A.铭记(remember)

【简要题解】：拆位计算贡献。

在部分分的提示下，容易发现题目所求的式子中每一个二进制位是相互独立的，因此可以分开计算。

具体来说，如果该位是从低到高的第 i 位，该位为 0 的数有 c_0 个，该位为 1 的数有 c_1 个。则其对于按位与的贡献为 $\frac{c_1(c_1-1)}{2}2^{i-1}$，对于按位或的贡献为 $\frac{n(n-1)-c_0(c_0-1)}{2}2^{i-1}$，对于按位异或的贡献为 $c_0c_12^{i-1}$。

对于每个二进制位分别求以上答案再求和即可，时间复杂度 $O(n\log n)$。注意答案会爆 `long long`。

B.世界(world)

【简要题解】：回头路只会最后再走，动态维护调整下坠高度集合。

大部分情况下，在行走的过程中你所在位置的 x 坐标是单调递增的，也就是不会走回头路。但是还是会走回头路的，那什么时候会走回头路呢？

有一个结论：回头路只会在 y 坐标为 0 的时候走。证明：首先你在回头路的过程中一定不会遇到下坡，否则你可以在第一次来到这一列（x 坐标相等）时下去，这种方法一定更优。由于不会遇到下坡，所以横向移动与纵向移动的先后并不影响答案。

因此我们只需要维护 $(1,a_1)$ 到第 i 列的最小代价即可。显然对于你承受的总伤害只与每次下坠的高度有关，为了满足最多受到伤害的限制，我们需要存下所有下坠的长度。

如果当前的总伤害超出限制了，我们要怎么改变策略呢？注意到对于正整数 i,j，若 $i>j$，则 $i^2-(i-1)^2>j^2-(j-1)^2$。也就是说对于若干次下坠，每次改变的肯定是其中高度最大的那一次。有因为有 $i^2\ge(i-1)^2+1$，所以对于一个下坠，我们可以在下坠前手动挖掉一格来减少伤害并增加一个时刻的时间。

注意到改变次数对于总伤害具有单调性，因此可以进行二分。具体来说，二分的时候可以先二分一个最大值，再二分有多少次下坠高度可以取到这个最大值。为了保证复杂度，对于高度相同的需要合并处理，每次操作时对一段高度相同的下坠进行上述操作。

可以用 `set` 存储每一段高度相同的下坠。注意实时更新受到伤害的总和以及答案。时间复杂度 $O(n\log n)$。

C.桥梁(bridge)

【简要题解】：只保留变化信息，缩小信息量后暴力做。

不带限制的时候是一个最小生成树板题，因此难点在于限制。注意到 q 很小，因此**不**考虑优化掉 2^q 枚举。~~应该不会有人想折半搜索吧~~

发现其实有很多信息是不用枚举时去计算的，例如某些边在所有枚举中一定会被选中，某些边一定不会被选中。

考虑找到一定会被选中被边。发现强制钦定选定所有带限制的边后，跑最小生成树会被选中的边就是无论如何都会被选中的边。

再考虑找到一定会被不选中被边。去掉上述强制钦定选中的边，保留其余被选中的边，再次跑最小生成树，除带限制的边外不被选中的边就一定在所有方案中不会被选中。

到此为止我们确定了一部分一定会被选中的边和另一部分一定不会被选中的边，那么是否到现在为止未知信息是否已经足够少了呢？注意到，其余的未确定的边必须至少有一个端点与任意一条带限制的边的端点相同，正确性比较显然。因为带限制的边数至多为 $2q$，所以满足条件的端点至多为 $4q$，未确定的本质不同边数至多为 $O(q^2)$，已经足够少了。

通过精细实现可以做到时间复杂度 $O(m\log m+2^q q^2)$。

D.生长(growing)

【简要题解】：维护若干高度相同连续段，计算增长高度、相邻段合并时间并快速合并。

首先容易发现，如果相邻的两棵树在某一时刻高度相同，那么它们之后的高度会一直相同。因此我们可以把这些树作为若干个连续段考虑。因为相邻两个连续段会在某个时刻合并，因此我们需要也只要关注相邻两个连续段会在什么时候合并。

考虑一场风暴来袭时一个连续段的树木会不会生长，也就是对于每一个连续段，分别求出其会不会在从左到右和从右到左的风暴中会不会生长。显然其等价于其左边的连续段和其右边的连续段的高度是否大于它的高度，其可以用一个式子快速表示。

根据这个我们得出了一个连续段树木的高度变化规律，同时也可以得出相邻两段高度差的缩小规律，同时对于所有可能的情况分析可以发现相邻两段高度差一定是单调非增的。因此我们可以知道这一段高度差会在什么时候变为零，相邻的两端合并成一段。

具体来说，令两段在**从左到右**的风暴中高度的变化情况**是否不相同**记为 a，两段在**从右到左**的风暴中高度的变化情况**是否不相同**记为 b，记录高度差的时刻为 t，记录时高度差为 h，第 i 个时刻及以前从左到右的风暴有 l_i 场，从右到左的风暴有 r_i 场，则第 x 个时刻其高度差为 $\max(0,h-a(l_x-l_i)-b(r_x-r_i))$。我们可以根据这条式子来快速计算出相邻两段高度变为相等的时刻，分别记录 $a=1,b=0$，$a=0,b=1$，$a=1,b=1$ 三类情况即可，显然单次记录与查询可以做到 $O(1)$。

当两段合并时，只会修改这两段的信息以及更新这两段与其相邻段的高度差信息，单次修改的信息量是 $O(1)$ 的。因此总时间复杂度是 $O(n+m)$ 的。