简要题解

pufanyi

目录

1 USACO

1.1 USACO Open Contest 2008, Cow Neighborhoods

https://www.luogu.com.cn/problem/P2906

现有 n 个点，$n \leq 10^5$，两点有边当且仅当其曼哈顿距离不超过一个定值 r，求连通块个数及点数最多的连通块有几个点。

先是曼哈顿与切比雪夫之间的转化。

把 (x, y) 变成 $(x + y, x - y)$。

我们不难发现 $\max\{|x'_a - x'_b|, |y'_a - y'_b|\} = |x_a - x_b| + |y_a - y_b|$。

于是就变成了求切比雪夫距离小于等于 r 的连通块。

考虑对 x 排序，维护 y。从左往右扫，维护一个 `set` 和一个并查集，`set` 维护到当前 x 坐标距离小于等于 r 的所有 y 坐标，我们发现每次插入一个值是仅需与 `set` 中的前趋和后继考虑一下是否合并即可。

1.2 USACO December Contest 2005, Cow Patterns

http://poj.org/problem?id=3167

https://www.spoj.com/problems/CPATTERN/

给你两个字符串 A, B，要你求 A 能匹配多少次 B。A, B 长度分别为 n, m。字符集大小为 S。$n \leq 10^5, m \leq 25000, S \leq 25$。

两个字符串匹配本来指的是长度相同且各个位相同；我们这里改变一下，称两个字符串匹配，当且仅当他们按大小离散化后得到的字符串相同（也就是对应位置上的数的 rank 是一样的）。

考虑 KMP，每次加一个字符的时候，判断一下加入这个字符在已经匹配过的字符串的 rank，即比它小的数的个数和与它相等数的个数。这个东西用前缀和预处理一下即可。

1.3 USACO December Contest 2007, Best Cow Line, Gold

https://www.luogu.com.cn/problem/P2870

有一个长度为 n 的字符串 `s`（$n \leq 30000$，洛谷上 $n \leq 500000$），还有一个初始为空串的字符串 `t`。每次可以从 s 头部或尾部删除一个字符，并将其加入到 `t` 尾部，直到 `s` 为空。求字典序最小的 `t`。

发现如果现在区间是 $[l, r]$，如果 s_l 的后缀比 s_r 的前缀（逆序）小，那么取 s_l，否则取 s_r。于是直接把 `s` 与 `s` 的逆序串接起来做一遍 `SA` 即可。

1.4 USACO December Contest 2018, Balance Beam

http://usaco.org/index.php?page=viewproblem2&cpid=864

给你一段长度为 n 的序列（$n \le 10^5$），你每次可以以 $\frac{1}{2}$ 的概率向左或向右移动，如果移动到 0 或 $n+1$ 则结束并收益为 0，或者可以选择结束并获得该点收益 f_i（$f_i \le 10^9$）。求在最优策略下从每个点开始的期望收益。答案乘以 10^5 后下取整。

首先考虑到如果一个点选择结束的收益比选择继续走的收益要小，那肯定是选择结束，我们把这些点成为“关键点”。如果我们找到所有的关键点，我们考虑计算所有非关键点。我们令 g_i 表示第 i 个点的答案，如果 i 不是关键点，那么有 $g_i = \frac{g_{i-1}+g_{i+1}}{2}$，发现 g_i 是个等差数列，于是我们可以在知道所有关键点的情况下得到答案了。

现在问题就变成了找到所有的关键点。我们考虑如果 i, j 是关键点（$i < j$）且 i, j 之间没有关键点，那 i, j 之间的点的答案连起来显然是一条直线（因为是等差数列）。由于 f_k 一定小于 g_k，也就是说 (k, f_k) 一定在 (k, g_k) 的下面。这样我们就有结论：关键点一定是 (i, f_i) 所得上凸壳上的点。

于是我们建出所有点，跑一遍凸壳，最后 $\mathcal{O}(n)$ 求一遍答案即可。由于不需要排序，所以总复杂度 $\mathcal{O}(n)$。

1.5 USACO December Contest 2018, Sort It Out

http://usaco.org/index.php?page=viewproblem2&cpid=865

给定一个长为 n 的排列 $\{a_i\}$（$n \le 10^5$），可以选择一个集合 S 使这个集合内部元素排到自己在整个序列中应该在的位置（即对于集合 S 内的每一个元素 i，使其排到第 i 号位置），使得整个排列在排序后为上升序列。求满足这样条件的，且集合大小最小的集合中从小到大排序后字典序第 k 小的集合。

题意显然可以转化成求字典序第 k 大的集合，我们考虑对每个点维护一当前点位开头最长上升子序列的长度与数量。这个东西我们从后往前枚举，线段树第 i 号位置记录 $a_j = i$ 时以 j 为开头的最长上升子序列的长度及数量，每次直接转移。

接下来我们开 l 个 `vector`（l 是最长上升子序列的长度），v[i]表示以此为开头的最长上升子序列长度为 i 的编号。

由于是字典序，考虑逐位确定即可。

1.6 USACO February Contest 2019, Cow Dating

http://www.usaco.org/index.php?page=viewproblem2&cpid=924

有一个长度为 n（$n \le 10^6$）的序列，第 i 个元素出现的概率为 p，找一段连续的区间，使其中恰好有且仅有一个元素出现的概率最大，求该概率。

考虑分治，考虑答案是 $\sum_{i=l}^{r} p_i \prod_{(l \le j \le r) \wedge (j \ne i)} (1 - p_j)$，维护前缀和后缀的答案，合并的时候发现这个东西跑完单调栈之后可以双指针，于是就可以合并了。

1.7 USACO December Contest 2018, The Cow Gathering

http://www.usaco.org/index.php?page=viewproblem2&cpid=866

给定一棵大小为 n 的树, 有 m 个限制（$n, m \leq 3000$），限制 (u, v) 表示 u 必须在 v 前删除，问每次删除一个叶子节点, 可能最后留下的点的集合。

首先需要发现一个性质，那就是所有可行点都应该是在一个连通块里的，因为如果 A, B 两个点都可行，那么从 A 点可以一直删到 B，从 B 点也可以一直删到 A，所以 A, B 之间的点都是可行的。

于是我们可以先贪心地找到第一个点，这个很好找，直接考虑每次删掉一个可以被删的点即可。

然后我们考虑对这个点进行拓展，如果不能往外拓展，那显然之后的点都不是可行点。我们可以对每个点进行 `dfs`，如果搜索树出现返祖边，即有限制说 u 必须在 v 之前删除但 u 又是 v 的祖先，这样显然是不可行的，否则就是可行点。

1.8 USACO February Contest 2019, Mowing Mischief

http://www.usaco.org/index.php?page=viewproblem2&cpid=926

给定平面上的一些点，求这些点的一个 `LIS`，并且还需要满足 $\sum_{i=1}^{n-1}(x_{i+1}-x_i)(y_{i+1}-y_i)$ 最小。$n \leq 2 \times 10^5, 0 \leq x_i, y_i \leq 10^6$，保证 x_i, y_i 互不相同。

f_i 表示以 i 为末尾的答案，于是我们就有一个 $\mathcal{O}(n^2)$ 的 `dp`。考虑把以该点开始的 `LIS` 值分层，发现每一层肯定是 x_i 增 y_i 减的。我们只需考虑层与层之间的转移即可。

然后我们考虑推式子。如果对于 i，有 j 和 k 两个决策点，如果 j 比 k 优，那么我们有：

$$f_j + (x_i - x_j)(y_i - y_j) \geq f_k + (x_i - x_k)(y_i - y_k)$$

即：

$$x_i(y_k - y_j) + y_i(x_k - x_j) \geq f_k - f_j + x_k y_k - x_j y_j$$

发现是有决策单调性的，如果 $x_j > x_k$ 的话，那么 $y_k - y_j > 0, x_k - x_j < 0$，也就是 x_i 越大越容易满足。

如果没有 $x_j < x_i, y_j < y_i$ 的限制，我们可以直接用传统的分治做法解决解决，`f(l,r,L,R)` 表示待转移的点在 $[l, r]$，决策点在 $[L, R]$，然后递归下去即可。

可惜这个限制使得转移的实际决策点不单调，于是我们只能继续寻找性质。

我们发现，对与每个 i，满足 $x_j < x_i, y_j < y_i$ 的 j 是连续的，也就是可行的决策点是连续的。于是我们可以对每一层的决策点建立线段树，对每个节点开一个 `vector` 记录一下可行的转移点，类似区间覆盖一样吧每个带转移的点覆盖到线段树上去，覆盖到的点都是可行的决策点。我们发现线段树上每个节点的决策点和转移点都是已经满足条件的。对树上每个节点做一遍分治即可。总复杂度 $\mathcal{O}\left(n \log^2 n\right)$。

2 CodeForces

2.1 CodeForces 1264C Beautiful Mirrors with queries

https://codeforces.com/contest/1264/problem/C

你有 n 个魔镜（$n \leq 10^5$），第 i 个魔镜有 p_i 的概率说你美。从第 1 天开始，你会依次询问魔镜 $1 \sim n$ 你美不美。若第 i 个魔镜说你美则你明天会继续询问第 $i+1$ 个魔镜。否则你明天会从该魔镜前面第一个复活点魔镜开始询问，初始时只有魔镜 1 是复活点。当第 n 个魔镜说你美的时候你会开心的一批。

现在有 q 次操作（$q \leq 10^5$），每次操作修改一个魔镜使其成为/不成为复活点。每次操作之后请你求出期望多少天你能开心的一批。

考虑每个点对答案的贡献，不难发现加入一个复活点其实只是对一段区间的贡献乘上了一个权值，线段树维护这个东西即可。

2.2 CodeForces 1264D Beautiful Bracket Sequence

https://codeforces.com/contest/1264/problem/D1

https://codeforces.com/contest/1264/problem/D2

给定一个长度为 n 的字符串，其中只有 `'('`, `')'`, `'?'` 三种字符，其中 `'?'` 可以为 `'('` 或者 `')'`。对于一个括号序列，定义其权值为其通过删除字符后可以得到的合法的括号匹配的最深的深度，求出所有可能的括号序列（即问号替换后）的权值和。

在 `D1` 中，$n \leq 2000$；在 `D2` 中，$n \leq 10^6$。

考虑枚举答案计算方案数，显然对于一个括号序列，存在一个划分点使得划分点左边的左括号个数等于划分点右边的有括号个数。则此时该括号序列的深度就是划分点左边的左括号个数。这样 `dp` 之后合并一下是 $\mathcal{O}(n^2)$ 的。这样 `D1` 就做完了。

考虑把前面 `dp` 的式子转换成组合数，我们考虑每个分割点，前面有 l 个左括号，r 个右括号，前面有 x 个问号，后面有 y 个问号。

其实答案就是：

$$\sum_{i=0}^{x}(l+i)\binom{x}{i}\binom{y}{l+i-r} = l\sum_{i=0}^{x}\binom{x}{i}\binom{y}{l+i-r} + \sum_{i=0}^{x} i\binom{x}{i}\binom{y}{l+i-r}$$

对于左边的式子：

$$\begin{aligned} & l\sum_{i=0}^{x}\binom{x}{i}\binom{y}{l+i-r} \\ =\ & l\sum_{i=0}^{x}\binom{x}{i}\binom{y}{y+r-l-i} \\ =\ & l\binom{x+y}{y+r-l} \end{aligned}$$

对于右边的式子：

$$\begin{aligned}&\sum_{i=0}^{x} i\binom{x}{i}\binom{y}{l+i-r}\\=&\sum_{i=0}^{x} x\binom{x-1}{i-1}\binom{y}{y-l-i+r}\\=&x\binom{x+y-1}{y-l+r-1}\end{aligned}$$

所以该点的贡献为：

$$l\binom{x+y}{y+r-l}+x\binom{x+y-1}{y-l+r-1}$$

2.3 CodeForces 286E Ladies' Shop

https://codeforces.com/problemset/problem/286/E

有 n 个背包，第 i 个背包重量为 a_i ，$1 \le a_1 < a_2 < \cdots < a_n \le m$，每个背包只能装总重量恰好为 a_i 的物品，不多不少。请你构造一些物品（设有 k 个），第 i 个物品重量为 p_i ，$1 \le p_1 < p_2 < \cdots < p_k$，（每种重量的物品有无穷多个，每个背包里可以装多个重量相同的物品），满足下列条件：

1. 对于任意一个背包 a_i，存在一组物品使得总重恰为 a_i；
2. 对于任意一组总重不超过 m 的物品，存在一个背包恰好可以放进这些物品；
3. 在满足上面 2 条的前提下，要求物品个数 k 最少。

如果无解输出 "NO"，否则输出你找到的解。

$1 \le n, m \le 10^6$

由第 2 条可以发现，$p \in a$。

对于两个背包，由于它们可以表示成若干个物品的和，所以如果它们加起来 $\le m$，那么必然有一个背包等于它们加起来。如果不存在这样的背包，则无解。

对于每个背包，它要么可以用某个物品直接得到，要么就是用某个背包加某个背包得到。

为了使背包个数最少，只需要选择不能用其他背包加起来得到的背包作为 p_i。

上面的判断只要求出：

$$\left(\sum_{i=1}^{n} x_i^a\right)^2$$

FFT 一下即可。

2.4 CodeForces 1288F Red-Blue Graph

https://codeforces.com/problemset/problem/1288/F

有一张二分图，左边有 n_1 个点，右边有 n_2 个点，m 条边。每个点可能有一种颜色 R 或者 B，也可能没有，也就是 U。现在要给一些边染色，把边染成 R 要花费 r 的代

价，把边染成 B 要花费 b 的代价，要求对于每个颜色为 R 的点，与之相邻的边中 R 的边**严格多于** B 的边；对于每个颜色为 B 的点，与之相邻的边中 B 的边**严格多于** R 的边。求花费最小的方案，输出任意一种，无解输出 -1。其中 $1 \le n_1, n_2, m, r, b \le 200$。

考虑网络流建图，对于每条边 $\langle u, v \rangle$，在网络流图上建立两条边：$u \to v$，如果流表示将该边染成红色，$v \to u$，表示将改变染成黑色。

建立超源 s 和超汇 t，考虑左边红色点，s 向该点连一条下界为 1 的边，表示强制流红大于流黑，对于左边黑色点，该点向 t 连一条下界为 1 的边，表示强制流黑大于流红，右边同理。

最后跑一遍费用流即可。

2.5 CodeForces 391F3 Stock Trading

https://codeforces.com/contest/391/problem/F3

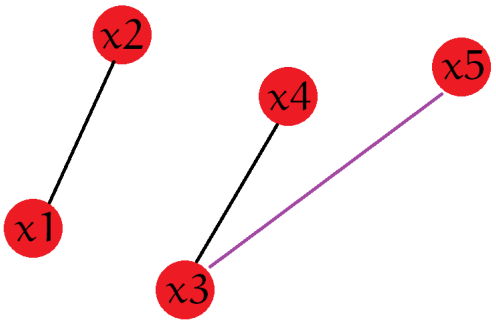
给出 n $(1 \le n \le 4000000)$ 天的股票价格，每天可买进或卖出一股，可以同时买进或卖出，也可以不操作，但最终手上只能有一股。问最多 k 次买进和卖出后的最大收益。

我们先来看一个 $\mathcal{O}(n \log n)$ 的做法。首先我们考虑如果是一段连续的递增股票，那我们相当于是一开始买进，最后卖出。如果这样连续的段小于等于 k，那么就直接可以返回答案了。如果比 k 大，那么我们就需要做出这样两种选择：要么选择一个上升的区间放弃，要么选择一段下降的区间把两段上升的区间给连起来。这个东西我们可以用一个堆或者 set 来维护。

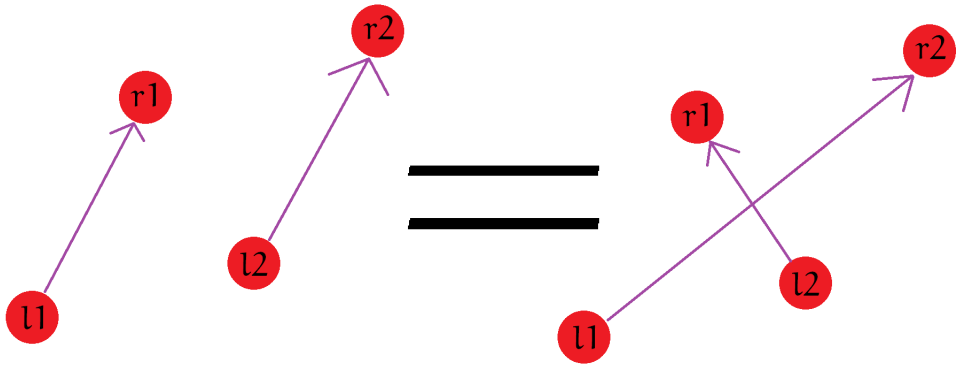
但是我们还有一个 $\mathcal{O}(n)$ 的做法，需要观察一些性质。我们把读入的股票看成是一个个单调不减的区间。当然有些区间可能只有一个数。

我们考虑一些区间可以合并成一个等价的区间，我们考虑将区间从左往后一个个加入。我们再开一个数组 b，记录那些不可能跟后面的区间合并的区间。

我们首先可以得到一个结论：在任一时刻对于那些可能与后面合并的区间，其买入时的价格一定是单调递增的。因为如果不是递增的，如下图情况，原来默认的匹配是 $[x_1, x_2]$，$[x_3, x_4]$，现在加入了一个右端点 x_5（左端点在哪儿我们并不需要关心），如果 x_1 跟 x_5 配，那显然不如 x_3 跟 x_5 配，因为这样不仅节省了一段区间（$[x_1, x_2]$），答案还更优。



我们还需要发现一个性质，那就是如果现在有两个区间 $[l_1, r_1], [l_2, r_2]$，如果有 $l_1 < l_2$ 且 $r_1 < r_2$，我们可以把它等价成为 $[l_1, r_2], [l_2, r_1]$ 这样两段区间。而 $[l_2, r_1]$ 是不可能再跟后面的匹配了，所以我们可以直接将其扔进 b 中。



根据这样两个性质，我们可以得到很多个区间。不难发现这些区间是独立的。最后我们取前 k 大的区间即可。

2.6 CodeForces 571D Campus

https://codeforces.com/contest/571/problem/D

有 n 个宿舍，一开始第 i 个宿舍隶属于第 i 个大学和第 i 个军部。有 m 个操作：

- `U a b`：合并两个大学。
- `M a b`：合并两个军部。
- `A x`：设隶属于 x 大学的宿舍数为 k，则所有隶属于 x 大学的宿舍内学生都加 k。
- `Z x`：隶属于 x 军部的宿舍全部清空。
- `Q x`：询问宿舍 x 的人数。

考虑不进行 `Z` 操作，此时可以直接维护。所以答案为当前询问的答案 $-$ 最近一次 `Z` 操作的答案，这个东西离线预处理一下即可。

具体操作可以直接线段树合并，也可以并查集。

由于线段树合并的空间是 $\mathcal{O}(n \log n)$ 的，但考虑到两棵线段树的节点结构相同，都是 `ls,rs,tag`，我们可以将其公用，然后手写一个内存池维护删除的节点即可卡入 `256MB`。

2.7 CodeForces 708E Student's Camp

https://codeforces.com/contest/708/problem/E

有一个 $(n+2) \times m$ 的长方形，除了第一行和最后一行，其他每一行每一天最左边和最右边的格子都有 p 的概率被摧毁，每行之间独立且左边和右边独立，求 k 天之后最上面一行与最下面一行四联通的概率。

其中 $1 \le n, m \le 1500$, $0 \le k \le 10^5$，答案对 $10^9 + 7$ 取模。

一个简单的想法就是我们令 $h_{i,l,r}$ 表示 k 天后第 i 行还剩下 $[l,r]$，前 i 行联通的概率。$P_{l,r}$ 表示单独一行只剩 $[l,r]$ 的概率。

首先我们有：

$$h_{i,l,r} = P_{l,r} \sum_{[l,r]\cap[l',r']\neq\emptyset} h_{i-1,l',r'}$$

接下来我们考虑 $P_{l,r}$，我们令 $q_i = \binom{k}{i} p^i (1-p)^{k-i}$，不难发现：

$$P_{l,r} = q_{l-1} q_{m-r}$$

然后我们令 $f_{i,r}$ 表示右端点为 r 的所有 h 之和，同理 $g_{i,l}$ 表示以左端点为 l 的所有 h 之和，再令 $F_{i,r}$ 表示右端点小于等于 r 的所有 h 之和，同理 $G_{i,l}$ 表示以左端点大于等于 l 的所有 h 之和。

具体地：

$$\begin{aligned}
f_{i,r} &= \sum_{l=1}^{r} h_{i,l,r} \\
g_{i,l} &= \sum_{r=l}^{m} h_{i,l,r} \\
F_{i,r} &= \sum_{R=1}^{r} f_{i,R} \\
G_{i,l} &= \sum_{L=l}^{m} g_{i,L}
\end{aligned}$$

于是我们就有：

$$h_{i,l,r} = P_{l,r} \left(F_{i-1,m} - F_{i-1,l-1} - G_{i-1,r+1}\right)$$

将 h 代入 f 和 g，我们有：

$$\begin{aligned}
f_{i,r} &= \sum_{l=1}^{r} q_{l-1} q_{m-r} \left(F_{i-1,m} - F_{i-1,l-1} - G_{i-1,r+1}\right) \\
&= q_{m-r} \left(\sum_{l=1}^{r} q_{l-1} \left(F_{i-1,m} - F_{i-1,l-1}\right) - G_{i-1,r+1} \sum_{l=1}^{r} q_{l-1} \right) \\
g_{i,l} &= \sum_{r=l}^{m} q_{l-1} q_{m-r} \left(F_{i-1,m} - F_{i-1,l-1} - G_{i-1,r+1}\right) \\
&= q_{l-1} \left(\sum_{r=l}^{m} q_{m-r} \left(F_{i-1,m} - G_{i-1,r+1}\right) - F_{i-1,l-1} \sum_{r=l}^{m} q_{m-r} \right)
\end{aligned}$$

考虑到最后答案是 $F_{n,m}$，我们发现我们可以绕过 h 来求 F。

最终复杂度 $\mathcal{O}(k+nm)$。

2.8 CodeForces 888G Xor-MST

https://codeforces.com/contest/888/problem/G

是有一个 $n\,(n \le 200000)$ 个点的完全图，每个节点的编号为 $a_i\,(0 \le a_i < 2^{30})$，$i$ 与 j 的边的权值是 $a_i \bigotimes a_j$，$\bigotimes$ 是按位异或，求该图的最小生成树。

大概用到了 Boruvka 的思想，先用所有点权建一棵 Trie 树，考虑 Trie 树上一棵子树，显然一开始肯定是他们自己连成一棵 MST，然后再和其兄弟连一条边，因为跨越子树像兄弟连一条边需要 2^d 的代价，d 是节点深度，显然自己连更优，于是建出 Trie 树之后分治（其实就是 dfs）下去即可。

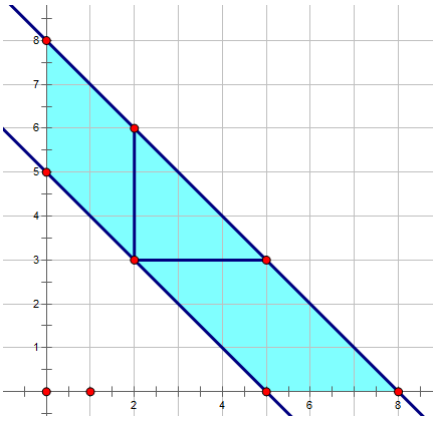
2.9 CodeForces 575I Robots protection

https://codeforces.com/contest/575/problem/I

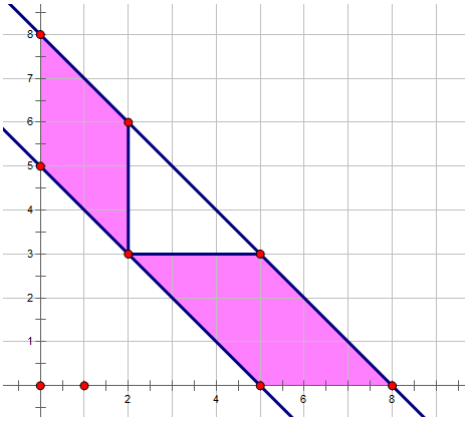
你需要在平面直角坐标系上进行 q 次操作。每次操作有两种，要么放置一个两条直角边平行于坐标轴的等腰直角三角形，要么查询某一个点被多少个三角形覆盖。保证所有点的坐标都是整数且 $\in [1,n]$。$n \le 5 \times 10^3, q \le 10^5$。

考虑到四种方向是对称的，我们只考虑其中一种情况，比如说直角顶点在左下角的情况，令三个顶点为 $(x,y),(x+l,y),(x,y+l)$。

首先我们发现需要加的所有点 (x',y')，都满足 $x+y \le x'+y' \le x+y+l$，这个东西用一棵树状数组就可以维护：



然后我们就是要减去这两块：



我们考虑如何减去，观察这些点的性质，首先均满足 $x'+y' \in [x+y, x+y+l]$，然后其还满足 $x' < x$ 或 $y' < y$。于是我们可以考虑开两棵二维树状数组，一棵维护下标 $(x+y, x)$，另一棵维护下标 $(x+y, y)$，然后在这两棵树状数组上矩阵加即可。

时间复杂度 $\mathcal{O}(q \log^2 n)$，空间 $\mathcal{O}(n^2)$。

2.10 CodeForces 516D Drazil and Morning Exercise

https://codeforces.com/contest/516/problem/D

给你一棵树，边有边权，定义 $f_x = \max_{i=1}^{n} \operatorname{dist}(x, i)$。$q$ 次询问，每次给出一个数 l 求最大的连通块 s 满足 $\max_{x \in s} f_x - \min_{x \in s} f_x \le l$。$n \le 10^5, q \le 50$。

首先假设这棵树的直径的两个端点为 u, v，不难发现 $f_x = \max(\operatorname{dist}(u, x)\,\mathrm{v}, \mathrm{x})$。这个用证明直径的方法可以证明。

然后我们需要发现一个性质，那就是将点按照 f 从大到小排序，那么儿子一定排在父亲的前面。

然后就好办了，我们考虑将点按 f 从大到小排序，然后我们考虑在上面 `two points`，一边用并查集维护两个指针所在区间的联通性，插入一个点直接将其儿子合并，删除节点是发现只会删除叶子节点，不会改变连通性，只要把该连通块 `size` 减掉即可。

2.11 CodeForces 576D Flights for Regular Customers

https://codeforces.com/contest/576/problem/D

条件是你已经走过了 d_i 条边，问 $1 \to n$ 至少需要走多少条边，或输出无解。$n \le m \le 150, 0 \le d_i \le 10^9$。

首先有一个 simple 的想法就是枚举路径上的 d 最大的边，然后先用比该边 d 小的边满足要求地走 d_i 步，然后再用 d 小于等于该边的边走到终点。

我们考虑将边从大到小排序，首先要算出前 d_i 步能到哪儿，这不难用矩乘，然后是走用小于等于 d_i 的边走到终点，这一个 `bfs` 即可解决。

2.12 CodeForces 512D Fox And Travelling

https://codeforces.com/contest/512/problem/D

有一张 n 个点 m 条边的简单无向图，每次选择一个度数小于等于 1 的点然后将其删除，对于每个 k 求删去 k 个点的方案数。$n \le 100, m \le \frac{n \times (n-1)}{2}$。

首先环上点肯定删不掉，如果一个连通块之间有两个环，那么被两个环夹着的点肯定也删不掉。

接下来我们考虑两件事情，一件事情就是如果一个连通块本身不是一棵树，那么如果它的一部分点能被删掉，那这部分点肯定组成一个森林，每棵树肯定之靠着一个环。外面的点先被删，而环旁边的点是最后被删的，那我们相当于对于每棵树就可以以环旁

边那个点为根，然后 dp 上去。这个 dp 很方便，考虑到没两个点只会在其 LCA 上被合并一次，所以这部分的复杂度是 $\mathcal{O}(n^2)$ 的。

然后我们考虑这个连通块本身就是一棵树的情况，这个情况下其实就是每个点都是可以作为 dp 的根的，这样做出来的话如果全删完那显然是对的，但是发现如果不删完的话是会算重的。我们考虑不删完的一种方案，如果留了 x 个节点，那么对于留下来的这 x 个节点，每个节点作为根 dp 的时候都会把这种方案算一遍。于是其实就是这种方案对答案的贡献是 x 而不是 1 了。于是我们就可以直接把 dp 出来的结构除以 x 便使答案了。

最后每个连通块的合并直接暴力卷一下就可以了，别忘了乘个组合数。

3 AtCoder

3.1 AtCoder Grand Contest 023D Go Home

https://atcoder.jp/contests/agc023/tasks/agc023_d

数轴上有 n 个公寓，第 i 个公寓坐标在 x_i，住着 p_i 个人。现在有一辆“民主巴士”，上面坐着所有人，现在这辆巴士在位置 s。巴士在每个时刻的行驶方向由车上的乘客投票得到，少数服从多数，投票出现平局时，巴士会向数轴负方向行驶。每个居民都是最聪明的，他们都想要尽早到家，如果两个方向到家时间相同，他们会投给负方向。其中 $1 \le n \le 10^5, 1 \le s \le 10^9, 1 \le x_1 < x_2 < \cdots < x_N \le 10^9, 1 \le p_i \le 10^9$。

考虑最左边的公寓和最右边的公寓，我们发现谁先到达时看那边人多，因为就算最后只剩下他们俩了，那肯定也是多的那方赢，所以少的一方唯一的方法就是帮着多的一方尽快到达。

于是我们就可以让少的那一方全部搬家到多的那一方，然后就可以递归做下去了。

3.2 AtCoder Grand Contest 032E Modulo Pairing

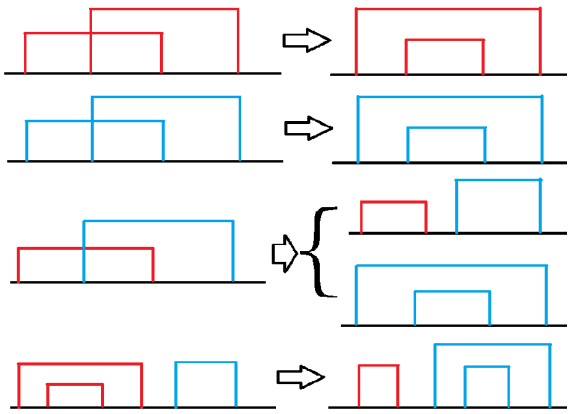
https://atcoder.jp/contests/agc032/tasks/agc032_e

给一个长度为 $2n$ 的序列，要求将其两两匹配成 n 组，假设第 i 组为 (x_i, y_i)，求 $\max_{i=1}^{n}(x_i + y_i) \mod m$ 的最小值，$1 \le n \le 10^5, 1 \le m \le 10^9, 0 \le a_i < m$。

我们首先我们有：

$$(x+y) \mod m = \begin{cases} x+y & x+y \le m \\ x+y-m & x+y > m \end{cases}$$

接下来我们看几种情况：



如果我们有 $x_1 \le x_2 \le x_3 \le x_4$，那么我们有这样 4 中情况：

第一种，$x_3 + x_4 < m$，那显然 $(x_1, x_4), (x_2, x_3)$ 更优。

第二种，$x_1 + x_2 \ge m$，和第一种一样。

第三种，$\begin{cases} x_1+x_3<m \\ x_2+x_4\geq m \end{cases}$，考虑到 $x_2+x_4-m<x_2$，故有 $\max\{x_1+x_3,x_2+x_4-m\}=x_1+x_3$，而考虑到 $\begin{cases} x_1+x_2\leq x_1+x_3 \\ x_3+x_4-m<x_3 \end{cases}$，故有 $\max\{x_1+x_3,x_2+x_4-m\}\geq \max\{x_1+x_2,x_3+x_4-m\}$。所以 $(x_1,x_2),(x_3,x_4)$ 一定比 $(x_1,x_3),(x_2,x_4)$ 优。

我们再来看 $(x_1,x_4),(x_2,x_3)$ 的情况。如果有 $\begin{cases} x_1+x_4\geq m \\ x_2+x_3\geq m \end{cases}$，那就有 $\begin{cases} x_1+x_4-m<x1 \\ x_2+x_3-m<x_2 \end{cases}$，所以 $\max\{x_1+x_4-m,x_2+x_3-m\}<x_2\leq\max\{x_1+x_2,x_3+x_4-m\}$，此时 $(x_1,x_4),(x_2,x_3)$ 的方案更优。但是如果有 $x_1+x_4\leq m$ 或是 $x_2+x_3\leq m$，则答案将会更大。

第四种，$\begin{cases} x_1+x_4<m \\ x_2+x_3<m \\ x_3+x_6\geq m \\ x_4+x_5\geq m \end{cases}$。根据前面的讨论，我们仅需比较 $(x_1,x_4),(x_2,x_3),(x_5,x_6)$ 和 $(x_1,x_2),(x_3,x_4),(x_5,x_6)$ 即可，不难发现 $\begin{cases} x_1+x_2\leq x_1+x_4 \\ x_3+x_6-m<x_3\leq x_2+x_3 \\ x_4+x_5-m\leq x_5+x_6-m \end{cases}$，于是 $\max\{x_1+x_2,x_3+x_6-m,x_4+x_5-m\}\leq\max\{x_1+x_4,x_2+x_3,x_5+x_6-m\}$。

根据第一、二、三种情况，我们发现，最终的划分结果肯定是被分成两部分，左边部分最小的匹配，次大的和次小的匹配……右边也这样匹配。其中左边所有的匹配 (x,y) 都有 $x+y<m$，右边的匹配 (x,y) 都有 $x+y\geq m$。

根据第四种情况，我们可以发现蓝色的匹配数越多越好，于是我们考虑二分蓝色的匹配数量，最终确定答案。

3.3 AtCoder Grand Contest 024D Isomorphism Freak

定义一棵树上两个点等价当且仅当以这两个点为根的有根树同构，一棵树的权值定义为这棵树的等价类个数。

给出一棵 n 个点的树，可以在这棵树基础不断的加叶子，使得最后树的权值最小，求出这个权值，以及满足之前条件的所有树中最小叶子数。

数据范围 $n\leq 100$。

我们发现其实最终所得的树只有两种，一种是以一条边为中心，左右两边同构，一种是以点为中心，呈放射状分布。

其实这两种都一样，以边为放射状的只要把这条边看成一个点就可以了。

这样我们考虑枚举中心，要求其权值，首先不难发现同构的数量其实就是以中心为根的树的深度。然后我们考虑叶子个数，我们只要想想构造这棵树的过程，其实就是把

每一层深度的所有点的儿子数补成相同，那么补完后树的叶子个数就是每层非叶子节点儿子个数的乘积，对于原树来讲就是每层儿子树最多的儿子个数的乘积。

复杂度 $\mathcal{O}(n^2)$，开 100 大概是为了不爆 `long long`。

3.4 AtCoder Grand Contest 029F Construction of a tree

https://atcoder.jp/contests/agc029/tasks/agc029_f

题意大概就是给你 $n-1$ 个点集，全集是 $\{1,2,\ldots,n\}$，现在要从每个点点集中抽出两个点来连边，最终形成一棵树。让你输出方案或判断无解。n 有 10^5，所有集合大小之和不超过 2×10^5。

考虑构造，儿子向父亲连边。也就是说，每个集合最后都是儿子 $\to$ 父亲的一条边，那么我们可以随便选一个点作为跟，其余点肯定对应一个集合，表示这个集合所对应的边，一端是他，另一端是他父亲。其实也就是一个匹配，左边是除了根节点以外的所有点，右边是所有点集。点与点集有边当且仅当这个点在点集中出现。

显然地，如果这个匹配无解那么最终肯定无解，因为根本找不到一个完美的儿子 $\to$ 父亲的映射。找到匹配之后我们把根加上去，跟前面的连法一样。如果连上之后这张图仍然不连通，那么显然无解，因为两个独立的连通块显然没有生成树。

那如果上面的判掉之后是否就一定有解了呢？我们考虑构造解。我们从根节点开始 `bfs`，我们考虑根节点所有连出去的集合，这些集合的匹配点的父亲都定作是这个根，然后这样递归下去。不难发现，这样做是肯定有解的。

考虑复杂度，边数 m 是集合的总大小，用 `Dinic` 做二分图匹配，复杂度 $\mathcal{O}(m\sqrt{n})$，之后的构造解复杂度 $\mathcal{O}(m)$，所以总复杂度 $\mathcal{O}(m\sqrt{n})$。

3.5 AtCoder Regular Contest 099F Eating Symbols Hard

https://atcoder.jp/contests/arc099/tasks/arc099_d

有一段标号从 $-\infty$ 到 $+\infty$ 的序列 A，初始是 $A_i=0$。现在有一个人在 0 位置，有一段操作序列 s，依次进行操作。

假设 i 时刻这个人在位置 p，如果 s_i 是 `'+'`，则将 A_p 加 1，如果是 `'-'`，则减 1；如果是 `'<'` 则该人左移一格，否则是 `'>'` 则右移一格。

要求数出 s 的所有连续子串中最终得到的 A 与 s 最终得到相同 A 序列的个数。

数据范围 $|s|\le 250000$。

考虑构造一波类似生成函数的东西 $f(x)=\sum_{i=-\infty}^{+\infty}A_ix^i$，另外还有一个指针 g，开始是 g 指向 0 位置，显然初始时 $f(x)=0$。

我们令 $f_i(x)$ 表示 $1\sim i$ 这一段子串操作所得到的 f 值，g_i 表示 $1\sim i$ 操作后的指针位置。考虑在末尾加入一个字符 c,如果 c 是 `'+'` 或 `'-'`,那显然 $\begin{cases} f_i(x)=f_{i-1}(x)\pm x^{g_i} \\ g_i=g_{i-1} \end{cases}$，

同理如果 c 是 '<' 或 '>'，那显然 $\begin{cases} f_i(x) = f_{i-1}(x) \\ g_i = g_{i-1} \mp 1 \end{cases}$。

如果要计算 $[l,r]$ 的 f 值，不难发现就是 $\frac{f_r(x)-f_{l-1}(x)}{g_{l-1}}$。

于是我们要算的就是 $f_n(x) = \frac{f_r(x)-f_{l-1}(x)}{g_{l-1}}$，考虑随便选一个 x，算出这个多项式的值然后哈希一下即可。

4 SPOJ

4.1 SPOJ TSUM Triple Sums

有一个长为 N 的序列 A，求每 3 个互不相同位置上的值的和有多少种取值，并对每种取值求出方案数。$n \leq 40000, |A_i|$ 20000。

暂且忽略 $i < j < k$ 的要求，构造多项式

$$A(x) = \sum_{1 \leq i \leq N} x^{A_i}$$

则 $A^3(x)$ 中 x^S 项的系数就是所求结果。

容斥原理

$$\begin{aligned} \left(\sum x\right)^3 &= \sum x^3 + 3\sum x^2 y + 6\sum xyz \\ \left(\sum x^2\right)\left(\sum x\right) &= \sum x^3 + \sum x^2 y \\ \left(\sum x^3\right) &= \sum x^3 \end{aligned}$$

即

$$\sum xyz = \frac{\left(\sum x\right)^3 - 3\left(\sum x^2\right)\left(\sum x\right) + 2\left(\sum x^3\right)}{6}$$

5 UVa

5.1 UVa 12633 Super Rooks on Chessboard

超级车可以攻击行、列、主对角线 3 个方向。$R \times C$ 的棋盘上有 N 个超级车，问被攻击的格子总数。$R, C, N \leq 50000$。

设 R, C, D 分别为行上、列上、对角线上有车的格子集合，结果即为 $|R \cup C \cup D|$。

设

$$R = r_1, r_2, \ldots, r_n$$
$$C = c_1, c_2, \ldots, c_m$$
$$D = d_1, d_2, \ldots, d_k$$

即计算 $r_i - c_j = d_k$ 的 (i, j, k) 数量。

构造多项式

$$R(x) = \sum_{i=1}^{n} x^{r_i}$$
$$C(x) = \sum_{i=1}^{m} x^{-c_i}$$

计算 $(R \cdot C)(x)$ 中 x^{d_k} 的系数。

6 其他

6.1 PKUWC 2018 Minimax

https://www.luogu.com.cn/problem/P5298

https://loj.ac/problem/2537/

有一棵 n ($n \leq 3 \times 10^5$) 个节点的二叉树，每个叶子节点有一个与众不同的权值，每个非叶子节点有一个概率 p_i，每个非叶子节点的权值有 p_i 的概率是子节点的最大值，$1-p_i$ 的概率是子节点的最小值。设 1 号节点的权值有 m 种可能性，第 i 小的权值为 v_i，概率为 d_i，求：

$$\sum_{i=1}^{m} i \cdot V_i \cdot D_i^2$$

你需要输出答案对 998244353 取模的值。

考虑朴素 n^2 dp，$f_{i,j}$ 表示到第 i 个节点权值为 j 的答案，维护一下前缀和可以直接转移。

$$f_{i,j} = f_{s,j} \times \left(\sum_{k=1}^{j} f_{t,k} \times p_i + \sum_{k=j+1}^{n} f_{t,k} \times (1-p_i) \right)$$

然后我们考虑线段树合并优化 dp，维护区间乘法标记，合并的时候维护前缀和和后缀和即可。

6.2 CERC 2017 Cumulative Code

https://codeforces.com/gym/101620/

给你一棵 k ($k \leq 30$) 阶的满二叉树，从上到下、从左到右从 1 开始编号，设 $\{p_n\}$ 是它的 Prüfer 序列。给你三个数 a, d, m，求 $\sum_{i=0}^{m-1} p_{a+id}$。

这题主要是要发现一个性质，就是考虑定义 $f_{i,k,h}(x)$ 表示以 x 为根，深度为 k 的子树，现在要开始删的是 Prüfer 序列中的第 i 位，对答案的共献，h 表示 x 是否有父亲，那么 $f_{i,k,h}(x)$ 可以表示成 $a_{i,k,h}x + b_{i,k,h}\lfloor \frac{x}{2} \rfloor + c_{i,k,h}$ 的形式。

令 Q 表示询问集合，考虑左儿子对答案的贡献为 $f_{i,k-1,1}(2x)$，也就是对父亲的贡献是 $2a_{i,k-1,1}x + b_{i,k-1,1}\lfloor \frac{2x}{2} \rfloor + c_{i,k-1,1} = (2a_{i,k-1,1} + b_{i,k-1,1})\,x + c_{i,k-1,1}$，即 $\{2a_{i,k-1,1} + b_{i,k-1,1}, 0, c_{i,k-1,1}\}$，右儿子同理可推出，为 $\{2a+b, 0, a+c\}$。

我们考虑记搜，记搜的答案直接存的是 a, b, c，这样我们就可以省掉 x 对答案的影响。我们发现 i 其实可以表示成到下一个询问的距离，也就是其 lower_bound，这样我们可以省一点空间。

但是我们发现还是需要 2^n 级别，我们考虑小范围暴搜，其余记搜，即 $k > 15$ 是不计入状态，然后如果递归到一棵子树发现它没有询问节点，那么直接 return 掉，这样就使得 i 小于等于子树 size 了。

其实 $h=0$ 我们也不用计入状态，因为不难发现 $h=0$ 只会搜索 $\mathcal{O}(k)$ 次，完全可以接受。

最后稍稍判一下边界即可。

这样深度小于等于 $\frac{k}{2}$ 的节点只有 $\mathcal{O}(2^{\frac{k}{2}})$ 个，深度大于 $\frac{k}{2}$ 的只有 $\mathcal{O}(k2^{\frac{k}{2}})$ 个，所以单次询问时空复杂度均为 $\mathcal{O}(k2^{\frac{k}{2}})$。

6.3 洛谷 P5631 最小 mex 生成树

https://www.luogu.com.cn/problem/P5631

给定 n 个点 m 条边的无向连通图，边有边权，现在你要求出一个这个图的生成树，使得其边权集合的 mex 尽可能小。

$1\le n\le 10^6, 1\le m\le 10^5, 0\le w\le 10^5$

考虑一个做法：枚举答案 x，把边权为 x 的所有边全部删掉，看看能不能构成一棵生成树。

建一棵以边权为下标的线段树，对于每条边权为 w 的边将其放到 $[0,w-1]$ 和 $[w+1,10^5]$ 上。然后在线段树上遍历，遍历到一个节点就将它上面挂着的边连上，这样子到叶结点时就连上了所有不包含该边权的边，用可撤销并查集判断图是否连通即可。

用可撤销按秩合并并查集，复杂度 $\mathcal{O}(n\log^2 n)$。并不是很懂为什么两只 log 能过 10^6。