杂题选讲

Harry27182

目录

题意

题目大意

一张 n 个点 m 条边的无向图，只有 a,b 两种边权 $(a<b)$，对于每个 i，求图的所有最小生成树中 $1 \sim i$ 距离的最小值。

$n \le 70$ $m \le 200$。

分析性质

- 下文中把边权为 a 的边叫做轻边，边权为 b 的边叫做重边。

分析性质

- 下文中把边权为 a 的边叫做轻边，边权为 b 的边叫做重边。
- 轻边会把原图分成若干连通块，在最小生成树上，每个连通块内一定以轻边连接，然后用重边连接各连通块。

分析性质

- 下文中把边权为 a 的边叫做轻边，边权为 b 的边叫做重边。
- 轻边会把原图分成若干连通块，在最小生成树上，每个连通块内一定以轻边连接，然后用重边连接各连通块。
- 所以如果一条路径在最小生成树上，当且仅当其不会多次进入同一个轻边连通块。

分析性质

- 下文中把边权为 a 的边叫做轻边，边权为 b 的边叫做重边。
- 轻边会把原图分成若干连通块，在最小生成树上，每个连通块内一定以轻边连接，然后用重边连接各连通块。
- 所以如果一条路径在最小生成树上，当且仅当其不会多次进入同一个轻边连通块。
- 于是可以设计 dp，设 $dp_{s,i}$ 表示经过了 s 内连通块，当前在点 i 的最小距离。

分析性质

- 下文中把边权为 a 的边叫做轻边，边权为 b 的边叫做重边。
- 轻边会把原图分成若干连通块，在最小生成树上，每个连通块内一定以轻边连接，然后用重边连接各连通块。
- 所以如果一条路径在最小生成树上，当且仅当其不会多次进入同一个轻边连通块。
- 于是可以设计 dp，设 $dp_{s,i}$ 表示经过了 s 内连通块，当前在点 i 的最小距离。
- 注意转移是可以循环转移的，所以使用 dijkstra 进行转移，复杂度 $O(2^n m \log(2^n m))$。

正解

- 考虑进一步优化，我们有结论：不需要考虑大小 ≤ 3 的连通块。

正解

- 考虑进一步优化，我们有结论：不需要考虑大小 ≤ 3 的连通块。
- 因为你发现你多次进入这种连通块本身就不会优，走内部的边是比一进一出权值更小的。

正解

- 考虑进一步优化，我们有结论：不需要考虑大小 ≤ 3 的连通块。
- 因为你发现你多次进入这种连通块本身就不会优，走内部的边是比一进一出权值更小的。
- 所以只需要记录大小 ≥ 4 的连通块，复杂度 $O(2^{\frac{n}{4}} m \log(2^{\frac{n}{4}} m))$。

题意

题目大意

有一棵 n 个点的树，可以进行以下询问 query(u,V)，会返回 $\sum_{x\in V} dis(u,x)$，你需要在 A 次询问内找到树上的所有边，同时需要保证 $\sum |V| \le B$。

$n \le 1000, A = 8500, B = 3 \times 10^5$。

部分分

- 点分治，重心是到所有点距离之和最小的点，可以通过这种方式找到重心。

部分分

- 点分治，重心是到所有点距离之和最小的点，可以通过这种方式找到重心。
- 然后询问每个点的 dep，对所有 $dep \neq 1$ 的点去找它们在哪个子树，可以二分判断当前点是否在一个前缀的子树里。

部分分

- 点分治，重心是到所有点距离之和最小的点，可以通过这种方式找到重心。
- 然后询问每个点的 dep，对所有 $dep \neq 1$ 的点去找它们在哪个子树，可以二分判断当前点是否在一个前缀的子树里。
- $A = O(n \log^2 n), B = O(n^2)$。

正解

- 首先随便定个根，然后求出每个点的 dep。

正解

- 首先随便定个根，然后求出每个点的 dep。
- 考虑逐层处理相邻两层之间的连边。设集合 A 的深度是集合 B 的深度 -1。

正解

- 首先随便定个根，然后求出每个点的 dep。
- 考虑逐层处理相邻两层之间的连边。设集合 A 的深度是集合 B 的深度 -1。
- 我们将 A 随机取出一半作为 S 集合，然后我们发现有一个必要条件，如果 v 是 u 的父亲，那么一定有 $\sum_{x\in S} dis(x,v)+|S|=\sum_{x\in S} dis(x,u)$。

正解

- 首先随便定个根，然后求出每个点的 dep。
- 考虑逐层处理相邻两层之间的连边。设集合 A 的深度是集合 B 的深度 -1。
- 我们将 A 随机取出一半作为 S 集合，然后我们发现有一个必要条件，如果 v 是 u 的父亲，那么一定有 $\sum_{x\in S} dis(x,v)+|S|=\sum_{x\in S} dis(x,u)$。
- 这个条件显然不充分，但可以将 B 中元素划分成两份，父亲在 S 中的和不在 S 中的。递归处理即可。

正解

- 首先随便定个根，然后求出每个点的 dep。
- 考虑逐层处理相邻两层之间的连边。设集合 A 的深度是集合 B 的深度 -1。
- 我们将 A 随机取出一半作为 S 集合，然后我们发现有一个必要条件，如果 v 是 u 的父亲，那么一定有 $\sum_{x\in S} dis(x,v)+|S|=\sum_{x\in S} dis(x,u)$。
- 这个条件显然不充分，但可以将 B 中元素划分成两份，父亲在 S 中的和不在 S 中的。递归处理即可。
- $A=O(n\log n),B=O(n^2)$。

1. CF1149D
2. qoj5015
3. luogu P7216
4. qoj7839
5. qoj5022

题意

题目大意

给定 n 个矩形，需要求出一组 k 个点，满足每个矩形内都有至少一个点，输出任意一组解即可。保证有解。

$n \leq 2 \times 10^5, k \leq 4$。

$k \leq 3$

- 如果可以有一条横线或竖线穿过所有矩形，按照一维的问题直接贪心即可。

$k \leq 3$

- 如果可以有一条横线或竖线穿过所有矩形，按照一维的问题直接贪心即可。
- 否则我们一定要覆盖最小右边界，最大左边界，最小上边界，最大下边界。

$k \le 3$

- 如果可以有一条横线或竖线穿过所有矩形，按照一维的问题直接贪心即可。
- 否则我们一定要覆盖最小右边界，最大左边界，最小上边界，最大下边界。
- 当 $k \le 3$ 时，由于三个点覆盖四条线，所以必须有一个点在两条线的交点上，暴搜即可，复杂度 $O(4^k n)$。

正解

- 当 $k = 4$ 时，先跑一下上面的做法。如果跑不出来说明四个点都在边界上且不在端点上。

正解

- 当 $k=4$ 时，先跑一下上面的做法。如果跑不出来说明四个点都在边界上且不在端点上。
- 显然每个矩形都与至少一条边界相交。如果有交边界数 ≥ 3，显然无论怎么选择都合法。

正解

- 当 $k=4$ 时，先跑一下上面的做法。如果跑不出来说明四个点都在边界上且不在端点上。
- 显然每个矩形都与至少一条边界相交。如果有交边界数 ≥ 3，显然无论怎么选择都合法。
- 对于只与一条边界有交的矩形，是限制这条边界的点在一段区间内。

正解

- 当 $k=4$ 时，先跑一下上面的做法。如果跑不出来说明四个点都在边界上且不在端点上。
- 显然每个矩形都与至少一条边界相交。如果有交边界数 ≥ 3，显然无论怎么选择都合法。
- 对于只与一条边界有交的矩形，是限制这条边界的点在一段区间内。
- 对于与两条边界有交的矩形，我们可以使用 2-SAT，对于这个矩形建立两个状态分别表示这个矩形依靠哪个边界上的点消掉。

正解

- 当 $k = 4$ 时，先跑一下上面的做法。如果跑不出来说明四个点都在边界上且不在端点上。
- 显然每个矩形都与至少一条边界相交。如果有交边界数 ≥ 3，显然无论怎么选择都合法。
- 对于只与一条边界有交的矩形，是限制这条边界的点在一段区间内。
- 对于与两条边界有交的矩形，我们可以使用 2-SAT，对于这个矩形建立两个状态分别表示这个矩形依靠哪个边界上的点消掉。
- 限制就是对于两个矩形，它们如果它们在一个边界上的区间不相交，那么就不能同时选这个边界。前缀和优化建图即可。复杂度 $O(4^k + n\log n)$。

1. CF1149D
2. qoj5015
3. luogu P7216
4. qoj7839
5. qoj5022

题意

题目大意

给定一棵 n 个点的树。点带权 w_i，初始为 0。定义一段区间 $[l,r]$ 的最小虹为 $[l,r]$ 内所有点形成的虚树。
需要维护以下两种操作：将 $[l,r]$ 的虹对应的点的 w_i 都 $+1$，求 $(\sum_{i=l}^{r} 19901991^{z_{\gcd(i,u)} w_i}) \bmod 20242024$，$u$ 给定。
$n,q \leq 80000$，l,r 随机生成。

注意力

- 注意到 $19901991^2 \equiv 1 \pmod{20242024}$。

正解

- 我们只需要关注 $z_{\gcd(i,u)}, w_i$ 在 $\bmod 2$ 意义下的值。

正解

- 我们只需要关注 $z_{\gcd(i,u)}, w_i$ 在 $\bmod 2$ 意义下的值。
- 用 bitset 维护 W, B 序列的值，首先考虑 W 如何求。

正解

- 我们只需要关注 $z_{\gcd(i,u)}, w_i$ 在 mod2 意义下的值。
- 用 bitset 维护 W, B 序列的值，首先考虑 W 如何求。
- 我们需要求出 $[l, r]$ 区间对应的最小虹 S，然后让 $W \oplus = S$。

正解

- 我们只需要关注 $z_{\gcd(i,u)}, w_i$ 在 mod2 意义下的值。
- 用 bitset 维护 W, B 序列的值，首先考虑 W 如何求。
- 我们需要求出 $[l, r]$ 区间对应的最小虹 S，然后让 $W \oplus = S$。
- 设 P_u 表示 u 到根节点的路径集合，那么 $S' = (P_l \cup P_{l+1} \dots \cup P_r), S = C_{S'} P_{lca(l,l+1,\dots,r)}$。

正解

- 分块，预处理每个整块的每个前缀，后缀，若干个整块的 $\bigcup P_i$。如果 l, r 在不同块即可直接查询，否则暴力。

正解

- 分块，预处理每个整块的每个前缀，后缀，若干个整块的 $\bigcup P_i$。如果 l, r 在不同块即可直接查询，否则暴力。
- 由于数据随机，暴力的次数不会很多，取 $B = \sqrt{n}$，可以平衡得到 $O(n\sqrt{n})$。

正解

- 分块，预处理每个整块的每个前缀，后缀，若干个整块的 $\bigcup P_i$。如果 l, r 在不同块即可直接查询，否则暴力。
- 由于数据随机，暴力的次数不会很多，取 $B = \sqrt{n}$，可以平衡得到 $O(n\sqrt{n})$。
- 这样就可以求出 W 了，考虑如何求 B。

正解

- 分块，预处理每个整块的每个前缀，后缀，若干个整块的 $\bigcup P_i$。如果 l, r 在不同块即可直接查询，否则暴力。
- 由于数据随机，暴力的次数不会很多，取 $B = \sqrt{n}$，可以平衡得到 $O(n\sqrt{n})$。
- 这样就可以求出 W 了，考虑如何求 B。
- 直接暴力递推即可，每次去掉最小质因子去暴力改变所有需要变化的位，打表发现操作次数在 $X = 43474197$ 以内。

正解

- 分块，预处理每个整块的每个前缀，后缀，若干个整块的 $\bigcup P_i$。如果 l, r 在不同块即可直接查询，否则暴力。
- 由于数据随机，暴力的次数不会很多，取 $B = \sqrt{n}$，可以平衡得到 $O(n\sqrt{n})$。
- 这样就可以求出 W 了，考虑如何求 B。
- 直接暴力递推即可，每次去掉最小质因子去暴力改变所有需要变化的位，打表发现操作次数在 $X = 43474197$ 以内。
- 复杂度 $O(n\sqrt{n} + \frac{n^2}{w} + X)$，注意很多东西都要离散化来减小 $O(\frac{n^2}{w})$ 的空间常数。

1. CF1149D
2. qoj5015
3. luogu P7216
4. qoj7839
5. qoj5022

题意

题目大意

给定一个长度为 n 的序列 $a_1 \sim a_n$，进行 q 次以下两种操作：
将 $a[l \sim r]$ 替换为它的异或差分，即令 $b_i = a_i \oplus a_{i-1}$，然后令 $a_i = b_i$。查询 a_{pos} 的值。
$n \leq 2.5 \times 10^5, q, \leq 10^5, a_i \leq 2^{30}$
特殊性质挺重要的，去原题看特殊性质。

特殊性质

- 对于 CD 性质的问题，可以转化为简单的网格图路径计数，组合数即可。复杂度 $O(c(n+q))$。

特殊性质

- 对于 CD 性质的问题，可以转化为简单的网格图路径计数，组合数即可。复杂度 $O(c(n+q))$。
- 对于 DE 性质的问题，虽然起点变多了，但终点只在最后一行。

特殊性质

- 对于 CD 性质的问题，可以转化为简单的网格图路径计数，组合数即可。复杂度 $O(c(n+q))$。
- 对于 DE 性质的问题，虽然起点变多了，但终点只在最后一行。
- 发现我们关注的只是这个组合数的奇偶性，考虑 Kummer 定理。

特殊性质

- 对于 CD 性质的问题，可以转化为简单的网格图路径计数，组合数即可。复杂度 $O(c(n+q))$。
- 对于 DE 性质的问题，虽然起点变多了，但终点只在最后一行。
- 发现我们关心的只是这个组合数的奇偶性，考虑 Kummer 定理。
- 我们关心的是 $\binom{q}{x_t - x_s}$，所以 $x_t - x_s$ 必须是 q 的子集，类似高维前缀和即可求出最后能贡献到 x_t 上的数异或和。复杂度 $O(n\log n+q)$。

正解

- 考虑分块，每 $B = 2^P$ 个位置一块，考虑计算块内贡献和块之间的贡献。

正解

- 考虑分块，每 $B = 2^P$ 个位置一块，考虑计算块内贡献和块之间的贡献。
- 修改操作肯定是散块重构，整块打 tag，考虑一个块 pushdown 的过程。

正解

- 考虑分块，每 $B = 2^P$ 个位置一块，考虑计算块内贡献和块之间的贡献。
- 修改操作肯定是散块重构，整块打 tag，考虑一个块 pushdown 的过程。
- 这个过程就是性质 DE 的问题，高维前缀和处理即可。

正解

- 考虑分块，每 $B = 2^P$ 个位置一块，考虑计算块内贡献和块之间的贡献。
- 修改操作肯定是散块重构，整块打 tag，考虑一个块 pushdown 的过程。
- 这个过程就是性质 DE 的问题，高维前缀和处理即可。
- 考虑块外对块内的贡献，我们需要记录每一次修改之前的 a_{l-1}，从晚到早第 i 次修改前的记为 b_i，然后考虑它们对当前块内的贡献。

正解

- 考虑分块，每 $B = 2^P$ 个位置一块，考虑计算块内贡献和块之间的贡献。
- 修改操作肯定是散块重构，整块打 tag，考虑一个块 pushdown 的过程。
- 这个过程就是性质 DE 的问题，高维前缀和处理即可。
- 考虑块外对块内的贡献，我们需要记录每一次修改之前的 a_{l-1}，从晚到早第 i 次修改前的记为 b_i，然后考虑它们对当前块内的贡献。
- 容易发现 b_k 对于 a'_i 的贡献就是 $\binom{k-1}{i-l}$，考虑再对这个式子用一次 Kummer 定理，我们做一个高维后缀和，处理出来 $c_i = \sum_{i \subseteq j} b_j$，然后将 c_i 贡献到 a'_{l+i}。

正解

- 最后一个问题在于如何快速求出 b_i。

正解

- 最后一个问题在于如何快速求出 b_i。
- 我们类似地考虑 b_i 从哪些东西转移过来，可以每次重构块时预处理出来如果经过 i 次差分，哪些能贡献到 a_r，这个也是一个高维前缀和。

正解

- 最后一个问题在于如何快速求出 b_i。
- 我们类似地考虑 b_i 从哪些东西转移过来，可以每次重构块时预处理出来如果经过 i 次差分，哪些能贡献到 a_r，这个也是一个高维前缀和。
- 但还有块外的贡献，注意到块外的转移过来一定会经过上一个块的右端点，而上一个块的右端点到这个块的右端点的距离是已知的，就是 B。

正解

- 最后一个问题在于如何快速求出 b_i。
- 我们类似地考虑 b_i 从哪些东西转移过来，可以每次重构块时预处理出来如果经过 i 次差分，哪些能贡献到 a_r，这个也是一个高维前缀和。
- 但还有块外的贡献，注意到块外的转移过来一定会经过上一个块的右端点，而上一个块的右端点到这个块的右端点的距离是已知的，就是 B。
- 所以当且仅当 $i-j\equiv B$ 时，b'_j 能转移到 b_i，记录一下即可。

正解

- 最后一个问题在于如何快速求出 b_i。
- 我们类似地考虑 b_i 从哪些东西转移过来，可以每次重构块时预处理出来如果经过 i 次差分，哪些能贡献到 a_r，这个也是一个高维前缀和。
- 但还有块外的贡献，注意到块外的转移过来一定会经过上一个块的右端点，而上一个块的右端点到这个块的右端点的距离是已知的，就是 B。
- 所以当且仅当 $i-j \equiv B$ 时，b'_j 能转移到 b_i，记录一下即可。
- 复杂度 $O(qB\log B+\frac{nq}{B})$，平衡得到复杂度为 $O(q\sqrt{n\log n})$。