travel

首先考虑q=0怎么做。

下面假设图连通。

随便找出一棵生成树。

将一条非树边对应的环定义为非树边并上两个端点在树上的路径。

从1开始的任意路径的异或值就等价于从1开始的一条树链异或上一些非树边对应的环。

求出所有非树边对应的环的线性基，

对于每条从1开始的树链，在线性基上跑一遍，使得线性基上存在的那些位都通过异或变成0，

然后数一下有几个不同的结果，乘上2^(线性基大小)即可。

q>0的情况是类似的。

因为只用考虑1所在的连通块，所以加入的边如果两个端点都不在1所在的连通块我们就不用管它。

否则，如果两个端点都在1所在的连通块，就加入了一条非树边，那么我们在线性基中插入它对应的环，

如果插入成功，这种情况最多只会发生logW次，这时就暴力把所有树链重新跑一遍。

否则，两个端点一个在1所在的连通块，一个不在，这时就dfs一下新的连通块来得到新的连通块带来的树链和非树边，

然后分别执行树链的插入和非树边对应的环的插入。

时间复杂度$O(nlog^2W+(m+q)\alpha(n))$ 。

modulo

设原来的序列为a，定义差分序列b，$b_i=a_i-a_{i-1}(i>1),b_1=a_1$

则对 $a[l\cdots r]+1$ 相当于 $b[l]+1,b[r+1]-1$ (若 $r=n$ 则只有 $b[l]+1$);

a 变成 0 就相当于 b 变成 0 。

定义解的序列c，c[i]表示b[i]被加了几，就是说i每作为一次左端点c[i]就+1，每作为一次右端点+1 c[i]就-1。

注意，b[i]是模k意义下的，但c[i]不是。

只对[l,n]操作可以得到一个初始解c0，c0[i]=(k-b[i])%k。

注意到，对于答案的解序列，有c[i]=c0[i]或c[i]=c0[i]-k。

一个c合法的条件就是所有前缀和非负，一个c对应的操作次数就是c中正数的和。

考虑从c=c0开始，每次贪心将一个c[i]从c0[i]变成c0[i]-k，来得到答案的解序列。

那么显然一次改变会将答案减少c0[i]，而对前缀和的影响为将[i..n]减少k。

因此我们按c0[i]从大到小，c0[i]相等时从后往前贪心，判断能否将c[i]从c0[i]改成c0[i]-k即可。

考虑证明这个贪心的正确性。

对于同一个c0[i]，最优方案选择改变的i显然是一段后缀。

如果一个方案对于最大的c0[i]选择改变的后缀没有达到最长，显然可以从小的c0[i]那里调整，使得答案变大。

时间复杂度 $O(nk)$ 。

real

令 s_i 表示[1..i]有几个1。

则我们希望 $|f-\dfrac{s_r-s_l}{r-l}|$ 最小，

即 $|\dfrac{f\times r-s_r-f\times l+s_l}{r-l}|$ 最小。

令 $p_i=(i,f\times i-s_i)$，则我们希望两点斜率的绝对值最小。

答案必定可以由两个按 y 排序后相邻的点取到，证明用反证法是显然的。

因此就可以 $O(n\log n)$ 解决此题了。