Solution

witch

算法一

忘记部分分都是啥了，这题应该都会，直接讲正解。

按位独立考虑。

设 $f_{i,0/1,0/1,0/1}$ 表明以 i 为根的子树，i 到根异或和的当前位为 $0/1$，有没有选一个到根当前位为 0 的点，有没有选一个到根当前位为 1 的点。

枚举子树内的点转移。

用线段树合并优化。需要讨论一些下标的范围。

时间复杂度：$O(N \log N \log D)$。

数据是瞎造的，应该都能跑吧！

jigsaw

算法一

暴力计算。大概是 $O(N^3)$。

期望得分：12。

算法二

用取余加速暴力。大概是 $O(N^2 \log N)$。

期望得分：20。

算法三

直接递推即可。时间复杂度 $O(N^2)$。

期望得分：24。

算法四

考虑对于每个点对 (a,b) 算出有多少对 (i,j) 途径了这个状态。

从 (a,b) 倒推。其可以到达两个状态 $(a,a+b)$ 和 $(a+b,b)$。构成了一个树形结构，我们要求的就是整棵树的大小。

考虑将中间相同的元素缩起来，可以看作一个序列。一开始是 (a,b)，后来是 $(a,a+b,b)$。每一轮在两个数中间插入它们的和。

在每次新加入的位置统计贡献，那么所求即为：这个过程不断持续下去，得到的序列中 $\leq N$ 的数的个数。

观察一下得到的序列。我们可以把每个数表示为 $(x,y)=ax+by$。

得到如下性质：

任意时刻，对于两个相邻的数 $(a, b), (c, d)$ 有 $ad - bc = 1$。由归纳法可证。

所有满足 $ad - bc = 1$ 的非负整数 a, b, c, d 都会出现在序列中。由归纳法可证。

由裴蜀定理，我们可以得到：所有满足 $\gcd(i, j) = 1$ 的 i, j，$ai + bj$ 都会出现在序列中。

那么我们求的 $g(n, a, b) = \sum_{i=1}^{n} \sum_{j=1}^{n} [(i, j) = 1][ai + bj \le n]$。

所求即为 $f(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} g(n, i, j)$。可能会有一些常数的差别。

可以 $O(N^4)$ 计算，期望得分：4。

精细实现可以 $O(N^2 \log^2 N)$，期望得分：16。

考虑莫反。设 $G(n, a, b) = \sum_{i=1}^{n} \sum_{j=1}^{n} [ai + bj \le n] = \sum_{i=1}^{n} \lfloor \frac{n-ai}{b} \rfloor$。则 $g(n, a, b) = \sum_{i=1}^{n} \mu(i) G(\lfloor \frac{n}{i} \rfloor, a, b)$。

使用类欧计算 G，则可以做到 $O(N^3 \log N)$，期望得分：8。

带入 f，$f(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \mu(k) G(\lfloor \frac{n}{k} \rfloor, i, j)$。

交换求和顺序，得到：$f(n) = \sum_{k=1}^{n} \mu(k) \sum_{i=1}^{\lfloor \frac{n}{k} \rfloor} \sum_{j=1}^{\lfloor \frac{n}{k} \rfloor} G(\lfloor \frac{n}{k} \rfloor, i, j)$。

设 $F(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} G(n, i, j)$。

展开，$F(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \lfloor \frac{n-ki}{j} \rfloor = \sum_{k=1}^{n} \sum_{i=1}^{\lfloor \frac{n}{k} \rfloor} s(n - ki)$。其中 $s(i)$ 表示 $1 \sim i$ 的因子数之和。

$F(n)$ 可以 $O(N \log N)$ 计算，总时间复杂度：$O(N \log^2 N)$。期望得分：$48 \sim 64$。

$F(n) = \sum_{i=1}^{n} s(i) d(n - i)$，其中 $d(i)$ 表示 i 的因子数。

可以用 FFT 算出 F 的所有取值，总时间复杂度：$O(N \log N)$。常数很大。期望得分：$48 \sim 64$。

$F(n)$ 可以 $O(N)$ 计算，总时间复杂度：$O(N \log N)$。常数很小。期望得分：100。

如果使用整除分块常数大概还能少一半，不过不用也不会被卡。

pepper

灵感来源：[P7599 [APIO2021] 雨林跳跃]。~~好像做法没有任何关系~~。

算法一

使用 Floyd 计算最短路。时间复杂度：$O(N^3)$。

期望得分：4。

算法二

对每个点 BFS 求最短路。时间复杂度：$O(N^2)$。

期望得分：8。

算法三

处理出二维前缀和。时间复杂度：$O(N^2+Q)$。

期望得分：12。

算法四

建出笛卡尔树。则两种边分别连向 i 的父亲，以及 i 第一个拐弯的祖先。随机情况下树高为 $O(\log N)$，预处理出每个点对，二维数点一下即可。时间复杂度：$O(N\log^2 N)$。

结合之前的算法，期望得分：20。

算法五

$L_2=R_2$，可以尝试对每个节点维护出子树内每个点到自己的距离。

线段树合并。先把两个儿子的树合并，整体 $+1$。

需要进行一些修正。需要修正的是那些最短路中上一个节点在 x 的左儿子的右链中或者右儿子的左链中的点。可以暴力枚举两条链上的点，是 $O(N)$ 的。

这里定义一棵新树，在新树中每个节点以"第一个拐弯的祖先"为父亲（如果没有则父亲不变）。可以发现：新树中每个点的子树仍然是一个区间。那么修正的过程就是进行若干的区间加操作。

时间复杂度：$O(N\log N)$。

期望得分：24。

算法六

$L_1=R_1$。不会。

算法七

跑一遍算法五，然后分块。

考虑把整块的线段树进行合并。方法如下：每个点的根节点打 1 的标记，然后对线段树所有节点进行拓扑序 DP，求出每个节点的贡献，再差分。

时间复杂度：$O(NB\log N+\frac{N^2}{B}\log N)$。取 $B=\sqrt{N}$ 得到 $O(N\sqrt{N}\log N)$。

期望得分：64。

算法八

发现线段树上节点的贡献来源于若干区间加。考虑直接算每个区间加的贡献。DP 的对象从线段树换成笛卡尔树即可。

时间复杂度：$O(NB\log N+\frac{N^2}{B})$。取 $B=\sqrt{\frac{N}{\log N}}$ 得到 $O(N\sqrt{N\log N})$。

大数组的寻址很慢，改成离线会快很多（特意把空间卡了，就不会有人因为这个被卡常了！）

期望得分：100。

发现线段树合并是不必要的，离线 + 全局线段树即可做到 $O(N)$ 空间。