)\&1=p$，不变。

花费$2a-du_x=2(a+d)-(du_x+2d)$，不变。

如果改的都不是最大边，那么$p'=p-2d$，花费也相应变小，直到$p\le 0$之后这个点的贡献就只与奇偶性有关了。

因此，对于一个修改到的边，它不可能从非过半边变为过半边，如果修改到的是最大值，那么这次修改对于这个点没有意义。

首先不妨考虑除去端点以及lca，只剩下两条链的情况，很容易想到链剖，我们可以对于那些过半边既不是重儿子也不是父亲边的点在链剖上维护$\sum w_x$和p_{min}。

在重链上区间减p：在区间减之前找到可能会变为0的p_x，暴力修改这一部分的贡献，然后再进行区间减。由于只有链的端点会打破上面的结论，因此每一次操作的势能最多增加2，暴力删除的次数不超过$O(n+q)$。

在跳轻儿子的时候暴力处理贡献。

考虑暴力处理的点（端点、lca，轻儿子）：可以用树状数组简单差分维护du_x,z_i，然后暴力看一看过半边的贡献即可。

总的时间复杂度是$O(n\ log^2n)$的（n,q同阶）。

~~数据较弱还请见谅~~