

🔊 欢迎来到 Siyuan 的博客！希望我们能相互交流，共同进步～



Siyuan 的博客

你强归你强，我永不示弱。

置顶 CSP 2019 算法模板复习！ (<https://blog.orzsiyuan.com/archives/hehezhou-AK-CSP-2019/>)



👤 Siyuan (<https://blog.orzsiyuan.com/author/1/>) ⌚ 2019 年 09 月 22 日

有用的文章整理

(<https://blog.orzsiyuan.com/archives/%E5%A5%BD%E5%BA%B7%E7%9A%84/>)

以下文章不定期收集和更新，欢迎大家提供更多优秀文章！

👤 Siyuan (<https://blog.orzsiyuan.com/author/1/>) ⌚ 2019 年 11 月 04 日

「Codeforces 1228E」 Another Filling the Grid
(<https://blog.orzsiyuan.com/archives/Codeforces-1228E-Another-Filling-the-Grid/>)

题目链接: Codeforces 1228E (<https://codeforces.com/contest/1228/problem/E>)

你有一个 $n \times n$ 的网格和一个整数 k , 在每个格子中都填入一个整数, 满足如下条件:

- 所有格子中的整数都介于 1 到 k 之间。
- 第 i 行的最小值为 1 ($1 \leq i \leq n$) 。
- 第 j 列的最小值为 1 ($1 \leq j \leq n$) 。

请求出填数的方案数, 答案对 $10^9 + 7$ 取模。

数据范围: $1 \leq n \leq 250, 1 \leq k \leq 10^9$ 。

👤 Siyuan (<https://blog.orzsiyuan.com/author/1/>) © 2019 年 10 月 01 日

[Codeforces 1217D] Coloring Edges
(<https://blog.orzsiyuan.com/archives/Codeforces-1217D-Coloring-Edges/>)

题目链接: Codeforces 1217D (<https://codeforces.com/contest/1217/problem/D>)

你有一个包含 n 个点和 m 条边的有向图 (没有自环或重边) 。

定义一张图的 k 染色为: 将每条边染成 k 种颜色中的一种。一个 k 染色是好的当且仅当不存在一个环满足环上的所有边颜色相同。

你需要求出这张图的 k 染色, 并最小化 k 的值。

数据范围: $2 \leq n \leq 5000, 1 \leq m \leq 5000$ 。

👤 Siyuan (<https://blog.orzsiyuan.com/author/1/>) © 2019 年 09 月 14 日

[Codeforces 1204E] Natasha, Sasha and the Prefix Sums
(<https://blog.orzsiyuan.com/archives/Codeforces-1204E-Natasha-Sasha-and-the-Prefix-Sums/>)

题目链接: Codeforces 1204 (<https://codeforces.com/contest/1204/problem/E>)

Natasha 最喜欢的数字是 n 和 1, Sasha 最喜欢的数字是 m 和 -1 。某一天他们写下了长度为 $n + m$ 且包含恰好 n 个 1 和 m 个 -1 的所有可能的序列。对于每一个序列计算出它的最大前缀和 (允许为空); 形式化地, 我们定义 $f(a)$ 表示序列 $a_1, \dots, a_l (l \leq 0)$ 的最大前缀和, 那么有:

$$f(a) = \max \left(0, \max_{i=1}^l \sum_{j=1}^i a_j \right)$$

现在他们想要对于所有满足条件的序列, 求出 $f(a)$ 的总和。答案对 998244853 取模。

数据范围： $0 \leq n, m \leq 2000$ 。

👤 Siyuan (<https://blog.orzsiyuan.com/author/1/>) ⌚ 2019 年 09 月 08 日

「Codeforces 662C」 Binary Table (<https://blog.orzsiyuan.com/archives/Codeforces-662C-Binary-Table/>)

题目链接：Codeforces 662C (<https://codeforc.es/contest/662/problem/C>)

你有一个 $n \times m$ 的表格。每个格子都有一个数字 0 或 1，你可以任意选择某一行或者某一列并将其翻转。请问通过任意次操作后表格中 1 的个数的最小值是多少？

数据范围： $1 \leq n \leq 20, 1 \leq m \leq 10^5$ 。

👤 Siyuan (<https://blog.orzsiyuan.com/author/1/>) ⌚ 2019 年 08 月 31 日

「TJOI / HEOI 2016」 求和 (<https://blog.orzsiyuan.com/archives/TJOI-HEOI-2016-Sum/>)

题目链接：LOJ 2058 (<https://loj.ac/problem/2058>)

在 2016 年，佳媛姐姐刚刚学习了第二类斯特林数，非常开心。

现在他想计算这样一个函数的值：

$$f(n) = \sum_{i=0}^n \sum_{j=0}^i S(i, j) \cdot 2^j \cdot j!$$

$S(i, j)$ 表示第二类斯特林数，递推公式为： $S(i, j) = j \cdot S(i-1, j) + S(i-1, j-1)$, $1 \leq j \leq i-1$ 。

边界条件为： $S(i, i) = 1 (i \geq 0)$, $S(i, 0) = 0 (i \geq 1)$ 。

你能帮帮她吗？

数据范围： $1 \leq n \leq 10^5$ 。

👤 Siyuan (<https://blog.orzsiyuan.com/author/1/>) ⌚ 2019 年 08 月 31 日

「Codeforces 1174F」 Ehab and the Big Finale (<https://blog.orzsiyuan.com/archives/Codeforces-1174F-Ehab-and-the-Big-Finale/>)

题目链接：Codeforces 1174F (<https://codeforces.com/problemset/problem/1174/F>)

这是一道交互题。

给定一棵有 n 个点的树，节点 1 为根节点。

我们选择一个隐藏节点 x ，你需要进行以下三种操作来找到这个节点 x 的编号。

- $d\ u$ ：你会得到节点 u 和 x 之间的距离。两个节点之间的距离定义为最短路径上的边数。
- $s\ u$ ：你会得到节点 u 到 x 的最短路径上的第二个节点。但是如果 u 不是 x 的祖先，你会直接得到 Wrong answer 的结果！
- $! u$ ：回答隐藏节点 x 的编号为 u 。

你需要在 36 次询问（不包括回答）内找到 x 的编号。这个隐藏节点 x 不会根据你的询问而改变。

数据范围： $2 \leq n \leq 2 \times 10^5$ 。

👤 Siyuan (<https://blog.orzsiyuan.com/author/1/>) © 2019 年 08 月 07 日

「Codeforces 1189F」Array Beauty
(<https://blog.orzsiyuan.com/archives/Codeforces-1189F-Array-Beauty/>)

题目链接：Codeforces 1189F (<https://codeforces.com/contest/1189/problem/F>)

我们定义一个序列 $b_1, b_2, \dots, b_n (n > 1)$ 的「美丽值」为 $\min_{1 \leq i < j \leq n} |b_i - b_j|$ 。

我们给定一个序列 a_1, a_2, \dots, a_n 个一个数字 k 。请计算出所有长度恰好为 k 的子序列的「美丽值」之和，答案对 998244353 取模。

数据范围： $2 \leq k \leq n \leq 1000, 0 \leq a_i \leq 10^5$ 。

👤 Siyuan (<https://blog.orzsiyuan.com/author/1/>) © 2019 年 08 月 05 日

「Codeforces 1189D2」Add on a Tree: Revolution
(<https://blog.orzsiyuan.com/archives/Codeforces-1189D2-Add-on-a-Tree-Revolution/>)

题目链接：Codeforces 1189D2 (<https://codeforces.com/contest/1189/problem/D2>)

你有一个棵 n 个点的树，初始所有的边上的数字都是 0。对于每次操作，你可以选择两个不同的叶子节点 u, v 和一个任意整数 x 并把 $u - v$ 这条简单路径上的边加上 x 。

每条边都有一个目标状态，用一个两两不同的非负偶数表示。你需要判断这个目标状态是否可以通过有限次操作达到。如果可行则输出 YES 和构造的方案；否则输出 NO。

注意叶子节点的定义为度数为 1 的点。

数据范围： $2 \leq n \leq 10^5$ 。

「Codeforces 1189D1」Add on a Tree (<https://blog.orzsiyuan.com/archives/Codeforces-1189D1-Add-on-a-Tree/>)

题目链接: Codeforces 1189D1 (<https://codeforces.com/contest/1189/problem/D1>)

你有一个棵 n 个点的树, 初始所有的边上的数字都是 0。对于每次操作, 你可以选择两个不同的叶子节点 u, v 和一个任意实数 x 并把 $u - v$ 这条简单路径上的边加上 x 。

我们令 w_i 表示最终第 i 条边上的实数, 是否对于所有的 $w_i \in \mathbf{R}, 1 \leq i < n$, 都存在有限的操作使得所有的边都满足条件? 如果可行则输出 YES 否则输出 NO。

注意叶子节点的定义为度数为 1 的点。

数据范围: $2 \leq n \leq 10^5$ 。


[1 \(https://blog.orzsiyuan.com/page/1/\)](https://blog.orzsiyuan.com/page/1/)
[2 \(https://blog.orzsiyuan.com/page/2/\)](https://blog.orzsiyuan.com/page/2/)
[3 \(https://blog.orzsiyuan.com/page/3/\)](https://blog.orzsiyuan.com/page/3/)
[4 \(https://blog.orzsiyuan.com/page/4/\)](https://blog.orzsiyuan.com/page/4/)
[...](#)
[19 \(https://blog.orzsiyuan.com/page/19/\)](https://blog.orzsiyuan.com/page/19/)
[➤ \(https://blog.orzsiyuan.com/page/2/\)](https://blog.orzsiyuan.com/page/2/)


热门文章

(<https://blog.orzsiyuan.com/archives/ZJOI-2019/>)  6051

(<https://blog.orzsiyuan.com/archives/hehezhou-AK-CSP-2019/>)  2892

CSP-2019) (<https://blog.orzsiyuan.com/archives/Polynomial-Template/>)  1080

(<https://blog.orzsiyuan.com/archives/SDOI-2017-Number-Table/>)  1026

Number-Theory (https://blog.orzsiyuan.com/archives/TJOI-2019-Sing-Table/)

2019- Dance-Rap-and-Basketball/

Sing- 👁 843

Dance-

Rap-

and-

Basketball/)

博客信息

📄 文章数目	187
💬 评论数目	243
📅 运行天数	1年25天
🔄 最后活动	4 个月前

标签云

- Codeforces (https://blog.orzsiyuan.com/tag/Codeforces/)
- 数据结构 (https://blog.orzsiyuan.com/tag/Data-Structure/)
- 动态规划 (https://blog.orzsiyuan.com/tag/Dynamic-Programming/)
- 数论 (https://blog.orzsiyuan.com/tag/Number-Theory/)
- 图论 (https://blog.orzsiyuan.com/tag/Graph-Theory/)
- 贪心 (https://blog.orzsiyuan.com/tag/Greedy/)
- 多项式 (https://blog.orzsiyuan.com/tag/Polynomial/)
- 字符串 (https://blog.orzsiyuan.com/tag/%E5%AD%97%E7%AC%A6%E4%B8%B2/)
- LOJ (https://blog.orzsiyuan.com/tag/LOJ/)
- FFT NTT (https://blog.orzsiyuan.com/tag/FFT-NTT/)
- 网络流 (https://blog.orzsiyuan.com/tag/Network-Flow/)
- LCT (https://blog.orzsiyuan.com/tag/LCT/)
- 计数 (https://blog.orzsiyuan.com/tag/%E8%AE%A1%E6%95%B0/)
- 后缀数组 (https://blog.orzsiyuan.com/tag/%E5%90%8E%E7%BC%80%E6%95%B0%E7%BB%84/)
- 线段树 (https://blog.orzsiyuan.com/tag/Segment-Tree/)
- 构造 (https://blog.orzsiyuan.com/tag/%E6%9E%84%E9%80%A0/)
- HDU (https://blog.orzsiyuan.com/tag/HDU/)
- SPOJ (https://blog.orzsiyuan.com/tag/SPOJ/)
- Luogu (https://blog.orzsiyuan.com/tag/Luogu/)
- BZOJ (https://blog.orzsiyuan.com/tag/BZOJ/)
- 树状数组 (https://blog.orzsiyuan.com/tag/Binary-Indexed-Tree/)
- CDQ 分治 (https://blog.orzsiyuan.com/tag/CDQ-Divide-and-Conquer/)
- UOJ (https://blog.orzsiyuan.com/tag/UOJ/)
- 主席树 (https://blog.orzsiyuan.com/tag/Chairman-Tree/)
- 高斯消元 (https://blog.orzsiyuan.com/tag/Gaussian-Elimination/)
- 莫比乌斯反演 (https://blog.orzsiyuan.com/tag/Mobius-Inversion/)
- AtCoder (https://blog.orzsiyuan.com/tag/AtCoder/)

多项式乘法 (<https://blog.orzsiyuan.com/tag/%E5%A4%9A%E9%A1%B9%E5%BC%8F%E4%B9%98%E6%B3%95/>)

并查集 (<https://blog.orzsiyuan.com/tag/Union-Find-Set/>)

最大流 (<https://blog.orzsiyuan.com/tag/Maximum-Flow/>)

费用流 (<https://blog.orzsiyuan.com/tag/Minimum-Cost/>) Splay (<https://blog.orzsiyuan.com/tag/Splay/>)

离线 (<https://blog.orzsiyuan.com/tag/Off-Line/>)

二分答案 (<https://blog.orzsiyuan.com/tag/Binary-Search-Answer/>)

权值线段树 (<https://blog.orzsiyuan.com/tag/Weight-Segment-Tree/>)

容斥 (<https://blog.orzsiyuan.com/tag/%E5%AE%B9%E6%96%A5/>)

数论分块 (<https://blog.orzsiyuan.com/tag/%E6%95%B0%E8%AE%BA%E5%88%86%E5%9D%97/>)

计算几何 (<https://blog.orzsiyuan.com/tag/Geometry/>) 组合数学 (<https://blog.orzsiyuan.com/tag/Combinatorics/>)

矩阵 (<https://blog.orzsiyuan.com/tag/Matrix/>) 最小割 (<https://blog.orzsiyuan.com/tag/Minimum-Cut/>)

随机化 (<https://blog.orzsiyuan.com/tag/Randomization/>)

斜率优化 (<https://blog.orzsiyuan.com/tag/Slope-Optimization/>) NOI (<https://blog.orzsiyuan.com/tag/NOI/>)

概率期望 (<https://blog.orzsiyuan.com/tag/%E6%A6%82%E7%8E%87%E6%9C%9F%E6%9C%9B/>)

后缀自动机 (<https://blog.orzsiyuan.com/tag/%E5%90%8E%E7%BC%80%E8%87%AA%E5%8A%A8%E6%9C%BA/>)

位运算 (<https://blog.orzsiyuan.com/tag/%E4%BD%8D%E8%BF%90%E7%AE%97/>)

生成函数 (<https://blog.orzsiyuan.com/tag/%E7%94%9F%E6%88%90%E5%87%BD%E6%95%B0/>)

莫队 (<https://blog.orzsiyuan.com/tag/Mo-Algorithm/>) BJOI (<https://blog.orzsiyuan.com/tag/BJOI/>)

线性基 (<https://blog.orzsiyuan.com/tag/Linear-Base/>) 分块 (<https://blog.orzsiyuan.com/tag/Partition/>)

凸包 (<https://blog.orzsiyuan.com/tag/Convex-Hull/>) POJ (<https://blog.orzsiyuan.com/tag/POJ/>)

平衡树 (<https://blog.orzsiyuan.com/tag/Balanced-Tree/>)

线性筛 (<https://blog.orzsiyuan.com/tag/Euler-Sieve-Method/>) FWT (<https://blog.orzsiyuan.com/tag/FWT/>)

单调栈 (<https://blog.orzsiyuan.com/tag/%E5%8D%95%E8%B0%83%E6%A0%88/>)

杜教筛 (<https://blog.orzsiyuan.com/tag/%E6%9D%9C%E6%95%99%E7%AD%9B/>)

多项式指数函数 (<https://blog.orzsiyuan.com/tag/%E5%A4%9A%E9%A1%B9%E5%BC%8F%E6%8C%87%E6%95%B0%E5%>)

行列式 (<https://blog.orzsiyuan.com/tag/Determinant/>)

欧拉函数 (<https://blog.orzsiyuan.com/tag/Euler-Function/>) 树形 DP (<https://blog.orzsiyuan.com/tag/Tree-DP/>)

Two Pointers (<https://blog.orzsiyuan.com/tag/Two-Pointers/>)

模拟退火 (<https://blog.orzsiyuan.com/tag/Simulated-Annealing/>) NOIP (<https://blog.orzsiyuan.com/tag/NOIP/>)

偏序 (<https://blog.orzsiyuan.com/tag/Partial-Order/>) TJOI (<https://blog.orzsiyuan.com/tag/TJOI/>)

整体二分 (<https://blog.orzsiyuan.com/tag/Binary-Search-Whole/>) ZJOI (<https://blog.orzsiyuan.com/tag/ZJOI/>)

积性函数 (<https://blog.orzsiyuan.com/tag/Multiplicative-Function/>)

RMQ (<https://blog.orzsiyuan.com/tag/RMQ/>)

决策单调性 (<https://blog.orzsiyuan.com/tag/%E5%86%B3%E7%AD%96%E5%8D%95%E8%B0%83%E6%80%A7/>)

二分 (<https://blog.orzsiyuan.com/tag/%E4%BA%8C%E5%88%86/>)

多项式求逆 (<https://blog.orzsiyuan.com/tag/%E5%A4%9A%E9%A1%B9%E5%BC%8F%E6%B1%82%E9%80%86/>)

多项式开根 (<https://blog.orzsiyuan.com/tag/%E5%A4%9A%E9%A1%B9%E5%BC%8F%E5%BC%80%E6%A0%B9/>)

数学归纳法 (<https://blog.orzsiyuan.com/tag/%E6%95%B0%E5%AD%A6%E5%BD%92%E7%BA%B3%E6%B3%95/>)

多项式自然对数 (<https://blog.orzsiyuan.com/tag/%E5%A4%9A%E9%A1%B9%E5%BC%8F%E8%87%AA%E7%84%B6%E5%>

多项式快速幂 (<https://blog.orzsiyuan.com/tag/%E5%A4%9A%E9%A1%B9%E5%BC%8F%E5%BF%AB%E9%80%9F%E5%B9%>

最小圆覆盖 (<https://blog.orzsiyuan.com/tag/Smallest-Enclosing-Circle/>)

BSGS (<https://blog.orzsiyuan.com/tag/BSGS/>) 可持久化 (<https://blog.orzsiyuan.com/tag/Persistence/>)

拉格朗日插值 (<https://blog.orzsiyuan.com/tag/Lagrange-Interpolation/>)

同余 (<https://blog.orzsiyuan.com/tag/Congruence/>)

线性同余方程 (<https://blog.orzsiyuan.com/tag/Linear-Congruence-Theorem/>)

exGCD (<https://blog.orzsiyuan.com/tag/exGCD/>) CRT (<https://blog.orzsiyuan.com/tag/CRT/>)

exCRT (<https://blog.orzsiyuan.com/tag/exCRT/>) 逆矩阵 (<https://blog.orzsiyuan.com/tag/Matrix-Inversion/>)

最短路 (<https://blog.orzsiyuan.com/tag/Shortest-Path/>) Floyd (<https://blog.orzsiyuan.com/tag/Floyd/>)

类欧几里德算法 (<https://blog.orzsiyuan.com/tag/Similar-Euclidean-Algorithm/>)

叉积 (<https://blog.orzsiyuan.com/tag/Cross-Product/>) HEOI (<https://blog.orzsiyuan.com/tag/HEOI/>)

最大子段和 (<https://blog.orzsiyuan.com/tag/Maximum-Interval-Sum/>)

递推 (<https://blog.orzsiyuan.com/tag/Recursion/>) 缩点 (<https://blog.orzsiyuan.com/tag/Shrinking-Point/>)

单调队列 (<https://blog.orzsiyuan.com/tag/%E5%8D%95%E8%B0%83%E9%98%9F%E5%88%97/>)

重心 (<https://blog.orzsiyuan.com/tag/%E9%87%8D%E5%BF%83/>)

上下界网络流 (<https://blog.orzsiyuan.com/tag/%E4%B8%8A%E4%B8%8B%E7%95%8C%E7%BD%91%E7%BB%9C%E6%B8%>

AHOI (<https://blog.orzsiyuan.com/tag/AHOI/>)

倍增 (<https://blog.orzsiyuan.com/tag/%E5%80%8D%E5%A2%9E/>)

二分图 (<https://blog.orzsiyuan.com/tag/%E4%BA%8C%E5%88%86%E5%9B%BE/>)

差分 (<https://blog.orzsiyuan.com/tag/%E5%B7%AE%E5%88%86/>)

Dirichlet 卷积 (<https://blog.orzsiyuan.com/tag/Dirichlet-%E5%8D%B7%E7%A7%AF/>)

多省联考 (<https://blog.orzsiyuan.com/tag/%E5%A4%9A%E7%9C%81%E8%81%94%E8%80%83/>)

优先队列 (<https://blog.orzsiyuan.com/tag/%E4%BC%98%E5%85%88%E9%98%9F%E5%88%97/>)

启发式合并 (<https://blog.orzsiyuan.com/tag/%E5%90%AF%E5%8F%91%E5%BC%8F%E5%90%88%E5%B9%B6/>)

Trie (<https://blog.orzsiyuan.com/tag/Trie/>) Tarjan (<https://blog.orzsiyuan.com/tag/Tarjan/>)

线段树合并 (<https://blog.orzsiyuan.com/tag/%E7%BA%BF%E6%AE%B5%E6%A0%91%E5%90%88%E5%B9%B6/>)

SDOI (<https://blog.orzsiyuan.com/tag/SDOI/>) 交互 (<https://blog.orzsiyuan.com/tag/%E4%BA%A4%E4%BA%92/>)

欧拉路径 (<https://blog.orzsiyuan.com/tag/%E6%AC%A7%E6%8B%89%E8%B7%AF%E5%BE%84/>)

多项式除法 (<https://blog.orzsiyuan.com/tag/%E5%A4%9A%E9%A1%B9%E5%BC%8F%E9%99%A4%E6%B3%95/>)

多项式取模 (<https://blog.orzsiyuan.com/tag/%E5%A4%9A%E9%A1%B9%E5%BC%8F%E5%8F%96%E6%A8%A1/>)

多项式三角函数 (<https://blog.orzsiyuan.com/tag/%E5%A4%9A%E9%A1%B9%E5%BC%8F%E4%B8%89%E8%A7%92%E5%>

树链剖分 (<https://blog.orzsiyuan.com/tag/%E6%A0%91%E9%93%BE%E5%89%96%E5%88%86/>)

第二类斯特林数 (<https://blog.orzsiyuan.com/tag/%E7%AC%AC%E4%BA%8C%E7%B1%BB%E6%96%AF%E7%89%B9%E6%96%A1%E7%90%86/>)

二项式定理 (<https://blog.orzsiyuan.com/tag/%E4%BA%8C%E9%A1%B9%E5%BC%8F%E5%AE%9A%E7%90%86/>)

© 2020 Copyright 浙ICP备19008446号-1 (<http://www.beian.miit.gov.cn>)