杂题选讲

Harry27182

2025 年 1 月 27 日

luogu P9339

- 有 n 种饼干，第 i 种有 a_i 个。需要把饼干装入盒子里，每个盒子需要满足不能装两个同种饼干，且饼干数必须是 $b_1 \sim b_m$ 之一。求最少需要多少盒子。
- $n \le 15000$。

- 饼干和盒子之间能形成一种匹配关系。将盒子看做左部点，饼干看做右部点。设第 i 个盒子的大小为 w_i，条件转化为 $\sum w_i = \sum a_i$ 且二分图具有完美匹配。

- 饼干和盒子之间能形成一种匹配关系。将盒子看做左部点，饼干看做右部点。设第 i 个盒子的大小为 w_i，条件转化为 $\sum w_i = \sum a_i$ 且二分图具有完美匹配。
- 二分图完美匹配要考虑 Hall 定理。对于每个左部点集合 S，都需要满足其对应匹配的右部点集合大小大于 $\sum_{i \in S} w_i$。

luogu P9339

- 饼干和盒子之间能形成一种匹配关系。将盒子看做左部点，饼干看做右部点。设第 i 个盒子的大小为 w_i，条件转化为 $\sum w_i = \sum a_i$ 且二分图具有完美匹配。
- 二分图完美匹配要考虑 Hall 定理。对于每个左部点集合 S，都需要满足其对应匹配的右部点集合大小大于 $\sum_{i \in S} w_i$。
- 对应的右部点集合大小就是 $\sum_{i=1}^{n} \min(a_i, |S|)$。

- 容易发现，对应右部点集合大小只和 $|S|$ 有关，所以只需要考虑按照 w_i 从大到小排序后每个前缀合法即可。

- 容易发现，对应右部点集合大小只和 $|S|$ 有关，所以只需要考虑按照 w_i 从大到小排序后每个前缀合法即可。
- 设 $dp_{i,j,k}$ 表示考虑了前 i 个大小的盒子，填了 j 个盒子，总大小为 k 是否合法。转移类似完全背包。复杂度 $O((\sum a)^3)$。

- 容易发现，对应右部点集合大小只和 $|S|$ 有关，所以只需要考虑按照 w_i 从大到小排序后每个前缀合法即可。
- 设 $dp_{i,j,k}$ 表示考虑了前 i 个大小的盒子，填了 j 个盒子，总大小为 k 是否合法。转移类似完全背包。复杂度 $O((\sum a)^3)$。
- 考虑优化，由于 b_i 互不相同，所有 j 这一维的大小是调和级数，同时使用 bitset 优化即可做到 $O(\frac{(\sum a)^2 \log \sum a}{w})$。

AGC057D

- 给定一个正整数 S，称一个正整数集合 A 是好的，当且仅当满足以下条件：
- A 中元素互不相同且在 $[1, S)$ 中且对于任意非负整数序列 x_i 满足 $\sum a_i x_i \neq S$。
- 求元素数量最多且排序后字典序最小的集合 A。给定 k，求 A 集合中第 k 小的元素。
- $T \leq 1000, S \leq 10^{18}$。

- 注意到 $|A| = \lfloor \frac{S-1}{2} \rfloor$。上界是因为 i 和 $S-i$ 只能选一个，下界是因为 $\{\lfloor \frac{S-1}{2} \rfloor + 1, ..., S-1\}$。

- 注意到 $|A| = \lfloor \frac{S-1}{2} \rfloor$。上界是因为 i 和 $S-i$ 只能选一个，下界是因为 $\{\lfloor \frac{S-1}{2} \rfloor + 1, ..., S-1\}$。
- 继续观察性质，如果 $a, b \in A$，那么 $a + b \in A$，因为 $S - a - b \notin A$。

AGC057D

- 注意到 $|A| = \lfloor \frac{S-1}{2} \rfloor$。上界是因为 i 和 $S-i$ 只能选一个，下界是因为 $\{\lfloor \frac{S-1}{2} \rfloor + 1, ..., S-1\}$。
- 继续观察性质，如果 $a, b \in A$，那么 $a + b \in A$，因为 $S - a - b \notin A$。
- 从小到大贪心加入。能加入的第一个数 m 应当满足 m 不是 S 的因子，且 $1 \sim m-1$ 都是 S 的因子。有 $m \leq 43$。

AGC057D

- 判断一些数的线性组合能否得到 S 可以考虑同余最短路，记录 f_i 表示能凑出来的最小的 $x \bmod m = i$ 的 x。

- 判断一些数的线性组合能否得到 S 可以考虑同余最短路，记录 f_i 表示能凑出来的最小的 $x \bmod m = i$ 的 x。
- 我们只需要考虑主动加入的数，不需要考虑因为性质被动加入的数。对于每个 i，主动加入的显然只有一个点，所以加入的点数量为 $O(m)$ 的。每次加入后暴力更新 f_i，复杂度 $O(m^3)$。

- 判断一些数的线性组合能否得到 S 可以考虑同余最短路，记录 f_i 表示能凑出来的最小的 $x \bmod m = i$ 的 x。
- 我们只需要考虑主动加入的数，不需要考虑因为性质被动加入的数。对于每个 i，主动加入的显然只有一个点，所以加入的点数量为 $O(m)$ 的。每次加入后暴力更新 f_i，复杂度 $O(m^3)$。
- 每次需要找到最小的可以加入的数，假设对于一个 i 需要加入 $qm+i$，那么需要对于任意 $j \in [1, m-1]$ 满足 $j(qm+i) + f_{(S-ij) \bmod m} > S$，可以求出 q 的最小值。

- 判断一些数的线性组合能否得到 S 可以考虑同余最短路，记录 f_i 表示能凑出来的最小的 $x \bmod m = i$ 的 x。
- 我们只需要考虑主动加入的数，不需要考虑因为性质被动加入的数。对于每个 i，主动加入的显然只有一个点，所以加入的点数量为 $O(m)$ 的。每次加入后暴力更新 f_i，复杂度 $O(m^3)$。
- 每次需要找到最小的可以加入的数，假设对于一个 i 需要加入 $qm+i$，那么需要对于任意 $j \in [1, m-1]$ 满足 $j(qm+i) + f_{(S-ij) \bmod m} > S$，可以求出 q 的最小值。
- 查询二分答案即可。复杂度 $O(T(m^3 + m\log S))$。

CF1770F

- 给定 n, x, y，对于所有 $\sum_{i=1}^{n} a_i = x, OR_{i=1}^{n} a_i = y$ 求 $\bigoplus_{i=1}^{n} a_i$ 的异或和。
- $n \le 2^{40}, x \le 2^{60}, y \le 2^{20}$。

CF1770F

- 根据对称性，n 为偶数时答案为 0，n 为奇数是答案为 $\oplus a_1$。

- 根据对称性，n 为偶数时答案为 0，n 为奇数是答案为 $\oplus a_1$。
- 发现 y 很小，考虑对 y 进行容斥。记 f_s 表示 $OR_{i=1}^{n} a_i$ 是 s 的子集的答案。那么有 $f_s = \sum_{p \in S} 2^p \times (([\sum_{i=1}^{n} a_i = x - 2^p][a_1 \subset s - 2^p][a_2 \subset s] ... [a_n \subset s]) \bmod 2)$。

CF1770F

- 根据对称性，n 为偶数时答案为 0，n 为奇数是答案为 $\oplus a_1$。
- 发现 y 很小，考虑对 y 进行容斥。记 f_s 表示 $OR_{i=1}^{n} a_i$ 是 s 的子集的答案。那么有 $f_s = \sum_{p\in S} 2^p \times (([\sum_{i=1}^{n} a_i = x - 2^p][a_1 \subset s - 2^p][a_2 \subset s]...[a_n \subset s]) \bmod 2)$。
- 考虑逆用 kummer 定理，
  $f_s = \sum_{p\in S} 2^p \times (([\sum_{i=1}^{n} a_i = x - 2^p]\binom{s-2^p}{a_1}\binom{s}{a_2}...\binom{s}{a_n}) \bmod 2)$。

- 根据对称性，n 为偶数时答案为 0，n 为奇数是答案为 $\oplus a_1$。
- 发现 y 很小，考虑对 y 进行容斥。记 f_s 表示 $OR_{i=1}^{n} a_i$ 是 s 的子集的答案。那么有 $f_s = \sum_{p \in S} 2^p \times (([\sum_{i=1}^{n} a_i = x - 2^p][a_1 \subset s - 2^p][a_2 \subset s]...[a_n \subset s]) \bmod 2)$。
- 考虑逆用 kummer 定理，$f_s = \sum_{p \in S} 2^p \times (([\sum_{i=1}^{n} a_i = x - 2^p]\binom{s-2^p}{a_1}\binom{s}{a_2}...\binom{s}{a_n}) \bmod 2)$。
- 也就是 $f_s = \sum_{p \in S} 2^p \times (\binom{ns-2^p}{x-2^p} \bmod 2) = \sum_{p \in S}[x - 2^p \subset ns - 2^p]2^p$。

- 根据对称性，n 为偶数时答案为 0，n 为奇数是答案为 $\oplus a_1$。
- 发现 y 很小，考虑对 y 进行容斥。记 f_s 表示 $OR_{i=1}^{n} a_i$ 是 s 的子集的答案。那么有 $f_s = \sum_{p\in S} 2^p \times (([\sum_{i=1}^{n} a_i = x - 2^p][a_1 \subset s - 2^p][a_2 \subset s]\ldots[a_n \subset s]) \bmod 2)$。
- 考虑逆用 kummer 定理，$f_s = \sum_{p\in S} 2^p \times (([\sum_{i=1}^{n} a_i = x - 2^p]\binom{s-2^p}{a_1}\binom{s}{a_2}\ldots\binom{s}{a_n}) \bmod 2)$。
- 也就是 $f_s = \sum_{p\in S} 2^p \times (\binom{ns-2^p}{x-2^p} \bmod 2) = \sum_{p\in S}[x - 2^p \subset ns - 2^p]2^p$。
- 对于单个 s 可以 $O(\log y)$ 计算 f_s，容斥回去复杂度为 $O(y \log y)$。

qoj5018

- 有一个可重 S 内元素 $\le M = 2^{21}$，集合分成两部分，小 A 和小 B 分别知道其中一部分，每个部分中每个数至多出现一次。求集合内第 k 小的数，要求信息传输次数 ≤ 90。

qoj5018

- 可以通过二分答案做到 $\frac{1}{2}\log^2 M$ 次传递。

- 可以通过二分答案做到 $\frac{1}{2}\log^2 M$ 次传递。
- 将问题简化。不妨设 $a = |A|, b = |B|, a < b$，对 c 的大小分类讨论。

- 可以通过二分答案做到 $\frac{1}{2}\log^2 M$ 次传递。
- 将问题简化。不妨设 $a=|A|, b=|B|, a<b$，对 c 的大小分类讨论。
- $1 \le c \le a$，等价于在 $A_{1\sim c}, B_{1\sim c}$ 求出第 c 大。

qoj5018

- 可以通过二分答案做到 $\frac{1}{2}\log^2 M$ 次传递。
- 将问题简化。不妨设 $a=|A|,b=|B|,a<b$，对 c 的大小分类讨论。
- $1\le c\le a$，等价于在 $A_{1\sim c},B_{1\sim c}$ 求出第 c 大。
- $a<c\le b$，等价于在 $\{0\}\cup A_{1\sim a},B_{c-a\sim c}$ 中求出第 $a+2$ 大。

- 可以通过二分答案做到 $\frac{1}{2}\log^2 M$ 次传递。
- 将问题简化。不妨设 $a = |A|, b = |B|, a < b$，对 c 的大小分类讨论。
- $1 \le c \le a$，等价于在 $A_{1\sim c}, B_{1\sim c}$ 求出第 c 大。
- $a < c \le b$，等价于在 $\{0\} \cup A_{1\sim a}, B_{c-a\sim c}$ 中求出第 $a+2$ 大。
- $b < c \le a+b$，等价于在 $A_{c-b\sim a}, B_{c-a\sim b}$ 中求出第 $a+b-c+2$ 大。

- 可以通过二分答案做到 $\frac{1}{2}\log^2 M$ 次传递。
- 将问题简化。不妨设 $a=|A|, b=|B|, a<b$，对 c 的大小分类讨论。
- $1 \le c \le a$，等价于在 $A_{1\sim c}, B_{1\sim c}$ 求出第 c 大。
- $a < c \le b$，等价于在 $\{0\} \cup A_{1\sim a}, B_{c-a\sim c}$ 中求出第 $a+2$ 大。
- $b < c \le a+b$，等价于在 $A_{c-b\sim a}, B_{c-a\sim b}$ 中求出第 $a+b-c+2$ 大。
- 上述问题均满足 $|A|=|B|, c=|A|(+1)$，这样就简化了问题。

- 考虑传输 A 的低中位数和 B 的高中位数。如果 $A_m < B_M$，那么可以删去 $A_{1\sim m-1}, B_{M+1\sim|B|}$。反之同理。这样就能通过一次操作将问题规模变成一半，传递次数仍然是 $O(\log^2 n)$。

qoj5018

- 考虑传输 A 的低中位数和 B 的高中位数。如果 $A_m < B_M$，那么可以删去 $A_{1\sim m-1}, B_{M+1\sim|B|}$。反之同理。这样就能通过一次操作将问题规模变成一半，传递次数仍然是 $O(\log^2 n)$。
- 但是这个做法相比二分我们只需要比较两个数的大小，那么可以利用二进制的性质从高位向低位比较。

qoj5018

- 考虑传输 A 的低中位数和 B 的高中位数。如果 $A_m < B_M$，那么可以删去 $A_{1\sim m-1}, B_{M+1\sim|B|}$。反之同理。这样就能通过一次操作将问题规模变成一半，传递次数仍然是 $O(\log^2 n)$。
- 但是这个做法相比二分我们只需要比较两个数的大小，那么可以利用二进制的性质从高位向低位比较。
- 如果 $A_m, B_M \in [L, R]$，那么答案也应该 $\in [L, R]$。

- 考虑传输 A 的低中位数和 B 的高中位数。如果 $A_m < B_M$，那么可以删去 $A_{1\sim m-1}, B_{M+1\sim|B|}$。反之同理。这样就能通过一次操作将问题规模变成一半，传递次数仍然是 $O(\log^2 n)$。
- 但是这个做法相比二分我们只需要比较两个数的大小，那么可以利用二进制的性质从高位向低位比较。
- 如果 $A_m, B_M \in [L, R]$，那么答案也应该 $\in [L, R]$。
- 所以按照上述做法从高位向低位比较，如果不同就删去一半，否则可以考虑值域上下一位。也就是每比较一次，可以将值域减半或者集合大小减半，传递次数 $4\log M$。

qoj5029

- 有一棵未知的树，保证树的大小为奇数，你需要找到这棵树重心的编号。
- 你可以询问 query(x,y,z) 表示询问三个点中相对顺序中间的一个点，如果不存在一条简单路径经过三个点返回 0。
- $T = 100, n = 30000$，询问次数限制 5×10^7。

qoj5029

- 对于链的情况，可以先找到链的一个端点，然后 nth$_e lement$

- 对于链的情况，可以先找到链的一个端点，然后 nth$_e lement$
- 对于前者，不断 query(x,y,z) 将返回值从备选答案集合中删除即可。对于后者，在确定端点后是可以通过一次 query 操作比较两个数大小的。

qoj5029

- 对于链的情况，可以先找到链的一个端点，然后 nth$_e$*lement*
- 对于前者，不断 query(x,y,z) 将返回值从备选答案集合中删除即可。对于后者，在确定端点后是可以通过一次 query 操作比较两个数大小的。
- 对于树的情况，check 一个点是否是重心可以先摩尔投票投出来绝对众数，再判断两个点是否位于同一子树即可。

- 对于链的情况，可以先找到链的一个端点，然后 $\mathrm{nth}_e lement$
- 对于前者，不断 query(x,y,z) 将返回值从备选答案集合中删除即可。对于后者，在确定端点后是可以通过一次 query 操作比较两个数大小的。
- 对于树的情况，check 一个点是否是重心可以先摩尔投票投出来绝对众数，再判断两个点是否位于同一子树即可。
- 考虑将问题转化到链上。由于重心的性质，我们每次随机两个点，期望随机 $O(1)$ 次就能使得重心在这两个点形成的链上。

- 对于链的情况，可以先找到链的一个端点，然后 nth$_e$*lement*
- 对于前者，不断 query(x,y,z) 将返回值从备选答案集合中删除即可。对于后者，在确定端点后是可以通过一次 query 操作比较两个数大小的。
- 对于树的情况，check 一个点是否是重心可以先摩尔投票投出来绝对众数，再判断两个点是否位于同一子树即可。
- 考虑将问题转化到链上。由于重心的性质，我们每次随机两个点，期望随机 $O(1)$ 次就能使得重心在这两个点形成的链上。
- 问题转化为找到带权链的中点，并 check 其是否是重心。

qoj5029

- 对于链的情况，可以先找到链的一个端点，然后 nth$_{e}lement$

qoj5029

- 对于链的情况，可以先找到链的一个端点，然后 $\text{nth}_e lement$
- 对于前者，不断 query(x,y,z) 将返回值从备选答案集合中删除即可。对于后者，在确定端点后是可以通过一次 query 操作比较两个数大小的。

qoj5029

- 对于链的情况，可以先找到链的一个端点，然后 $nth_element$
- 对于前者，不断 query(x,y,z) 将返回值从备选答案集合中删除即可。对于后者，在确定端点后是可以通过一次 query 操作比较两个数大小的。
- 对于树的情况，check 一个点是否是重心可以先摩尔投票投出来绝对众数，再判断两个点是否位于同一子树即可。

- 对于链的情况，可以先找到链的一个端点，然后 $\mathrm{nth}_e lement$
- 对于前者，不断 query(x,y,z) 将返回值从备选答案集合中删除即可。对于后者，在确定端点后是可以通过一次 query 操作比较两个数大小的。
- 对于树的情况，check 一个点是否是重心可以先摩尔投票投出来绝对众数，再判断两个点是否位于同一子树即可。
- 考虑将问题转化到链上。由于重心的性质，我们每次随机两个点，期望随机 $O(1)$ 次就能使得重心在这两个点形成的链上。

- 对于链的情况，可以先找到链的一个端点，然后 nth$_e$*lement*
- 对于前者，不断 query(x,y,z) 将返回值从备选答案集合中删除即可。对于后者，在确定端点后是可以通过一次 query 操作比较两个数大小的。
- 对于树的情况，check 一个点是否是重心可以先摩尔投票投出来绝对众数，再判断两个点是否位于同一子树即可。
- 考虑将问题转化到链上。由于重心的性质，我们每次随机两个点，期望随机 $O(1)$ 次就能使得重心在这两个点形成的链上。
- 问题转化为找到带权链的中点，并 check 其是否是重心。

qoj5029

- 如果套用不带权的做法或者排序，询问次数都是 $O(n\log n)$。

- 如果套用不带权的做法或者排序，询问次数都是 $O(n \log n)$。
- 考虑带权随机，在当前应该考虑的集合内随机一个全局的点，扫一遍目前剩余的链找到它位于哪个点的子树里。

- 如果套用不带权的做法或者排序，询问次数都是 $O(n \log n)$。
- 考虑带权随机，在当前应该考虑的集合内随机一个全局的点，扫一遍目前剩余的链找到它位于哪个点的子树里。
- 对于该链上的点判断其是否是链的中点，如果不是就递归一侧，否则进行摩尔投票。

qoj5029

- 如果套用不带权的做法或者排序，询问次数都是 $O(n\log n)$。
- 考虑带权随机，在当前应该考虑的集合内随机一个全局的点，扫一遍目前剩余的链找到它位于哪个点的子树里。
- 对于该链上的点判断其是否是链的中点，如果不是就递归一侧，否则进行摩尔投票。
- 这样询问次数就是期望 $O(n)$ 的了。

- 给定一棵树 $T=(V,E)$，定义一个点集 V_0 对应的 $f_E(V_0)$ 表示 V_0 内点形成虚树的边集，$f_V(V_0)$ 表示点集。
- 有一个物品集合 S，需要给每个物品 i 挂在一个位置 a_i。
- 有 q 个限制，每个形如 (V_i,S_i)，需要满足 $f(V_i)$ 和 $f(\{a_k|k\in S_j\})$ 无交。
- 还有 $m=|S|$ 个限制，$a_i\in f_V(A_i)$。判断是否合法并构造方案。
- $n,m\le 2000,q,\sum|V_j|,\sum|S_j|\le 5\times 10^5,|V_j|,|S_j|\le 50$。

qoj7869

- 对于链的情况，限制可以用以下形式刻画：$a_i = a_j$ 或 $\max(a_i, a_j) \le l$ 或 $\min(a_i, a_j) \ge r$。

- 对于链的情况，限制可以用以下形式刻画：$a_i = a_j$ 或 $\max(a_i, a_j) \le l$ 或 $\min(a_i, a_j) \ge r$。
- 令 $f_{i,j} = [a_i < j]$，那么以上三种限制都可以用 2-SAT 刻画。并且 2-SAT 的方案可以对应一组唯一的构造方案。

- 对于链的情况，限制可以用以下形式刻画：$a_i = a_j$ 或 $\max(a_i, a_j) \le l$ 或 $\min(a_i, a_j) \ge r$。
- 令 $f_{i,j} = [a_i < j]$，那么以上三种限制都可以用 2-SAT 刻画。并且 2-SAT 的方案可以对应一组唯一的构造方案。
- 对于树的情况，考虑拓展链的做法，$[a_i < j]$ 的限制实际上是规定了 a_i 在一条边的左边或右边。

- 对于链的情况，限制可以用以下形式刻画：$a_i = a_j$ 或 $\max(a_i, a_j) \le l$ 或 $\min(a_i, a_j) \ge r$。
- 令 $f_{i,j} = [a_i < j]$，那么以上三种限制都可以用 2-SAT 刻画。并且 2-SAT 的方案可以对应一组唯一的构造方案。
- 对于树的情况，考虑拓展链的做法，$[a_i < j]$ 的限制实际上是规定了 a_i 在一条边的左边或右边。
- 那么对于树的情况，对每个 a_i 令每条边的方向为一个 01 变量，表示 a_i 在这条边的哪个方向。最后会形成一棵内向树，树根就是 a_i 所在位置。

- 仍然用 2-SAT 刻画条件。令 $f_{i,j}$ 表示任意定根后 i 对应的树，$j \to fa_j$ 的边是否向上连。

qoj7869

- 仍然用 2-SAT 刻画条件。令 $f_{i,j}$ 表示任意定根后 i 对应的树，$j \to fa_j$ 的边是否向上连。
- 根据经典结论，内向树等价于每个点的出度至多为 1。所以可以以每个点为中心前后缀优化建边来限制这个条件，边数为 $O(nm)$。

- 仍然用 2-SAT 刻画条件。令 $f_{i,j}$ 表示任意定根后 i 对应的树，$j \to fa_j$ 的边是否向上连。
- 根据经典结论，内向树等价于每个点的出度至多为 1。所以可以以每个点为中心前后缀优化建边来限制这个条件，边数为 $O(nm)$。
- a_i 的范围这个条件会钦定一些边的方向，在虚树外部的边一定是指向虚树方向的。这部分的边数是 $O(nm)$ 的。

qoj7869

- 仍然用 2-SAT 刻画条件。令 $f_{i,j}$ 表示任意定根后 i 对应的树，$j \to fa_j$ 的边是否向上连。
- 根据经典结论，内向树等价于每个点的出度至多为 1。所以可以以每个点为中心前后缀优化建边来限制这个条件，边数为 $O(nm)$。
- a_i 的范围这个条件会钦定一些边的方向，在虚树外部的边一定是指向虚树方向的。这部分的边数是 $O(nm)$ 的。
- 对于另一个条件，如果 (u, v) 在 V 的虚树上，那么对于任意 $i, j \in S$，有 $f_{i,u} = f_{j,u}$。暴力连边复杂度 $O(n \sum |S|)$，不能通过。

- 优化上述做法。发现操作类似于并查集合并若干元素。

qoj7869

- 优化上述做法。发现操作类似于并查集合并若干元素。
- 考虑使用树剖给 V_i 对应虚树上的所有边对应的线段树节点合并 S_i 内的元素。复杂度 $O(\sum |V||S| \log^2 n)$。

- 优化上述做法。发现操作类似于并查集合并若干元素。
- 考虑使用树剖给 V_i 对应虚树上的所有边对应的线段树节点合并 S_i 内的元素。复杂度 $O(\sum |V||S| \log^2 n)$。
- 最后用线段树分治和可撤销并查集结合就能求出每个位置对应的并查集情况，复杂度 $O(nm \log n)$，2-SAT 边数 $O(nm)$。

- 优化上述做法。发现操作类似于并查集合并若干元素。
- 考虑使用树剖给 V_i 对应虚树上的所有边对应的线段树节点合并 S_i 内的元素。复杂度 $O(\sum|V||S|\log^2 n)$。
- 最后用线段树分治和可撤销并查集结合就能求出每个位置对应的并查集情况，复杂度 $O(nm\log n)$，2-SAT 边数 $O(nm)$。
- 由于树剖跑不满可以轻松通过。

致谢

谢谢大家！