

试题泛做

金策

题 1

试题编号	Codeforces 273E
试题名称	Dima and Game
题目大意	算法讨论
有 n 对整数 $(l_i, r_i) (1 \leq l_i < r_i \leq p)$ 。每次可以选择满足 $r_i - l_i > 2$ 的一对数 (l_i, r_i) ，将其替换为 $(l_i + \lfloor \frac{r_i - l_i}{3} \rfloor, l_i + 2 \lfloor \frac{r_i - l_i}{3} \rfloor)$ 或者 $(l_i, r_i - \lfloor \frac{r_i - l_i}{3} \rfloor)$ 。两人轮流操作，不能操作者输。给定 $n, p (n \leq 1000, p \leq 10^9)$ ，求满足先手必胜的初始局面数量。	关键在于处理 SG 函数， $SG(l_i, r_i) = g(r_i - l_i)$ ， g 定义为 $g(1) = g(2) = 0, g(x) = \text{mex} \{g(\lfloor \frac{x}{3} \rfloor), g(x - \lfloor \frac{x}{3} \rfloor)\} (x \geq 3)$ 。打表发现 $g(x)$ 构成若干个连续段，段数量仅有 100 左右，可以用二路归并进行预处理。处理完后可得知 $SG = 0, 1, 2$ 分别有几种情况。再用一个 DP 求出 n 个 SG 值异或和不为 0 的方案数即可。
时空复杂度	空间 $O(d)$ ，时间 $O(n + d)$ ，其中 d 表示段数，在 $p \leq 10^9$ 时 d 大约为 100。

题 2

试题编号	Codeforces 317C
试题名称	Balance
题目大意	算法讨论
一张无向图表示一个输水网络， $n (n \leq 300)$ 个顶点表示容器， e 条边表示管道。在任意时刻，每个容器的水量不能超过最大容量 v ，也不能小于 0。构造一个输水方案，对于每一个容器 i 将其水量由当前状态 a_i 变为目标状态 b_i 。输水的总步骤数不能超过 $2n^2$ 。	对于每一个联通块单独处理：取一棵生成树。每次操作时取一个叶子节点并将其 a_i 调整为与 b_i 相同（类似于网络流的增广操作，在树上进行 DFS 即可实现，所用到的步骤数量不会超过树的边数），然后删除这个叶子并对剩下的树重复上述操作。容易证明这个方法用到的步骤数量小于 $n^2/2$ 。
时空复杂度	空间 $O(n + e)$ ，时间 $O(n^2 + e)$ 。

题 3

试题编号	GCJ 2010 Final C
试题名称	Candy Store
题目大意	算法讨论
有 $k (k \leq 1000)$ 位顾客依次光顾你的商店，每个人想要买的糖果价值为区间 $[1, C] (C \leq 10^{12})$ 中的任意整数。你事先准备若干个任意整数价值的糖果盒，以保证无论顾客的需求如何，总是可以恰好满足每位顾客。求最少所需要的糖果盒数量。	假设准备的 m 个盒子价值为 $a_1 \leq a_2 \leq \dots \leq a_m$ 。考虑所有 k 名顾客均购买 $a_i - 1$ 价值的糖果的情况，可能被用到的盒子仅有 a_1, a_2, \dots, a_{i-1} 。于是 $\sum_{j=1}^{i-1} a_j \geq k(a_i - 1)$ ，即 $a_i \leq 1 + \lfloor \frac{1}{k} \sum_{j=1}^{i-1} a_j \rfloor$ 。另一方面，考虑每位顾客均买 C 价值糖果的情况，有 $\sum_{j=1}^m a_j \geq kC$ 。结合这两个式子可以用一个循环算法求得 m 的下界。对于上述构造出来的 m 个盒子，分配策略是每次取尽可能大的盒子分给顾客，直到恰好达到需求。可以用归纳法证明这个策略是可行的。
时空复杂度	空间 $O(1)$ ，时间 $O(\text{Answer}) = O(k \log C)$ 。

题 4

试题编号	Codeforces 251D
试题名称	Two Sets
题目大意	<p>算法讨论</p> <p>给定 $n(n \leq 10^5)$ 个非负整数 $a_i(a_i \leq 10^{18})$, 将其划分为两个集合 A, B, 令 s_0, s_1 分别表示 A, B 内数字的异或和。输出方案最大化 $s_0 + s_1$, 若有多种方案, 还需最小化 s_1。</p>
时空复杂度	<p>定义 01 变量 $x_i = [a_i \in B] (1 \leq i \leq n)$, 集合 $P_d = \{i : a_i \text{ 的第 } d \text{ 二进制位是 } 1\}$。</p> <p>对于第 i 位, 如果 P_i 为奇数, 则第 i 位对 $s_0 + s_1$ 的贡献一定是 2^i; 如果 P_i 为偶数, 则贡献是 0 或 2×2^i, 要使贡献较大则需要使得异或和 $\bigoplus_{j \in P_i} x_j = 1$。从高位往低位遍历 i 可以得到一系列异或方程, 每次将新得到的方程加入已有的方程组, 检查是否导出矛盾, 无矛盾则保留, 有矛盾则去除。由于 2 的幂次的性质, 这样的贪心顺序可以使得 $s_0 + s_1$ 最大。</p> <p>接下来最小化 s_1, 同理从高到低枚举 i, 每次检查 $\bigoplus_{j \in P_i} x_j = 0$ 是否可以满足。</p> <p>每次检查新加入的方程是否矛盾时, 可以将系数矩阵消元形成阶梯型矩阵 (可用 bitset 优化)。</p> <p>空间 $O(n)$, 时间 $O(nd^2)$, 其中 d 为 a_i 的二进制位数 (< 64)。若使用 bitset 优化则为 $O(\frac{n}{32}d^2)$。</p>

题 5

试题编号	Codeforces 249E
试题名称	Endless Matrix
题目大意	<p>算法讨论</p> <p>将全体正整数按漩涡状排成方阵, 如下所示:</p> <pre> 1 2 5 10 17 26 4 3 6 11 18 27 9 8 7 12 19 28 16 15 14 13 20 29 25 24 23 22 21 30 36 35 34 33 32 31 </pre> <p>记第 i 行第 j 列数字为 a_{ij}, 求 $\sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} a_{ij}$。需要判断答案是否多于 10 位数, 并且输出答案模 10^{10} 的结果。($x_2, y_2 \leq 10^9$)</p>
时空复杂度	<p>可以转化为求 $\sum_{i=1}^x \sum_{j=1}^y a_{ij}$。</p> <p>如果 $x \leq y$, 所求子矩阵的前 x 列组成正方形, 恰好包含 $1, 2, \dots, x^2$ 的整数, 很容易求和; 剩下的 $y - x$ 列, 每列的数字和可以写成关于 j 的二次式, 也容易求和。$x > y$ 的情况可类似处理。</p> <p>判断答案是否小于 10^{10}, 可以另取几个大于 10^{10} 的质数作为模进行检验。因为答案的大小不会超过 $(10^9)^4 = 10^{36}$ 级别, 若取两个大于 10^{13} 质数即可保证不出错。(即使只取一个质数, 出错概率也可忽略不计)。</p> <p>空间 $O(1)$, 时间 $O(1)$。</p>

题 6

试题编号	Codeforces 241D
试题名称	Numbers
题目大意	<p>算法讨论</p> <p>给定一个 $1, 2, \dots, n$ 的全排列, 从中划去任意项, 使得剩下的数列 (非空) 满足: 异或和为 0;</p> <p>从左到右将数拼接成的十进制数模 p 为 0。($n, p \leq 50000$)</p> <p>如果有解需要输出方案。</p>
时空复杂度	<p>考虑动态规划, $f[i][j][k]$ 表示前 i 个数, 异或和为 j, 组成数字模 p 为 k, 是否可行。由此得到一个比较显然的 $O(n^2p)$ 算法。</p> <p>注意到当 n 比较大时, 无解的概率很低。取 $n = 31$ 并将数列中小于 32 的数取出进行 DP, 即可 AC。事实上, $1 \sim 31$ 有 $2^{26} - 1$ 个非空子集满足元素异或和为 0, 有解的概率为 $P = 1 - \left(\frac{p-1}{p}\right)^{2^{26}-1}$, 而 $(P^{31!})^{50000}$ 几乎等于 1, 所以这样做是很可靠的。</p> <p>空间、时间 $O(\min\{31, n\}^2 p)$。</p>

题 7

试题编号	Codeforces 319E
试题名称	Ping-Pong
题目大意	算法讨论
区间 (a, b) 向 (c, d) 连一条有向边当且仅当 $c < a < d$ 或 $c < b < d$ 。支持两种操作（操作数量 $\leq 10^5$ ）： (1) 插入一个区间，保证插入区间的长度按照插入顺序严格递增。 (2) 询问从第 i 个区间能否经过有向边到达第 j 个区间。	可以相互到达的区间 I 组成强连通分量 c 。维护当前所有强连通分量组成的集合 C ，并用新区间 $I_i = (a_i, b_i)$ 来代表每一个强连通分量 $c_i \in C$ ，其中 $a_i = \min\{I.l : I \in c_i\}, b_i = \max\{I.r : I \in c_i\}$ 。每次插入一个新区间 $I' = (p, q)$ 的时候，可以证明，如果 $a_i < p < b_i$ 或 $a_i < q < b_i$ ，则 I' 应该可以和 c_i 中的所有区间相互到达。所以只需查询所有这样的 c_i 并与 I' 合并。 为了高效查询，建立一棵线段树并对每个结点维护一个 vector/链表，记录严格包含此结点的区间有哪些。
时空复杂度	空间 $O(n \log n)$ ，时间 $O(n \log n)$ 。

题 8

试题编号	Codeforces 286E
试题名称	Ladies' Shop
题目大意	算法讨论
k 种物品（每种物品有无限个）的质量分别为 $1 \leq p_1 < p_2 < \dots < p_k \leq m$ 。任意取一个或几个物品组合可凑成的所有总质量中，不超过 m 者组成集合 $A = \{a_1, a_2, \dots, a_n\}, a_i \leq m$ 。已知 m 和 a_1, a_2, \dots, a_n ，求出一组符合条件的 p_1, \dots, p_k 并使得 k 最小，或者判断无解。	如果存在 i, j ，使得 $a_i + a_j \leq m$ ，但 $a_i + a_j \notin A$ ，则无解；否则必然有解。 对于某个质量 a_x ，如果不存在 i, j 使得 $a_x = a_i + a_j$ ，则必然存在某个物品的质量等于 a_x 。取出所有这样的 a_x 即为答案。 需要统计的内容具有卷积的形式，可用 FFT 实现。
时空复杂度	空间 $O(m)$ ，时间 $O(m \log m)$ 。

题 9

试题编号	Codeforces 306C
试题名称	White, Black and White Again
题目大意	算法讨论
共有 w 件好事和 b 件坏事（每件事都被视为不同的）在 n 天内发生，每天至少发生一件事（同一天发生的多件事是要计次序的）。这 n 天恰被分成 3 部分，每部分至少包含 1 天，第一、第三部分只发生好事，第二部分只发生坏事。求总方法数量。（ $n, w, b \leq 4000$ ）	设第二部分持续 i 天，容易知道答案为 $w!b! \sum_{i=1}^{n-2} (n-i-1) \binom{b-1}{i-1} \binom{w-1}{n-i-1}$ $= w!b!(w-1) \sum_i \binom{b-1}{i-1} \binom{w-2}{n-i-2}$ $= w!b!(w-1) \binom{w+b-3}{n-3}$ 。
时空复杂度	空间 $O(1)$ ，时间 $O(n \log n)$ （若在线性时间预处理逆元，可优化到 $O(n)$ ）。

题 10

试题编号	Codeforces 274C
试题名称	The Last Hole!
题目大意	算法讨论
白色平面上有 n 个点，以它们为圆心画黑色实心圆，在 t 时刻所有圆的半径为 t 。一个洞定义为封闭的白色区域。判断有无产生过洞，并求最后一个洞消失的时刻。（ $n \leq 100$ ）	考虑洞消失前一瞬间所在的位置，可以证明它是某个锐角三角形的外心或某个矩形的中心。枚举所有锐角三角形的外心和矩形的中心（共有 $O(n^3)$ 个这样的点），再 $O(n)$ 求出其他点到这个点的最小距离以判断它是否已经被覆盖，若没有被覆盖则这个点可以产生洞，更新答案。
时空复杂度	空间 $O(n)$ ，时间 $O(n^4)$ 。

题 11

试题编号	Codeforces 331C3
试题名称	The Great Julya Calendar
题目大意	算法讨论
一个整数初始值为 n ，每次可以选择 n 某一位上出现的数 x ，并将 n 减去 x 。求最少几次操作可以将 n 变成 0。 ($n \leq 10^{18}$)	容易证明每次需要取当前各个数位上最大的数作为减数 x ，然后可以用动态规划解决。一种做法是考虑把数字 $\overline{a_1 a_2 \cdots a_t 99 \cdots 9 p}$ 消成 $\overline{a_1 a_2 \cdots a_{t-1} (a_t - 1) 99 \cdots 9 q}$ 所需的步骤数量，记为 $f[m][k][p]$ ，其中 $m = \max a_i$ ， k 为 9 的数量，同时需要记录所到达的 $q[m][k][p]$ 。
时空复杂度	空间 $O(l \times 10^2)$ ，时间 $O(l \times 10^3)$ ，其中 l 为 n 的位数。

题 12

试题编号	Codeforces 235C
试题名称	Cyclical Quest
题目大意	算法讨论
一个小写英文字符串 s ，每次询问 s 有多少个子串是和字符串 t_i 循环同构的。(询问总长和串长 $\leq 10^6$ ，询问个数 $\leq 10^5$)	对 s 建后缀自动机，拿 t_i 在自动机上跑。为了处理循环同构，首先需要将 t_i 的首字符删去，删去首字符时，检查当前结点的父亲的长度值是否恰为 $ t_i - 1$ ，是则退回到父亲，否则原地不动。然后将删去字符添加到 t_i 末尾，只要从当前结点沿着对应字母边往下走一步即可。 为避免重复计数，需要计算 t_i 的循环节，可以用 KMP 实现。
时空复杂度	空间 $O(26n)$ ，时间 $O(n)$ 。

题 13

试题编号	Codeforces 280D
试题名称	k-Maximum Subsequence Sum
题目大意	算法讨论
一个整数数列。支持： (1) 单点修改 (2) 询问 l, r 间取出至多 k 段连续子段的最大和。 ($n, m \leq 10^5, k \leq 20$ ，其中询问的数目 ≤ 10000)	此询问可以转化为一个费用流。由于图的特殊性，求最长路的过程相当于询问区间最大子段和，一次增广的过程相当于区间全部乘以 -1 ，可用线段树优化。
时空复杂度	空间 $O(n)$ ，时间：单点修改每次 $O(\log n)$ ，询问每次 $O(k \log n)$ 。

题 14

试题编号	Codeforces 293B
试题名称	Distinct Paths
题目大意	算法讨论
$n \times m$ 的方格，每个格子可以填入 k 中颜色中的一种。满足从左上角到右下角的任意路径（路径只能往右或往下，只能走有边相邻的格子）中不包含相同颜色的格子。 一些格子的颜色已经固定，求把剩下格子填满有几种方案。($k \leq 10$)	已知 $n + m - 1 \leq k$ 。直接搜索会 TLE。考虑到许多方案是等价的（可以通过颜色重编号到达另一个），所以可以仅搜索字典序最小的那一些，再分别计算它们可以代表的不同方案数量来更新答案。这样优化后即能 AC。
时空复杂度	空间 $O(nm)$ ，时间 $O(\text{本质不同的方案数量} \times nm)$ 。

题 15

试题编号	Codeforces 306D	
试题名称	Polygon	
题目大意	算法讨论	
输出一个凸 n 边形, 满足每个内角相等, 但边长度两两不等。 ($n \leq 100$)	易知 $n = 3, 4$ 时无解。 内角相等的凸 n 边形可以由正 n 边形将边适当平移而得。所以可以用各种各样的方法平移每条边使得边长两两不等。	
时空复杂度	空间 $O(n)$, 时间 $O(n)$ 。	

题 16

试题编号	Codeforces 241B	
试题名称	Friends	
题目大意	算法讨论	
n 个数字 a_1, \dots, a_n 两两异或所得的 $n(n-1)/2$ 个数中, 前 k 大的和是多少。 ($n \leq 50000, k \leq n(n-2)/2$, 数字 $\leq 10^9$)	建 01-Trie, 对于每个结点 u 还要额外记录 $count_u[i]$, 表示 u 子树的叶子所表示的数字中, 第 i 位上共有几个 1。在树上维护 n 个指针 (对每个 a_i 各维护一个, 用来统计 a_i 和其他所有数字异或的信息), 并沿树边同时不断下降, 下降时统计对答案的贡献。	
时空复杂度	空间 $O(n \log^2 n)$, 时间 $O(n \log^2 n)$ 。	

题 17

试题编号	Codeforces 235D	
试题名称	Graph Game	
题目大意	算法讨论	
一个 n 个点的基环 + 外向树。每次从当前连通分量中随机取一个点作为中心, 代价为当前分量的总点数, 然后去掉中心并对剩下的连通分量递归处理。求总代价的期望。 ($n \leq 3000$)	对于每一对点 (u, v) , 考虑以 u 为中心时, v 属于其所在连通分量中的概率。如果 u, v 间路径不经过环上的边, 则概率为 $\frac{1}{dis(u, v)+1}$, 其中 $dis(u, v)$ 表示 u, v 路径上的边数量; 如果经过环上的边, 则有三种情况, 一种是以 u 为中心时环依然完整, 另外两种是此时环只剩下某半边, 讨论一下即可。	
时空复杂度	空间 $O(n)$, 时间 $O(n^2)$ 。	

题 18

试题编号	Codeforces 335D	
试题名称	Rectangles and Square	
题目大意	算法讨论	
平面上 n 个边平行坐标轴的矩形, 互不相交。求是否有某些矩形恰好铺满一个正方形。 $n \leq 10^5, 0 \leq \text{坐标} \leq 3000$	拼成的正方形有四条边缘线; 其左上角顶点必然为某个矩形的左上角, 右下角顶点亦然。对于每个点预处理出它往上下左右的边缘线能延伸多远。对于每个左上角顶点, 计算出它往右下方向延伸成正方形的可能边长范围 (既要保证范围内没有空格子, 又要保证顶点往右、往下的边缘线没有被阻断); 对于每个右下角顶点, 同样往左上方向进行处理。如果某矩形的左上角顶点和某矩形的右下角顶点都在对方所需的边长范围之内, 则铺满了一个正方形, 可以对于每条左上-右下对角线分别用线段树维护所需信息。	
时空复杂度	空间 $O(m^2 + n)$, 时间 $O(m^2 + n \log m)$ 。	

题 19

试题编号	Codeforces 263E	
试题名称	Rhombus	
题目大意	算法讨论	
一个 $n \times m$ 的数阵, 给定 k , 对于 $k \leq x \leq n - k + 1, k \leq y \leq m - k + 1$ 求 $f(x, y) = \sum_{i=1}^n \sum_{j=1}^m a_{ij} \cdot \max(0, k - i - x - j - y)$ 的最大值。($n, m \leq 1000$)	所求的和形状为一个斜正方形, 中间的数字权重大。可以暴力求出 $f(k, k)$ 后往右、往下递推以求得所有 $f(x, y)$, 递推时需要剪掉一个三角形内的和, 加上另一个三角形内的和。可以通过预处理三角形形状的前缀和来高效计算。	
时空复杂度	空间 $O(nm)$, 时间 $O(nm)$ 。	

题 20

试题编号	Codeforces 342D	
试题名称	Xenia and Dominoes	
题目大意	算法讨论	
一个 $3 \times n$ 的棋盘, 有一些坏格, 和一个指定留空的格子。把剩下的格子用 1×2 骨牌铺满, 且至少有一个骨牌与空格相邻并指向空格。问方案数。 $n \leq 10000$	空格周围至多只有 3 个位置能够放下骨牌并且指向它, 可以用容斥原理解决, 然后转化为经典的状压 DP。	
时空复杂度	空间 $O(n \times 2^3)$, 时间 $O(2^3 \times 4^3 n)$ 。	

题 21

试题编号	Codeforces 319D	
试题名称	Have You Ever Heard About the Word?	
题目大意	算法讨论	
一个长度为 n 的字符串, 每次选择串中最短的重子串, 即形如 XX 的子串 (如果有多个最短的则取最左的), 将其替换为 X 。求最终不能继续操作时, 剩下的字符串。($n \leq 50000$)	容易证明删除的顺序一定是按长度递增的。升序枚举 l , 对于 X 串长度为 l 的情况, 可以用 $O(n/l \cdot \log n)$ 的时间找到所有发生替换的位置, 具体方法是: 对于 $i \in \{1, 1+l, 1+2l, \dots\}$, 计算 i 和 $i+l$ 的最长公共前缀、最长公共后缀, 如果其长度之和 $\leq l+1$, 则此处出现了重子串。而最长公共前 (后) 缀可以用哈希 + 二分 $O(\log n)$ 求得。于是判断的总复杂度是 $O(\log^2 n)$ 。如果发生替换, 则字符串会缩短, 需要用 $O(n)$ 时间重构哈希数组。可以证明重构的次数是 \sqrt{n} 级别的。	
时空复杂度	空间 $O(n)$, 时间 $O(n\sqrt{n} + n \log^2 n)$ 。如果用平衡树维护哈希数组, 可以做到 $O(\log^3 n)$ 。	

题 22

试题编号	Codeforces 285E	
试题名称	Positions in Permutations	
题目大意	算法讨论	
一个 $1 \sim n$ 的排列 p , 如果 $ p_i - i = 1$ 则称 i 是好位置。求恰好有 k 个好位置的排列个数。($n \leq 1000$)	先考虑按从小到大的顺序填入数字, 且只填在好位置上的方案数。DP 状态为 $f[i][j][a][b]$, 表示已填入的数字为 $1 \sim i$ 中的 k 个且都填在好位置上的方案数, a, b 分别表示 p_i, p_{i+1} 是否已经被占据。令 $g[j] = \sum_{a,b} f[n][j][a][b]$, 则剩下 $n - j$ 个数。可以看出一个恰有 k 个好位置的排列对于 $g[j](n - j)!$ 的贡献为 $\binom{k}{j}$, 据此可以求出答案。	
时空复杂度	空间 $O(n^2)$, 时间 $O(n^2)$ 。	

题 23

试题编号	Codeforces 339E
试题名称	Three Swaps
题目大意	算法讨论
一个排列初始时为 $1, 2, \dots, n$ ，经过不超过 3 次操作，每次操作是将 $[l_i, r_i]$ 左右翻转。给出操作后的排列，求出操作方案。 $(n \leq 1000)$	经过 3 次操作后至多把排列分成 7 段，所以可以以段为单位进行爆搜。注意有时候一个连续段是需要被拆开的，需要特别在分段时判断并多分出一段。
时空复杂度	空间 $O(n)$ ，时间 $O(n) + O(\text{爆搜})$ 。

题 24

试题编号	Codeforces 261E
试题名称	Maxim and Calculator
题目大意	算法讨论
一个整数对 (a, b) ，初始时为 $(1, 0)$ 。可以用两种操作：(1) $(a, b) \rightarrow (a, b + 1)$ ；(2) $(a, b) \rightarrow (ab, b)$ 。求在区间 $[l, r]$ 中有多少整数 x 满足：可以通过不超过 k 次操作，使得数对的第一个数变为 x 。 $(r \leq 10^9, k \leq 100)$	易知所变成的 x 的最大质因子不超过 $k - 1$ ，而这样的数在 10^9 范围内约有 3×10^6 个。求出所有这些数字后进行 k 次 DP，每次只要从小到大扫描更新，即可求出每个数所需要的最少步骤数。
时空复杂度	空间 $O(N)$ ，时间 $O(kN)$ ， N 为 r 以内最大质因子不超过 $k - 1$ 的数的数量，在极限情况约为 3×10^6 。

题 25

试题编号	Codeforces 333C
试题名称	Lucky Tickets
题目大意	算法讨论
给定 k ，在一个 8 位的 $0 \sim 9$ 数字串中任意插入 $+, -, \times, ()$ ，使得结果为 k 。输出 m 个不同的有解的数字串。 $0 \leq k \leq 10^4, m \leq 3 \times 10^5$	条件很宽松，有各种各样的构造方法。我的方法是把串的前四位（或后四位）设为 $k + d (d \leq 200)$ ，剩下四位放一个能凑成 d 的数串，这样的串可以通过爆搜得到。这个方法对每个 $k \leq 10^4$ 都能构造出 3×10^5 个解。
时空复杂度	空间 $O(200 \times 10000)$ ，时间 $O(10000 \times 2^4 \times 2^3 + m)$ 。

题 26

试题编号	Codeforces 301C
试题名称	Yaroslav and Algorithm
题目大意	算法讨论
一个程序每个指令形如 $a_i >> b_i$ 或 $a_i <> b_i$ ，其中 a_i, b_i 是串（可为空），串包含数字或问号。初始时输入一个数字 x ，每次循环时找到最小的 i 使得 x 包含 a_i ，并将 x 中出现 a_i 的地方替换为 b_i ，如果是 $>>$ 则继续下一次循环，如果是 $<>$ 则停止。如果找不到这样的 i 也停止。停止后输出当前的数字。构造一个程序，使得输入要求的数字后，程序输出它加一后的数。	用？作为指针，先把？移到最低位后，每次如果？之前的数字不为 9，则加一后停止；否则进位并将？左移一格。
时空复杂度	空间，时间 $O(1)$ 。

题 27

试题编号	Codeforces 338D
试题名称	GCD Table
题目大意	算法讨论
一个 n 行 m 列的表格, 第 i 行第 j 列为 $\gcd(i, j)$ 。给出 k 个数 a_1, \dots, a_k , 问它们是否在表格某一行中从某一位置开始按顺序连续出现。	令 $L = \text{lcm}(a_1, \dots, a_k)$, 可以证明如果在表格中出现, 则必然在第 L 行出现过。所以若 $L > n$ 则无解。 设 a_1 出现在第 L 行第 x 列, 则 $\gcd(L, x+i-1) = a_i$, 从而有 $x \equiv 1-i \pmod{a_i}$, 解这个线性同余方程组求得 x 。然后代回去进行检验。
时空复杂度	空间, 时间 $O(1)$ 。

题 28

试题编号	Codeforces 261D
试题名称	Maxim and Increasing Subsequence
题目大意	算法讨论
一个长为 n 的正整数列 b_1, \dots, b_n 重复 t 次得到一个长为 $n \times t$ 的数列 a_1, \dots, a_{nt} , 求新数列的最长严格上升子序列长度。($n, \max b \leq 10^5, t \leq 10^9, n \times \max b \leq 2 \times 10^7$)	不妨设 b_i 的值共有 tot 种, 那么 LIS 长度至多为 tot , 所以可以令 $t \leftarrow \min\{t, tot\}$ 。 然后考虑 DP, 令 $f_i[j]$ 表示由 a_1, \dots, a_i 中元素组成的, 且最大元素小于 j 的最长上升子序列长度。对 i 从 1 遍历到 nt 同时更新 f 。注意到 f 数组总是保持单调不降的, 更新的时候可以在适当地方 break, 使得每次更新都有某个 $f[k]$ 的值增加, 所以更新总次数不超过 $tot \times tot$ 。
时空复杂度	空间 $O(n)$, 时间 $O(n \cdot tot)$ 。

题 29

试题编号	Codeforces 257E
试题名称	Greedy Elevator
题目大意	算法讨论
模拟一个 n 个人、 m 层楼的电梯系统。已知每个人前来等待的时间, 以及每个人的起始楼层和终止楼层。电梯根据目前上面正在等待的人、电梯里想往上的人数总和与下面正在等待的人、电梯里想往下的人数总和进行比较以决定下一个时间是往上还是往下走。电梯初始时在 1 层。求出每个人到达终止楼层的时间。 ($n \leq 10^5, m \leq 2 \times 10^5$, 前来的时间 $\leq 10^9$)	有三种事件: 有人前来等待、有人上电梯、有人下电梯。每次取最近将要发生的一个事件进行处理。第一种只要顺序排序, 后两种分别用 set 维护即可。处理完后需要更新上、下的人数, 并求出下一个时间电梯的运行方向。
时空复杂度	空间 $O(n)$, 时间 $O(n \log n)$ 。

题 30

试题编号	Codeforces 316E3
试题名称	Summer Homework
题目大意	算法讨论
一个长度为 n 的数列 a , 支持三种操作: (1) 单点赋值 (2) 区间加 (3) 对区间 $[l, r]$ 询问 $\sum_{x=0}^{r-l} f_x \cdot a_{l+x} \pmod{10^9}$, 其中 $f_0 = f_1 = 1, f_k = f_{k-1} + f_{k-2}$ 。($n, m \leq 2 \times 10^5$)	令矩阵 $\mathbf{A} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$, 则 $(f_{k-1} \ f_k) \mathbf{A} = (f_k \ f_{k+1})$ 。对于线段树的每个结点 $[l, r]$ 维护这样一个 1×2 矩阵 $a_l (f_0 \ f_1) + a_{l+1} (f_1 \ f_2) + \dots + a_r (f_{r-l} \ f_{r-l+1})$ 。
时空复杂度	空间 $O(n)$, 时间 $O(n \log n)$ 。

题 31

试题编号	Codeforces 266E
试题名称	More Queries to Array...
题目大意	算法讨论
一个长度为 n 的数列 a , 支持两种操作: (1) 区间赋值 (2) 对区间 $[l, r]$ 和 k , 询问 $\sum_{x=l}^r a_x \cdot (x-l+1)^k \pmod{10^9+7}$ ($0 \leq k \leq 5, n, m \leq 10^5$)	由于 k 很小, 可以在线段树的每个结点 $[l, r]$ 中对每个 $0 \leq k \leq 5$ 维护 $\sum_{x=l}^r a_x \cdot x^k$ 的值。在区间覆盖的时候需要用到 k 次方和公式。 询问时根据二项式定理, $\sum_{x=l}^r a_x \cdot (x-l+1)^k = \sum_{x=l}^r a_x \cdot (x^k + \binom{k}{1}x^{k-1}(1-l) + \binom{k}{2}x^{k-2}(1-l)^2 + \dots)$
时空复杂度	空间 $O(kn)$, 时间 $O(kn \log n)$ 。

题 32

试题编号	Codeforces 309D
试题名称	Tennis Rackets
题目大意	算法讨论
一个正三角形, 每条边被 n 个分点等分为 $n+1$ 段, 其中前 m 个和后 m 个点是不能使用的。剩下的分点中, 从每条边上各取一个, 能连成的钝角三角形有几个。($n \leq 32000, 0 \leq m \leq \lfloor n/2 \rfloor$)	将取的三个点在各自边上的位置设为 a, b, c , 利用余弦定理可以表示出连成的三角形边长 x, y, z , 使得 $x^2 + y^2 < z^2$ 即为钝角三角形 (z 为长边)。枚举 a, b 即可得到 c 的取值范围, 累加答案。 由于 n 较大, 需要进行常数优化才能 AC。比如, 枚举 a, b 的过程中, c 的范围是有单调性的, 求范围的时候可以用直接枚举的方法来避免除法运算。另外, x, y 为最长边的情形和 z 一样, 可以省去 $1/3$ 的运算量。枚举钝角顶点时, 左右具有对称性, 可以再省去一半的运算量。
时空复杂度	空间 $O(1)$, 时间 $O(n^2)$ 。

题 33

试题编号	Codeforces 235E
试题名称	Number Challenge
题目大意	算法讨论
求 $\sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^c d(ijk)$, $d(x)$ 表示 x 的正约数个数。($a, b, c \leq 2000$)	考虑动态规划, 逐一考虑每个质数 p , 令 $f[a][b][c][p] = \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^c d(ijk)[i, j, k \text{ 中只含不超过 } p \text{ 的质因子}]$ 。 $f[a][b][c][p] = \sum_{x,y,z} f[a/p^x][b/p^y][c/p^z][last(p)](x+y+z+1)$
时空复杂度	空间、时间大约是 $O(\sum_{p \text{ prime}} (\sqrt{a} \log_p a)^3)$ 。

题 34

试题编号	Codeforces 305E
试题名称	Playing with String
题目大意	算法讨论
初始时一个纸带上写着字符串。每次可以任取一个纸带 s , 若存在位置 i 使得 $s[i-1] = s[i+1], 1 < i < s $, 则可以将纸带剪成三段: $s[1..i-1], s[i], s[i+1.. s]$ 。两人轮流操作, 不能操作者为输。问先手能否胜利, 如果胜利, 输出先手第一步可行的操作位置。	将原串中满足 $s[i-1] = s[i+1]$ 的位置 i 记作 ok, 剩下的位置非 ok。则串中组成了许多个 ok 连续段, 这些段之间是相互不影响的。对于一个长度为 k 的 ok 连续段, 记它的 SG 函数为 $g[k]$, 则 $g[-1] = g[0] = 0, g[k] = \text{mex}_i \{g[i-2] \text{ xor } g[k-i-1]\}$ 。
时空复杂度	空间 $O(n)$, 时间 $O(n^2)$ 。

题 35

试题编号	Codeforces 325E	
试题名称	The Red Button	
题目大意	算法讨论	
n 个按钮从 0 到 $n-1$ 编号。第一次和最后一次都要按 0。若上一次按了 x ，这一次只能按 $2x$ 或 $2x+1$ (模 n 意义下)。除了 0 按两次外，其他按钮都恰好按一次。构造方案或判断无解。($n \leq 10^5$)	若 n 为奇数则无解 (考虑 $0, n-1$ 的前驱)。若 n 为偶数时，注意到 x 或 $x+n/2$ 的下一步可以为 $2x$ 或 $2x+1$ 。把按钮当作有向边，建 $n/2$ 个结点从 0 到 $n/2-1$ 编号。第 i 个按钮代表的边为 $(\lfloor i/2 \rfloor, i \bmod n/2)$ 。新图连通，且每个点的入度出度均为 2，于是存在一条欧拉回路，即为所求的方案。	
时空复杂度	空间 $O(n)$ ，时间 $O(n)$ 。	

题 36

试题编号	Codeforces 260E	
试题名称	Dividing Kingdom	
题目大意	算法讨论	
平面上有 n 个点，用两条水平线和两条垂直线将平面分成 9 个区域 (线不能恰好穿过某点)，使得每个区域内的点数恰好为 $a[1], \dots, a[9]$ 。($n \leq 10^5, 坐标 \leq 10^9$)	共有 $9! = 362880$ 种分配方案，对每一种判断是否可行。判断时先根据点数找到四条直线的位置，然后检验每个区域内的点数量是否符合，检验时可以利用函数式线段树。	
时空复杂度	空间 $O(n \log n)$ ，时间 $O(n \log n + 9! \times \log n)$ 。	

题 37

试题编号	Codeforces 338E	
试题名称	Optimize!	
题目大意	算法讨论	
给定 $n, len, h, b[1..len], a[1..n]$ ，有多少个 i ，使得 $1 \leq i \leq n-len+1$ ，且 $b[1..len]$ 和 $a[i..i+len-1]$ 间可以建立完全匹配且每条匹配边上的两个数之和 $\geq h$ 。($len \leq n \leq 150000, 1 \leq a[i], b[i], h \leq 10^9$)	令 $a'[i] = h - a[i]$ ，则对于每条匹配边 (u, v) 都需要有 $b[u] \geq a'[v]$ ，则匹配策略显然是小的配小的、大的配大的。在数轴上标出所有 $a'[i], b[i]$ 的位置， $a'[i]$ 权值为 1， $b[i]$ 权值为 -1，则如果每个位置的前缀和都非负，则存在符合条件的匹配。遍历每一个 $a[i..i+len-1]$ ，并用线段树维护前缀和的最小值。	
时空复杂度	空间 $O(n)$ ，时间 $O(n \log n)$ 。	

题 38

试题编号	Codeforces 332D	
试题名称	Theft of Blueprints	
题目大意	算法讨论	
给定正整数 k 和一个 n 个点的带边权无向图，满足对于任意一个 k 元点集 S ，存在且仅存在一个点 u ，使得 u 与 S 中每个点都有连边，并定义这个点集 S 的权值为这 k 条边的权值之和。求所有 k 元点集 S 的权值的平均值。($n \leq 2000$)	对于某个点 u ，其对应的 S 有 $\binom{deg[u]}{k}$ 个，考虑这些集合的权值之和，其中每条与 u 相连的边都产生了 $\binom{deg[u]-1}{k-1}$ 的贡献。累加即可。	
时空复杂度	空间 $O(n^2)$ ，时间 $O(n^2)$ 。	

题 39

试题编号	Codeforces 311E
试题名称	Biologist
题目大意	算法讨论
<p>n 只狗, 初始时第 i 只狗性别 $gender[i]$, 将第 i 只狗变性需要花费 $v[i]$。</p> <p>有 m 个人, 第 i 个人有一个性别 $x[i]$, 并指定了 $k[i]$ 只狗。如果这 $k[i]$ 只狗的性别都是 $x[i]$, 你会收益 $w[i]$; 否则, 若第 i 个人是你的朋友你需要支付 g, 若不是则无需支付。求最大收益。($n \leq 10^4, m \leq 2000, k[i] \leq 10$)</p>	<p>如果一个人是你的朋友, 只需将其 $w[i] += g$ 即可等同非朋友处理, 所以假设每个人都是非朋友。考虑最小割模型。S 集合的狗为公, T 集合的狗为母。连边:</p> <p>(1) 源点 s 向每只公狗连边 $v[i]$; 每只母狗向汇点 t 连边 $v[i]$。</p> <p>(2) 每个喜欢公狗的人向其指定的每只狗连边 ∞, 源点 s 向这个人连边 $w[i]$; 每个喜欢母狗的人, 由他指定的每只狗向他连边 ∞, 并由他向汇点 t 连边 $w[i]$。</p> <p>求出最小割 c, 则 $-c + \sum w[i]$ 即为最大收益。</p>
时空复杂度	(最大流) 点数 $O(n+m)$, 边数 $O(n+km)$ 。

题 40

试题编号	Codeforces 360D
试题名称	Levko and Sets
题目大意	算法讨论
<p>有两列数 $a[1..n], b[1..m]$ 和一个质数 p, 对于每个 a_i 可以生成一个集合 $\{a_i^b \bmod p : b = k_1 b_1 + \dots + k_m b_m, k_j \geq 0\}$。求这 n 个集合的并集的元素个数。</p> <p>($1 \leq n \leq 10^4, 1 \leq m \leq 10^5, 2 \leq p \leq 10^9, 1 \leq a_i < p, 1 \leq b_i \leq 10^9$)</p>	<p>令 $d = \gcd(b_1, \dots, b_m, p-1)$, 则 a_i 生成的集合即为 $\{a_i^{kd}\}$。</p> <p>令 g 为模 p 的原根, $a_i = g^{t_i}, \gcd(t_i, p-1) = q_i$, 则 a_i 生成的集合即为 $\{g^{kq_i d}\}$。对于每个 a_i 只需要知道对应的 q_i 即可, 枚举 $p-1$ 的约数 x, 求出其中满足 $a_i^x \equiv 1 \pmod{p}$ 的最小 x, 则 $q_i = (p-1)/x$。</p> <p>令 $s = (p-1)/d, r_i = \gcd(q_i, s)$, 则问题转化为: 给定的 r_i 均为 s 的约数, 求出在 $0, 1, \dots, s-1$ 中有多少数是某个 r_i 的倍数。考虑容斥原理, 其表达式中每一项均为一个系数乘以 s 的某个约数, 用 $f_i[j]$ (j 是 s 的约数) 表示仅考虑 r_1, \dots, r_i 时, 容斥原理表达式中 j 的系数是多少。由 $f_i[]$ 数组和 r_{i+1} 可以求出 $f_{i+1}[]$ 数组。</p>
时空复杂度	空间 $O(n+m+d(p-1))$, 时间 $O(\sqrt{p} + n \log p + md(p-1) \log p + d(p-1)^2 \log p)$ 。 $d(x)$ 表示 x 的约数数量。

题 41

试题编号	Codeforces 286D
试题名称	Tourists
题目大意	算法讨论
<p>两个人在 q 时刻同步从数轴原点出发, 以 1 的速度朝正方向走, 两人中间会出现墙将两人隔开。共 m 座墙, 第 i 座墙在 t_i 时刻突然出现, 位置是区间 $[l_i, r_i]$ 的连续一段。求出两人被隔开的总时间。</p> <p>现在有 n 个询问 q_i, 对于每个询问, 回答两人出发时刻为 q_i 情况下的答案。($n, m \leq 10^5, l_i, r_i, t_i, q_i \leq 10^9$)</p>	<p>将数轴离散化成几个互不相交的段, 求出每一段内最早出现墙的时刻, 这可以用线段树或者排序 + 堆实现。</p> <p>对于某一段墙, 位置在 $[l, r]$, 出现时刻为 t, 如果 $q+r \leq t$ 则不会看到这堵墙, 如果 $q+l > t$ 则会被这堵墙完整地挡住 (遮挡时间为 $r-l$), 否则只会有 $q+r-t$ 的时间被遮挡。于是答案是一个关于 q 的分段线性函数, 将所有斜率发生变化的点排序, 然后扫一遍即可回答所有询问。</p>
时空复杂度	空间 $O(n+m)$, 时间 $O((n+m) \log(n+m))$

题 42

试题编号	Codeforces 258D
试题名称	Little Elephant and Broken Sorting
题目大意	算法讨论
一个 n 排列 $a[1..n]$, 进行 m 次操作, 每次操作是以 $1/2$ 的概率交换 $a[x_i], a[y_i]$, 以 $1/2$ 的概率不动。求最后序列的逆序对数量的期望。 ($n, m \leq 1000$)	令 $f[i][j]$ 表示 $a[i] > a[j]$ 的概率。进行操作 x, y 后, $f[x][y] \leftarrow f[y][x] \leftarrow 1/2, f[k][x] \leftarrow f[k][y] \leftarrow (f[k][x] + f[k][y])/2, f[x][k] \leftarrow f[y][k] \leftarrow (f[x][k] + f[y][k])/2$ ($k \neq i, j$)。最后答案即为 $\sum_{i < j} f[i][j]$ 。
时空复杂度	空间 $O(n^2)$, 时间 $O(n^2)$

题 43

试题编号	Codeforces 273D
试题名称	Dima and Figure
题目大意	算法讨论
一个 $n \times m$ 的方格板, 从中选出一个连通 (边相连) 块, 使得块内任意两格相互到达 (只能经过连通块内的格子) 所需的最短距离恰为它们的曼哈顿距离。求方案数。 ($n, m \leq 150$)	容易发现, 满足所选取的连通块在每一行、列中只出现最多一段连续段即可。按列的顺序 DP, 则每行的左端点是先不增后不降, 右端点是先不降后不增。令 $f[i][l][r][p][q]$ 表示当前第 i 行取的连续段为 $[l, r]$ 时的方案数, 其中 p 表示左端点是否增加过, q 表示右端点是否减少过。转移时需要用二维前缀和进行优化。
时空复杂度	空间 $O(m^2)$ (滚动数组), 时间 $O(nm^2)$

题 44

试题编号	Codeforces 283E
试题名称	Cow Tennis Tournament
题目大意	算法讨论
n 头奶牛有互不相同的能力值, 能力值高的能战胜低的。它们两两比赛后得到一张表。进行 k 次操作, 每次把所有能力值均在 $[l_i, r_i]$ 的两只奶牛的比赛的赛果在表格上反转。求操作完后, 表格上有多少个三元组 (a, b, c) 使得 a 战胜 b , b 战胜 c , c 战胜 a 。 ($n, k \leq 10^5$)	记每头牛的出度为 $d[i]$, 只需求出所有 $d[i]$ 即可, 容易证明答案 $= \binom{n}{3} - \sum_{i=1}^n \binom{d[i]}{2}$ 。 假设要知道 $a, b (a < b)$ 间的比赛结果, 只需统计有多少个操作区间 $[l_i, r_i]$ 将 a, b 完全包含在内, 根据其奇偶性即可判断。所以, 假设我们要对每个 i 统计 i 连向 $j (j > i)$ 的边数, 只要将操作区间按左端点排序, 从左到右枚举 i , 同时对 i 的右边维护一棵线段树, 支持区间求和、区间取反操作。接着反过来用同样方法统计 i 连向 $j (j < i)$ 的边数。于是即可求出所有 $d[i]$ 。
时空复杂度	空间 $O(n + k)$, 时间 $O((n + k)(\log n + \log k))$

题 45

试题编号	Codeforces 332E
试题名称	Binary Key
题目大意	算法讨论
有两个字符串 p, s 和整数 k , 需要求出一个长度为 k 的 01 串 q 且字典序尽量小, 使得用 p 和 q 能生成 s 。生成算法是将所有满足 $0 \leq i < p , q[i \bmod q] = 1$ 的 $p[i]$ 按顺序拼接得到 s 。($ p \leq 10^6, s \leq 200, k \leq 2000$)	首先对 p 串预处理, 将所有隔 k 个字符取出而得到的子序列的 hash 值都求出来。枚举串 q 中 1 的出现次数 t , 然后类似地求出所有 s 串中隔 t 个字符取出而得到的子序列的 hash 值。这样我们有两组 hash 值 $P[0..k-1]$ 和 $S[0..t-1]$, 它们之间对应相等的位置即为所求 q 串中 1 的位置。为了让 q 字典序尽量小, 从后往前贪心匹配即可。
时空复杂度	空间 $O(p + s + k)$, 时间 $O(p + k(k + s))$

题 46

试题编号	Codeforces 253E
试题名称	Printer
题目大意	算法讨论
n 个打印任务, 每个任务进队时间 t_i , 页数 s_i , 优先级 p_i 。优先级两两不同。每个单位时间打印机只能打印一页纸。打印机会从当前队列中挑选优先级最高的任务进行打印。 现在有某个任务 x 的优先级是未知的, 但是已知该任务完成的时刻 T_x 。恢复出它的优先级, 并求出所有任务的完成时刻 T_i 。 ($n \leq 50000, t_i, s_i, p_i \leq 10^9$)	显然 x 的优先级越高, 其完成时间越早。只要二分 x 的优先级 p_x 即可。用一个优先队列模拟任务的执行。
时空复杂度	空间 $O(n)$, 时间 $O(n \log^2 n)$

题 47

试题编号	Codeforces 243C
试题名称	Colorado Potato Beetle
题目大意	算法讨论
网格平面上每个格子有坐标。初始时位于 $(0,0)$, 进行 n 次移动, 每次往上下左右四个方向之一移动 x_i 格。求移动途中经过的格子与被这些格子包围住的格子总共有几个。 ($n \leq 1000, x_i \leq 10^6$)	将所有关键格的坐标排序进行离散化, 离散化后边长是 $O(n)$ 级别的, 格数是 $O(n^2)$ 级别的。从边缘往里 floodfill 即可。
时空复杂度	空间 $O(n^2)$, 时间 $O(n^2)$

题 48

试题编号	Codeforces 264E
试题名称	Roadside Trees
题目大意	算法讨论
有 n 个位置可以种树, 树每个月增长 1 米。每个月初进行一次操作, 操作有两种: (1) 在一个空位置种树, 初始树高为 $[1, 10]$ 内的整数; (2) 砍掉从左到右第 x 棵 (未被砍掉的) 树, ($x \leq 10$)。被砍了树的位置不再种树; 保证任意时刻没有相等的树高。共 m 次操作, 每次操作后输出当前最长上升子序列。 ($n \leq 10^5, m \leq 2 \times 10^5$)	倒数第 i 次操作中种树时将初始树高加上 i , 即可处理树的生长。考虑维护 $f[i]$, 表示为以 i 位置起始的 LIS 长度。 每次在 i 位置插入树高 h 时, 当前小于 h 的数的数量不超过 10 个, 可能位于 i 的左边和右边。所有位于 i 右边的树中, 除了这些树之外, 都可以用来更新 $f[i]$, 能够用来更新的树组成了不超过 10 个连续段, 可以通过不超过 10 次区间查询得到; 求出 $f[i]$ 后, 要更新左边小于 h 的树, 也只需更新不超过 10 次。 删除第 x 个数时, 可以把第 $1 \sim x$ 的数字全部删除, 再按 $x-1, x-2, \dots, 1$ 的顺序插回来。 为实现上述做法, 只要用两棵线段树维护 $f[]$, 分别按照下标和数值建树。
时空复杂度	空间 $O(n+m)$, 时间 $O(10m(10+\log m+\log n))$

题 49

试题编号	Codeforces 303D
试题名称	Rotatable Number
题目大意	算法讨论
给定 n, x , 求出最大的 $b < x$, 使得在 b 进制下存在长度为 n 的循环数。(比如十进制下的 142857 是循环数) ($1 \leq n \leq 5 \times 10^6, 2 \leq x \leq 10^9$)	有一个结论, 存在循环数当且仅当 $n+1$ 是质数, 且 b 是模 $n+1$ 的原根。 从 $x-1$ 向下枚举, 直到找到一个 $n+1$ 的原根即可。
时空复杂度	空间 $O(d(n))$, 时间 $O(\sqrt{n} + (x - \text{Ans})d(n) \log n)$

题 50

试题编号	Codeforces 295D
试题名称	Greg and Caves
题目大意	算法讨论
$n \times m$ 的方格, 存在 $1 \leq l \leq r \leq n$, 使得满足 $l \leq i \leq r$ 的第 i 行恰有两个黑格, 其余行全部是白格。存在 $t, l \leq t \leq r$, 使得第 i 行被夹在黑格间的部分是第 $i+1$ 行的子集 ($l \leq i < i+1 \leq t$), 第 $i+1$ 行被夹在黑格间的部分是第 i 行的子集 ($t \leq i < i+1 \leq r$)。求方案数量。 ($1 \leq n, m \leq 2000$)	所求形状可以被 t 分成上下部分, 两部分具有同样的结构。先只考虑其中一部分, $f[i][l]$ 表示底 (最长边) 位于第 i 行, 底边长度为 l 的方案数, DP 时需要维护 $f[i][l]$ 和 $l \cdot f[i][l]$ 的前缀和。然后考虑把两部分拼接起来得到总方案数。注意避免重复计数, 需要将分割行 t 的位置定义为最小的满足条件的 t 。
时空复杂度	空间 $O(n^2)$, 时间 $O(n^2)$

题 51

试题编号	Codeforces 240F
试题名称	TorCoder
题目大意	算法讨论
一个长度为 n 小写英文字符串 s 。进行 m 次操作, 每次对子串 $s[l..r]$ 进行操作, 将其字符重排使得 $s[l..r]$ 是回文串, 且其字典序尽量小。如果不能排成回文串则忽略此次操作。输出最后得到的串。 ($n, m \leq 10^5$)	能重排成回文串, 当且仅当区间内出现奇数次的字母不超过一种。只需建立线段树, 每个结点储存该区间内 26 个字母分别有几个。重排时, 要使字典序尽量小, 必然从小到大排成一段段从小到大的字母, 段数不超过 26×2 , 分别进行区间覆盖即可。
时空复杂度	空间 $O(26n)$, 时间 $O(26n \log n)$

题 52

试题编号	Codeforces 323B
试题名称	Tournament-graph
题目大意	算法讨论
构造一个有 n 个结点的竞赛图, 使得对任意两个结点 u 和 v ($u \neq v$), 从 u 到 v 的最短距离不超过 2。 ($3 \leq n \leq 1000$)	可以验证 $n = 4$ 时无解, $n = 3, 6$ 时有解。假如 $n = k$ 时有解, 则 $n = k+2$ 时也有解, 只要 $k+1 \rightarrow k+2, k+2 \rightarrow i, i \rightarrow k+1 (i \in [1, k])$ 即可。所以除了 $n = 4$ 外均能构造出解。
时空复杂度	空间 $O(n^2)$, 时间 $O(n^2)$

题 53

试题编号	Codeforces 264D
试题名称	Colorful Stones
题目大意	算法讨论
两个 RGB 串 s, t , 初始时两个人分别站在两个串第一个字符。每次下指令为 R,G,B 中的一个, 当前脚下为该字符的人要向前走一步。不能走到串外面。求有多少种状态时可以达到的。 ($ s , t \leq 10^6$)	可以通过两遍扫描, 得到 $l[i], r[i]$, 表示当 s 串的人在第 i 位时, t 串的人可能的最左位置和最右位置。对所有 i 累加答案即可。另外要注意当两个人走过的最后两个字符分别为 xy 和 yx 时 (x, y 是不同字符), 此状态也是不可达的。需要将多算的部分从答案中减去。
时空复杂度	空间 $O(n+m)$, 时间 $O(n+m)$

题 54

试题编号	Codeforces 314E
试题名称	Sereja and Squares
题目大意	算法讨论
给定一个长度为 n 的含有问号和小写字母的串，在问号处填上小写字母或大写字母，使得串能成为一个合法的括号序列，每个左括号为小写字母，其对应的右括号需要为相同的大写字母。求方案数模 2^{32} 。 ($n \leq 100000$)	稍加转化即为给定左括号和空格的串，求补充为合法括号序列的方案数。令 $f[i][j]$ 表示压入前 i 个括号，栈中还剩 j 个左括号的方案数，即可方便转移。稍微压一下常数可以 AC。
时空复杂度	空间 $O(n)$ (滚动数组)，时间 $O(n^2)$

题 55

试题编号	Codeforces 351D
试题名称	Jeff and Removing Periods
题目大意	算法讨论
对于一个数字序列，每次可进行的操作为：删除一个下标呈等差数列的子序列，且这些数字都相等；然后将剩下的数任意重新排列。这个序列的美好度为将其删完所需的最小步骤数量。 给定一个长度 n 的序列， q 次询问，每次询问区间 $[l, r]$ 组成的序列的美好度。 ($n, q \leq 10^5$)	问题可转化为两个子问题 (1) 求区间内不同数字的种类数 (2) 区间内是否有某种数字的分布呈等差数列。 预处理时记录每个位置的数字的上一次出现位置 $last[i]$ 。将询问按照右端点排序。从左往右枚举右端点同时回答询问，为维护第 (1) 个问题的答案，只要对 $(last[i], i]$ 区间加 1；第 (2) 个问题也可以处理，只要记录每种数字当前的公差，以及使得区间内该数字呈等差数列的左端点的最左位置，插入数字时检查等差数列是否能延伸，若不能则要改为新的公差。 由于操作都是区间加，询问都是单点查询，可以用树状数组实现。
时空复杂度	空间 $O(n)$ ，时间 $O(n \log n)$

题 56

试题编号	Codeforces 335F
试题名称	Buy One, Get One Free
题目大意	算法讨论
n 件物品各自有价格，买一件物品的同时可以顺便免费拿走一件价格严格小于它的物品。要得到所有物品最小花费是多少。 ($n \leq 500000, v \leq 10^9$) ($n, q \leq 10^5$)	关键在于处理相同价格的物品。预处理时将物品按价格降序排序，并求出每种价格的物品有几个。从高到低枚举价格，同时维护一个小根堆，堆的元素个数为当前已经配成的对子的数量，每个元素的含义为配成一个对子，或者拆开一个对子后配成两个对子，所得到的收益。尝试给当前价格的物品配对时，每次取出堆中最小元素，和当前价格比较以决定采取哪种方式，并将收益压入堆中。最后堆中的元素之和即为获得的总收益。
时空复杂度	空间 $O(n)$ ，时间 $O(n \log n)$

题 57

试题编号	Codeforces 293E
试题名称	Close Vertices
题目大意	算法讨论
统计树上边数不超过 l ，且边权和不大于 w 的路径条数。 ($n \leq 10^5, w \leq 10^9$)	点分治。统计时将子树中的点按照到根路径上的边权和排序，然后扫描，扫描的同时利用树状数组统计边数不超过 i 的点有几个。同一个子树内多算的部分要减去。
时空复杂度	空间 $O(n)$ ，时间 $O(n \log^2 n)$

题 58

试题编号	Codeforces 277D
试题名称	Google Code Jam
题目大意	算法讨论
一场 GCJ 共有 n 题, 时长为 t 。每题有小数据、大数据, 小数据、大数据分别所需要得解题时间、得到的分数是已知的, 大数据有一定的概率 FST。对于同一题, 只有做了小数据才能做大数据。罚时指最后一次正确提交的时刻。求最大期望分数和此时的最小期望罚时。 $(n \leq 1000, t \leq 1560)$	假设已知选择那些题, 将其按 $p_i \cdot t_{large_i} / (1 - p_i)$ 升序排即可知道最优顺序。将所有题目按此排序后, 就是一个背包问题, 背包过程中要同时记录最大期望分数和对应的最小期望罚时。
时空复杂度	空间 $O(n + t)$, 时间 $O(n \log n + nt)$

题 59

试题编号	Codeforces 321D
试题名称	Ciel and Flipboard
题目大意	算法讨论
一个 $(2x - 1) \times (2x - 1)$ 的矩阵 $A[1..2x - 1][1..2x - 1]$, 每次可以把其中一个 $x \times x$ 的子矩阵内的数字同时乘上 -1 。求经过操作后, 矩阵数字总和的最大值是多少。 $(n \leq 33)$	用 $a[i][j] = -1$ 或 1 表示是否以 (i, j) 为子矩阵左上角进行过操作, 共有 $x \times x$ 个可以操作的位置。令 $b[i][j] = \prod_{1 \leq p \leq i, 1 \leq q \leq j} a[p][q]$ 。枚举 $b[1][x], b[2][x], \dots, b[x][x]$ 后, 可以用 $O(x^2)$ 时间求出此情况下的最大总和。
时空复杂度	空间 $O(x^2)$, 时间 $O(2^x \times x^2)$

题 60

试题编号	Codeforces 238E
试题名称	Meeting Her
题目大意	算法讨论
一个 n 个点的有向图, 要乘车从 S 点到 T 点。你知道这张图的样子, 也知道有 k 种车, 每种车都会不断地发车, 其中第 i 种车每次发车时从 s_i 到 t_i 随意选择一条最短路 (每条边的长度是 1) 出发。你上车时知道这辆车是第几种车, 你可以选择任意站点下车。求最坏情况下你至少要乘几次车, 若最坏情况下无法到达则输出 -1 。 $(n, k \leq 100)$	考虑动态规划, $f[u][i]$ 表示当前位于第 i 种车上, 车目前停在点 u , 到达 T 需要再乘几次车; $f[u][0]$ 表示当前在点 u 的地面上, 到达 T 需要乘几次车。则 $f[u][i] = \max_{v \in NEXT(u, i)} \{ \min(f[v][i], f[v][0]) \}$ $f[u][0] = \min_{u \in MUST(s_i, t_i)} \{ 1 + f[u][i] \}$ $f[T][0] = 0$, 答案为 $f[S][0]$ 。关键在于转移顺序, 可以按照当前已知的 $f[u][i]$ 从小到大进行更新 (类似 Dijkstra)。
时空复杂度	空间 $O(n^2)$, 时间 $O(n^3)$

题 61

试题编号	Codeforces 269D
试题名称	Maximum Waterfall
题目大意	算法讨论
一些水平线段, 其中最顶上和最底下分别有两条无限长的线段。高的线段 a 往低的线段 b 连边的条件是, a, b 的水平投影有长度大于 0 的重叠, 且不存在高度在 a, b 之间的线段 c 使得 a, c, c, b 也满足该条件, 连边的边权为线段水平投影重叠的长度。求最顶到最底的一条路径, 使得路径上最小权最大。 $(n \leq 10^5, l_i , r_i \leq 10^9)$	将所有端点排序, 从左到右扫描, 同时维护一个以高度为关键字的 set。遇到 x 的左端点时, 将此线段 x 插入 set, 并找到它上、下最近的线段 p, q , 连边 $(p, x), (x, q)$, 并删除边 (p, q) ; 遇到 x 的右端点时, 从 set 中将 x 删除。可以看出连边数量的级别是 $O(n)$ 的。 二分边权, 判断从顶到底是否连通即可。
时空复杂度	空间 $O(n)$, 时间 $O(n \log n)$

题 62

试题编号	Codeforces 266D
试题名称	BerDonalds
题目大意	算法讨论
一个带边权的连通无向图，找出一点（可以在顶点上或边上任意一点），使得它到所有顶点的最短路中最长的那一条最短，求出这个距离。（ $n \leq 200$ ）	先用 Floyd 预处理两两最短路长，然后枚举每条边进行计算。假设点在边 (i, j) 上，与 i 点距离为 x ，那么答案即为 $\max_{k=1}^n \min \{x + d[i][k], w[i][j] - x + d[j][k]\}$ ，它的图象是许多个倒 V 字形放在一起，然后取最上面的轮廓线。排序后，用一个栈维护即可求出这条折线，并用它的所有极小值点来更新答案即可。
时空复杂度	空间 $O(n^2)$ ，时间 $O(n^3 \log n)$

题 63

试题编号	Codeforces 325D
试题名称	Reclamation
题目大意	算法讨论
有一个 $r \times c$ 的地图，把左边界和右边界粘起来使得形成一个圆柱，现在要不断地挖去其中的格子，要求任何时候都存在一条从最上方到最下方的路径（四连通），如果某次操作不满足要求则不做，问 n 次操作中最后有多少次操作是成功的。（ $r, c \leq 3000, n \leq 300000$ ）	不存在从上到下的四连通路径，当且仅当存在一条八连通路径绕过了圆柱面。 将地图复制一份变成 $r \times 2c$ 的地图，如果存在一条 (x, y) 到 $(x, y + c)$ 的八连通路径，则这条路径绕过了圆柱面。每次插入时只要用并查集判断是否存在即可。
时空复杂度	空间 $O(rc)$ ，时间 $O(n\alpha(n))$

题 64

试题编号	Codeforces 268D
试题名称	Wall Bars
题目大意	算法讨论
一个长度为 n 的仅由 1,2,3,4 组成的数串 s 。可以从 $s[1..h]$ 中的任意一个开始，每次可以往右跳到距离不超过 h 且数字相同的位置，最后能到达 $s[n - h + 1..n]$ 中的任意一个。有多少种这样的数字串？（ $n \leq 1000, h \leq \min(h, 30)$ ）	考虑动态规划， $f[i][d_1][d_2][d_3][d_4]$ 表示当前在第 i 位，上一次出现的数字 k 与 i 的距离为 d_k 。如果某一次某个数字的出现间隔大于 h ，那么这个数以后就作废了。这样的状态数是 $n \cdot h^4$ 的。 注意到如果 4 个数字都没有被作废，那么总有一个 $d_k = 0$ ， k 是 $s[i]$ ；如果有数字作废，那么剩下至多 3 种数字。所以状态数可以缩减到 $n \cdot h^3$ 。
时空复杂度	空间 $O(n \cdot h^3)$ 或 $O(h^3)$ ，时间 $O(n \cdot h^3)$

题 65

试题编号	Codeforces 317E
试题名称	Princess and Her Shadow
题目大意	算法讨论
坐标系中，有些格子被树占据，剩下为空地。初始时公主和影子的位置不同，公主往上下左右移动时，影子也向相应方向移动，除非影子被树阻挡。求一个移动方案使得最后公主和影子位于同一个格子。（树的数量 ≤ 400 , 初始坐标 ≤ 100 ）	如果公主和影子的初始位置无法连通，则无解。否则一定可以构造解： 取一条公主到影子初始位置的最短路，并沿着它行走；如果影子也跟着发生移动，就将最短路径相应延伸。如果这样下去不能抓住，则一定会走到树林的外部，这时只要利用横/纵坐标最大/小的那棵树，将影子堵住并抓住即可。
时空复杂度	空间 $O(x^2)$ ，时间 $O(x^2)$

题 66

试题编号	Codeforces 309B	
试题名称	Context Advertising	
题目大意	算法讨论	
一篇文章有 n 个单词，取出其中连续一段排版，使得不超过 r 行，每行不超过 c 个字符（包括隔开单词的空格）。求最多能取出几个单词。（ $n, r, c \leq 10^6$, 文章字符数 $\leq 5 \times 10^6$ ）	对于每个单词，求出以它为一行的开头时，这一行能往右取到多远。这可以用两个指针扫描得到。 为了求经过 r 条边后到达的位置，可以用倍增处理。	
时空复杂度	空间 $O(n \log n)$ 或 $O(n)$ ，时间 $O(n \log n)$	

题 67

试题编号	Codeforces 241F	
试题名称	Race	
题目大意	算法讨论	
一个矩形地图，每个格子或者是交叉口，或者属于道路。道路都是横平竖直的。走过每个格子所需的时间已知。告诉你起点、终点和中间经过的交叉口，问 k 分钟后你的位置。（ $n, m \leq 100, k \leq 100000, len \leq 1000$ ）	因为道路的两端都是交叉口，所以从一个交叉口直接走到另一个交叉口，而不经其他交叉口的方案只能是直线行走，所以直接模拟即可。	
时空复杂度	空间 $O(nm + len)$ ，时间 $O(nm + len + k)$	

题 68

试题编号	Codeforces 329D	
试题名称	The Evil Temple and the Moving Rocks	
题目大意	算法讨论	
在 100×100 的方阵中放置石头，每个格子至多一个石头，石头可以有上下左右四种方向。你一开始可以激活一个石头使它按自己的方向移动，直到撞到另一个石头时，它会激活这个石头，并使自己停下来。如果一个石头发生移动而撞到另一个时，会放出声音。构造至少发出 10^5 次声音的方案。	第一行形如 $\rightarrow \rightarrow \rightarrow \rightarrow \dots \rightarrow \cdot \rightarrow$ $\cdot \rightarrow \downarrow$ ，第二行形如 $\uparrow \downarrow \leftarrow \cdot \leftarrow$ $\dots \leftarrow \leftarrow \leftarrow \leftarrow$ ，以此类推，激活左上角以后，依次绕过每一行，最后从第一列底部传递上来，形成循环。	
时空复杂度	无	

题 69

试题编号	Codeforces 267C	
试题名称	Berland Traffic	
题目大意	算法讨论	
一个有源点、汇点的流网络，边可以双向流，且有容量。满足任意两点间，任何路径所经过的权值之和都是一样的。求出最大的流。（ $n \leq 100$ ）	给每个点指定一个高度，则两点间边权即为高度之差。以高度为未知数，根据每个非源汇点满足流守恒的性质，列出方程。可以先给源、汇分别指定 0,1 的高度，然后进行高斯消元。这样可以求出边权之间的比例，然后根据每条边不超过容量的约束即可求得。	
时空复杂度	空间 $O(n^2)$ ，时间 $O(n^3)$ 。	

题 70

试题编号	Codeforces 249D	
试题名称	Donkey and Stars	
题目大意	算法讨论	
第一象限有 n 个点。固定两个角度。从原点开始，按角度向第一象限射出两条射线，可以跳到射线夹住范围内的某一点。然后可以以这个点为原点继续操作，直到无法移动为止。求最多跳过几个点。 $(n \leq 10^5)$		通过一些坐标变换可以将夹角转为直角，于是每次跳到的点需要横、纵坐标都比原来大，这是一个经典的 LIS 问题，可以用树状数组/单调栈二分方法解决。
时空复杂度	空间 $O(n)$ ，时间 $O(n \log n)$ 。	

题 71

试题编号	Codeforces 254D	
试题名称	Rats	
题目大意	算法讨论	
$n \times m$ 网格中，有的格子是墙，有的格子是空格，有的空格内有老鼠。在空格可以放炸弹，能攻击到最短距离（不能穿墙）不超过 d 的格子。放两个炸弹炸死所有老鼠，求出两个炸弹的位置。 $(n, m \leq 1000, d \leq 8)$		一个炸弹能攻击到的格子数大约是 $2d^2$ ，一个老鼠能被影响到的格子数也是 $2d^2$ 级别。任意取一个老鼠，枚举能炸到它的格子，以它作为第一颗炸弹；然后从剩下没死的老鼠中任取一个，类似地枚举第二颗炸弹，然后只要判断有没有全部炸死即可。
时空复杂度	空间 $O(nm + d^2)$ ，时间 $O(nm + d^6)$ 。	

题 72

试题编号	Codeforces 335E	
试题名称	Counting Skyscrapers	
题目大意	算法讨论	
n 座摩天楼，每座摩天楼高度为 i 的概率是 2^{-i} ，若大于 h 则只算作高度为 h 。Bob 从左往右每次走最高的 zipline，假设高度为 i ，zipline 中间不能被别的楼（高度 $\geq i$ 的）挡住，然后给答案加上 2^i 。给定 Bob 的答案，求 n 的期望；给定 n ，求 Bob 答案的期望。 $(n \leq 30000, h \leq 30)$		可以证明，给定 Bob 的答案，则 n 的期望即为这个值。若给定 n ，从底往上考虑每一高度的 zipline 对答案产生的贡献，产生一个高度 i 的 zipline 的条件是两端高度 $\geq i$ ，中间高度 $< i$ 。产生这个 zipline 后要拿掉中间所有 $i-1$ 高度的 zipline。然后可以推出答案的式子。
时空复杂度	空间 $O(1)$ ，时间 $O(nh)$ 。	

题 73

试题编号	Codeforces 341E	
试题名称	Candies Game	
题目大意	算法讨论	
n 个非负整数，和不超过 10^6 。每次可以选择两个数 i, j ，满足 $a[i] \leq a[j]$ ，将其改为 $2a[i], a[j] - a[i]$ 。求一个方案，使得最终恰好只有两个数不为 0。 $(n \leq 1000)$		对于任意三个数，总可以将其中一个消成 0。假设这三个数为 $a < b < c$ ，令 $q = \lfloor b/a \rfloor$ ，将 q 表示为二进制数。每次令 a 翻倍：如果 q 的当前位是 1，则用 b 提供给 a ，否则用 c 提供给 a 。这样操作完后 b 变成了 $b \bmod a$ 。于是 a, b, c 中最小者至少减少 1。重复操作即可将最小者变为 0。
时空复杂度	空间 $O(n)$ ，时间 $O(n + 10^6 \cdot \log(10^6))$ 。	

题 74

试题编号	Codeforces 241E	
试题名称	Flights	
题目大意	算法讨论	
一个有向无环图，给每条边的权值赋值为 1 或 2，使得点 1 到 n 的任意路径的长度相等。 $(n \leq 1000, m \leq 5000)$	每个点都有一个确定的距离 $d[u]$ ，对于任意边 (u, v) 需满足 $d[u] + w(u, v) = d[v]$ ，即 $1 \leq d[v] - d[u] \leq 2$ ，这是一个经典的差分约束系统的模型。	
时空复杂度	空间 $O(n + m)$ ，时间 $O(nm)$ 。	

题 75

试题编号	Codeforces 323C	
试题名称	Two permutations	
题目大意	算法讨论	
给定两个 $1 \sim n$ 的排列。每次给出 l_1, r_1, l_2, r_2 ，询问在第一个排列中位置在 $[l_1, r_1]$ ，且在第二个排列中位置在 $[l_2, r_2]$ 的数字有几个。 $(n, m \leq 10^6, \text{强制在线})$	转化为询问区间内有多少数字在 $[l, r]$ 之间，可以用经典的可持久化线段树解决。	
时空复杂度	空间 $O(n \log n)$ ，时间 $O((n + m) \log n)$ 。	

题 76

试题编号	Codeforces 311C	
试题名称	Fetch the Treasure	
题目大意	算法讨论	
一串格子，长度 h 。每次可以从格子 1 开始往右走 k 步。有些格子有宝藏，宝藏有价值度。 有 3 种操作： (1) 增加一种步长 x 。每次可以选择往右的步数属于自己当前拥有的步长集合（初始时步长集合为 $\{k\}$ ）。 (2) 减少某格宝藏的价值。 (3) 询问当前可达格子集合中，价值度最大的格子，并将这个格子的宝藏清除。 其中操作 (1) 共 m_1 次，操作 (2)(3) 共 m_2 次。 $(h \leq 10^{18}, k \leq 10^4, x \leq h, m_1 \leq 20, m_1 + m_2 \leq 10^5)$	(2)(3) 操作是优先队列的经典操作。 因为 k 很小，所以考虑对每个 $0 \leq i < k$ ，维护可达的第一个模 k 为 i 的格子 $dis[i]$ 。 每次插入新的步长时，用最短路算法更新 $dis[]$ 数组。并重新建立优先队列。	
时空复杂度	空间 $O(n + k)$ ，时间 $O((n + m_2) \log n + m_1(n \log n + k \log k))$ 。	

题 77

试题编号	USACO Dec 06	
试题名称	Cow Patterns	
题目大意	算法讨论	
给定一个 n 长度数列和一个 k 长度的 pattern，求出 pattern 在数列中所有匹配的连续段的位置。匹配的含义是：pattern 中相同的数对应数列中相同的数，不同的数对应数列中不同的数，且保持相对大小关系。 $(n \leq 10^5, k \leq 25000, 1 \leq \text{值域} S \leq 25)$	类似于 KMP，首先对 pattern 自处理得到 $next$ 数组，然后跑原串。唯一有区别的是“相等”的含义，需要改为比较在当前串中的 $rank$ ，这在值域小时可以通过暴力统计小于它、不大于它的有几个，在值域大时需要用树状数组。	
时空复杂度	空间 $O(n + k + S)$ ，时间 $O((n + k)S)$ 或 $O((n + k) \log S)$ 。	

题 78

试题编号	USACO Open 07	
试题名称	Connect	
题目大意	算法讨论	
一个 2 行 c 列的网格图，相邻点才有可能连边。每次删边/加边，询问两点 $(r_1, c_1), (r_2, c_2)$ 之间连通性（只能在 $[c_1, c_2]$ 范围内走动。 ($c \leq 15000$, 操作次数 ≤ 50000)	因为只能在 $[c_1, c_2]$ 范围内走动，所以连通性和此区间外的信息无关系。可以考虑用线段树维护连通性，结点 $[l, r]$ 维护一个 2×2 矩阵，表示左边的上下两点和右边的上下两点的连通性。	
时空复杂度	空间 $O(c)$ ，时间 $O(c + m \log c)$ 。	

题 79

试题编号	USACO Open 07	
试题名称	Best Cow Line, Gold	
题目大意	算法讨论	
一个英文字符串，每次可以从头或尾取一个字符删掉，并输出这个字符。使得将字符串删完以后，输出的字符串字典序尽量小。 ($n \leq 30000$)	贪心，每次比较从头部往右和从尾部往左形成的字符串，并取字典序小的那一个方向。 比较字典序大小可以用后缀数组。由于这里要比较反串，所以还需要把字符串翻转一遍接在原串后面再求后缀数组。	
时空复杂度	空间 $O(n)$ ，时间 $O(n \log n)$ （或 $O(n)$ ）。	

题 80

试题编号	USACO Mar 08	
试题名称	Land Acquisition	
题目大意	算法讨论	
需要购买 n 块土地，每块土地长 $a[i]$ ，宽 $b[i]$ 。一次可以购买若干块土地，所付的钱为 $\max a \cdot \max b$ 。 问全部购买所需的最小价钱。 ($n \leq 50000$)	剔除所有能被完全包含的土地，剩下的土地组成了 a 值递增， b 值递减的序列。容易看出每次购买其中连续的一段才可能最优。 可以发现这是一个经典的斜率优化 DP。用一个单调队列维护当前决策组成的凸壳即可。	
时空复杂度	空间 $O(n)$ ，时间 $O(n \log n)$ 。	

题 81

试题编号	USACO Mar 09	
试题名称	Cleaning Up	
题目大意	算法讨论	
将一个长为 n 的数列分成若干段，每段的代价是这段内不同数字个数的平方。求最小总代价。 ($n \leq 40000$)	一个 $O(n^2)$ 的 DP 是显然的。为了优化，注意到最优总代价上界是 n （每个数分为一段），所以每段内至多有 \sqrt{n} 种不同的数字，所以每次的决策数量不超过 \sqrt{n} ，只要用一个链表/队列维护最近的 \sqrt{n} 个不同的数字及位置，即可 DP。	
时空复杂度	空间 $O(n)$ ，时间 $O(n\sqrt{n})$ 。	

题 82

试题编号	USACO Open 08
试题名称	Cow Neighborhoods
题目大意	算法讨论
平面上有 n 个点，曼哈顿距离小于等于 C 的两点之间有连边。求最后的连通块数目、最大的那个连通块的大小。 $(n \leq 100000)$	将坐标系旋转 45 度后，每个点的影响范围是一个边平行坐标轴的正方形，便于处理。 然后按 y 轴排序插入，同时维护一个关于 x 坐标的 multi-set/平衡树，每次插入后先将 y 坐标差超出范围的点删除，然后将插入的点与其前驱后继判断是否要连边。 维护连通块可用并查集。
时空复杂度	空间 $O(n)$ ，时间 $O(n \log n)$ 。

题 83

试题编号	USACO Jan 07
试题名称	Cow School
题目大意	算法讨论
n 次考试，每次得分 t_i ，满分 p_i 。老师的计分方法是去掉 t_i/p_i 最低的 d 场考试，剩下的 $\sum t_i / \sum p_i$ 作为成绩。但是你如果用不同的方法去掉 d 场考试，成绩可能更高。求出所有这样可能的 d 。 $(n \leq 50000)$	假设已经排列使得 t_i/p_i 递增。老师算出成绩为 $X_d = \sum_{i=d+1}^n t_i / \sum_{i=d+1}^n p_i$ ，即 $\sum_{i=d+1}^n (t_i - X_d p_i) = 0$ 。如果可以修改去掉的考试集合而使得成绩更高，需要有 $\max_{i=1}^d (t_i - X_d p_i) > \min_{i=d+1}^n (t_i - X_d p_i)$ 。我们只需要分别求出这个式子中最大值和最小值，比较即可。 对所有 d 求出最大值的过程可以采用分治实现，每次求出左半边的凸壳，用来更新右半边，然后递归两边。最小值是对称的，方法一样。
时空复杂度	空间 $O(n)$ ，时间 $O(n \log^2 n)$ （归并替代排序可 $O(n \log n)$ ）。

题 84

试题编号	USACO Dec 12
试题名称	First!
题目大意	算法讨论
n 个字母串，我们可以改变字母表的排列顺序，使得某个字符串的字典序是 n 个中最小的。哪几个字符串满足这一性质。 $(n \leq 30000, \text{总串长 } S \leq 300000)$	对所有串建 Trie 树，并 DFS。要使某个字符串字典序最小，在遍历 Trie 树的过程中可以得知某些字母间的相对顺序关系，这组成一个有向图，只要这个图无环，则这个串可以是字典序最小的。在遍历 Trie 树的同时维护这个图，在遇到叶子结点时对当前图判断是否有环。
时空复杂度	空间 $O(26n)$ ，时间 $O(26^2 n + 26S)$ 。

题 85

试题编号	USACO Open 13
试题名称	Photo
题目大意	算法讨论
从 $1 \sim n$ 中选出某几个数。给出 m 个区间 $[l_i, r_i]$ ，要求使得每个区间内有且仅有一个数被选中。问选出的数最多能有几个。 $(n \leq 2 \cdot 10^5, m \leq 10^5)$	记 $x[i]$ 为所有包含 i 的区间中，最小的左端点。记 $y[i]$ 为所有右端点小于 i 的区间中，最大的左端点。 考虑动态规划。用 $f[i]$ 表示取了第 i 个数的情况下，前 i 个数中最多取几个。 $f[i] = \max_{y[j] \leq j < x[i]} f[j] + 1$ 。由于 $x[], y[]$ 有单调性，所以能用单调队列优化。
时空复杂度	空间 $O(n + m)$ ，时间 $O(n + m)$ 。

题 86

试题编号	USACO Open 09	
试题名称	Tower of Hay	
题目大意	算法讨论	
把长为 n 的数组 $a[]$ 分为 h 段, 并从下往上摆成 h 层, 某一层的数字之和不小于其上一层的数字之和。求 h 的最大值。 $(n \leq 10^5)$	可以证明第一层最小的方案中存在能使段数最大的方案。用 $f[i]$ 表示将 $a[i..n]$ 摆起来, 底层的最小长度。 $f[i] = \min_{i < j \leq n+1, \text{sum}[i, j-1] \geq f[j]} \{\text{sum}[i, j-1]\}$ 。这是一个可以用单调队列优化的 DP。	
时空复杂度	空间 $O(n)$, 时间 $O(n)$ 。	

题 87

试题编号	USACO Jan 09	
试题名称	Travel	
题目大意	算法讨论	
一个无向图, 从结点 1 到其它每个结点的最短路径是唯一的。对于除 1 外每个结点, 求出不经过原最短路径最后一条边的情况下, 最短路径是多少。 $(n \leq 10^5, m \leq 2 \cdot 10^5)$	由题目限制可知道最短路径图是没有环的 (即最短路径树)。对于所有不在最短路径树上的边 (u, v) , 令 $l = LCA(u, v)$, 则 l 到 v 路径上所有点 x (含 v , 不含 l) 可通过 $1 \rightarrow u \rightarrow v \rightarrow x$ 走到而不经 x 的父亲, 长度为 $d[u] + w(u, v) + d[v] - d[x]$, 那么可以用 $d[u] + w(u, v) + d[v]$ 去更新这些 x 。更新可以用树链剖分 + 线段树; 也可以从小到大排序后更新, 并用并查集维护以跳过已经更新过的点。	
时空复杂度	空间 $O(n \log n)$ 或 $O(n)$, 时间 $O(n \log n)$ 。	

题 88

试题编号	USACO Open 10	
试题名称	Triangle Counting	
题目大意	算法讨论	
坐标平面上有 n 个点, 极角各不相同且不位于原点。取出三个点组成三角形, 使得三角形包围住原点的有几种。 $(n \leq 100000)$	反过来求出不包围原点的三角形有几个, 再用总数减去即可。不包围原点的三角形一定能被放进一个半平面内, 且原点在半平面的边界线上。将所有点按极角排序后, 用过原点的直线扫一圈, 同时维护当前半平面内的点的数量用以累加答案。	
时空复杂度	空间 $O(n)$, 时间 $O(n \log n)$ 。	

题 89

试题编号	USACO Dec 10	
试题名称	Threatening Letter	
题目大意	算法讨论	
给定字母串 s, t 。将 t 分成尽可能少的段, 使每段都是 s 的子串。 $(s , t \leq 50000)$	用 $f[i]$ 表示 $t[1..i]$ 的答案, 容易发现 i 需要尽可能往前匹配, 越长越好; 设匹配长度为 $l[i]$, 则 $f[i] = f[i - l[i]] + 1$ 。对 s 建出后缀自动机, 用 t 串在自动机上跑一遍, 即可求出所有 $l[i]$ 。	
时空复杂度	空间 $O(26n + m)$, 时间 $O(26n + m)$ 。	

题 90

试题编号	USACO Open 13	
试题名称	Figure Eight	
题目大意	算法讨论	
一个 $n \times n$ 网格，有些格子是坏格。框出一个 8 字形，8 字形由两个矩形组成，上面的矩形的底边是下面的矩形的顶边的子集。矩形边框不能是坏格。求上矩形面积 \times 下矩形面积的最大值。 $(n \leq 300)$	用 $up[i][l][r], down[i][l][r]$ 分别表示以第 i 行 $[l, r]$ 作为底（顶）边，往上（下）伸展所得的最大矩形的面积，这可以简单递推得到。再处理出 $down$ 数组的二维前（后）缀 \max ，然后即可方便的更新答案。	
时空复杂度	空间 $O(n^3)$ ，时间 $O(n^3)$ 。	

题 91

试题编号	USACO Mar 13	
试题名称	Hill Walk	
题目大意	算法讨论	
平面上有 n 条斜率为正的斜线段，出发点为第一条线段的左端点。不断地从左下往右上走，到达线段的右端点后会下坠，如果坠落到另一条线段上则继续，否则结束。求最后走过了几条线段。 $(n \leq 10^5)$	需要处理每一条线段的后继。将线段的端点排序后，从右往左处理，同时维护一棵当前线段的平衡树/set，遇到右端点时在 set 中查询后继，并将此线段插入 set；遇到左端点时将该线段从 set 中移除。	
时空复杂度	空间 $O(n)$ ，时间 $O(n \log n)$ 。	

题 92

试题编号	USACO Nov 12	
试题名称	Balanced Trees	
题目大意	算法讨论	
一棵树的每个结点是一个左括号或右括号。找出一条链，使得它是一个合法的括号序列，且其括号嵌套最深度最大。 $(n \leq 40000)$	考虑括号序列的前缀，两个合法的、且前缀和（左右括号分别视作 $1, -1$ ）相等的前缀 a, b ，将 b 翻转后接在 a 后面，可以得到一个完整的括号序列。于是可以采用点分治，对于每个子树，求出子树内结点到根路径组成的每种前缀和对应的最大可能深度。然后可以合并信息，更新答案。	
时空复杂度	空间 $O(n)$ ，时间 $O(n \log n)$ 。	

题 93

试题编号	USACO Dec 08	
试题名称	Largest Fence	
题目大意	算法讨论	
平面上有 n 个点，任三点不共线。找出以它们为顶点的凸包，使得凸包上的顶点数量最多。 $(n \leq 250)$	首先枚举凸包的最低点 t ，然后 dp: $f[i][j]$ 表示最大的 k ，使得 $x_1 = i \rightarrow x_2 \rightarrow x_3 \rightarrow \dots \rightarrow x_k = t$ 能够成为凸包上按逆时针序的某一段，且 $i \rightarrow x_2$ 在 $i \rightarrow j$ 的逆时针方向。dp 前需要预处理出 $next[i][j]$ ，表示以 i 为原点的极角序中， j 逆时针方向的下一个点。则 $f[i][j]$ 可以由 $f[i][next[i][j]]$ （跳过 j 不选）或 $1 + f[j][next[i][j]]$ （选 j 点）转移而来。	
时空复杂度	空间 $O(n^2)$ ，时间 $O(n^3)$ 。	

题 94

试题编号	USACO Mar 12	
试题名称	Cows in a Skyscraper	
题目大意	算法讨论	
有 n 个物品，第 i 个的重量为 a_i 。包容量为 w ，求最少需要几个包可以全部装下。 $(n \leq 18, w, a_i \leq 10^8)$	$f[S]$ 表示装入 S 集合的物品需要几个包， $g[S]$ 表示此时最后一个包的最大剩余容量。枚举当前 S 中的每一件物品即可进行转移。	
时空复杂度	空间 $O(2^n)$ ，时间 $O(n \cdot 2^n)$ 。	

题 95

试题编号	USACO Nov 08	
试题名称	Toys	
题目大意	算法讨论	
共有 n 天，第 i 天需要用掉 a_i 个玩具。玩具可以花 c 的单价买，也可以花 c_1 的单价快洗，花费 t_1 天；也可以花 c_2 的单价慢洗，花费 t_2 天。求最小总代价。 $(n \leq 10^5)$	假如已知买了 x 个玩具，可以用贪心求出最后的最小花费（或判定无解），只要每次尽量取买来的玩具，若取完则取之前最近的可以慢洗的玩具，若仍不行再尝试取快洗的。 可以证明求得的花费是关于 x 的单峰函数，于是三分即可求出最小值。	
时空复杂度	空间 $O(n)$ ，时间 $O(n \log(\sum a_i))$ 。	

题 96

试题编号	Codeforces 293D	
试题名称	Ksusha and Square	
题目大意	算法讨论	
平面上有一个 n 个顶点的凸包，从凸包内（包括边界）随机选出两个不同的整点，以它们为对角线的正方形面积的期望是多少。 $(n \leq 10^5, x_i, y_i \leq 10^6)$	假如选出的两点为 $(x_1, y_1), (x_2, y_2)$ ，则面积为 $2(x_1 - x_2)^2 + 2(y_1 - y_2)^2$ ，可以横纵坐标的贡献分别计算后相加。 由于坐标的范围并不大，可以统计出对于每个 x ，有多少个整点在凸包内，只需要在扫描的同时求出和当前 x 相交的凸包的边即可。 然后需要统计的答案为 $\sum_{a < b} cnt[a]cnt[b](b - a)^2$ 。在扫描的同时维护 $\sum cnt[a], \sum cnt[a] \cdot a, \sum cnt[a] \cdot a^2$ 即可统计。	
时空复杂度	空间 $O(n + X)$ ，时间 $O(n + X)$ 。	

题 97

试题编号	Codeforces 331D3	
试题名称	Escaping on Beaveractor	
题目大意	算法讨论	
一个范围 $b \times b$ 的坐标系中，有 n 个平行于坐标轴的箭头。你在其中以 1 的速度运动，遇到箭头则转向箭头方向，超出边界则停止。共有 q 个询问，每次给定起始点和起始方向（平行于坐标轴），问 t 秒后你的位置。 $(n, b, q \leq 10^5)$	首先求出每个箭头的后继，对 4 种方向分开处理，按对应方向排序后扫一遍，可以转化为区间覆盖、单点求值，用线段树可以解决。 然后用倍增，记录每个箭头的末端出发走过 2^i 个箭头后所处的位置和所用的时间。然后即可回答每个询问。	
时空复杂度	空间 $O(n \log n)$ ，时间 $O(n \log n)$ 。	

题 98

试题编号	Codeforces 325C
试题名称	Monsters and Diamonds
题目大意	算法讨论
<p>n 种怪物, 和 m 种分裂方法。分裂可以把某一种怪物分裂成几颗钻石和 (或) 某几种怪物。</p> <p>问给定一只某种怪物作为初始, 把所有怪物全部分裂成钻石, 是否可行, 可行时输出可以得到钻石数量的最小值、最大值 (或无穷)。</p> <p>$(n, m \leq 10^5, \sum \text{每种方法的后继数量} \leq 10^5)$</p>	<p>若要有解, 需要存在一些方法的后继只含钻石而不含怪物, 从这些出发去更新其它每种方法可得到的钻石最小值, 更新顺序类似 Dijkstra。于是我们可以知道每种怪物能生成的最小钻石数量, 也知道了哪些怪物是不可行的。</p> <p>然后对可行的怪物进行 DFS, DFS 的过程中如果遇到了环, 则可以生成无穷个钻石, 否则我们可以求出这个怪物的有限的最大值。</p>
时空复杂度	空间 $O(n)$, 时间 $O(n \log n)$ 。

题 99

试题编号	GCJ 2009 Final D
试题名称	Wi-fi Towers
题目大意	算法讨论
<p>平面上有 n 个点, 每个点有一个范围和一个得分 (可正可负)。如果选中某个点, 则也需要选中它的范围内的所有点。求选中的点的最大分数总和。 ($n \leq 500$)</p>	<p>这是一个经典的最大权闭合子图问题, 经典做法是转化为最小割。</p>
时空复杂度	(最大流) 点数 $O(n)$, 边数 $O(n^2)$ 。

题 100

试题编号	Codeforces 301E
试题名称	Yaroslav and Arrangements
题目大意	算法讨论
<p>多少种 r 元数字集合 (可重集合), 可以将其用至少 1 种不超过 k 种方法排成一个序列 $\{a_i\}$, 使得 $a_1 - a_2 = \dots = a_r - a_1 = 1$, $a_1 = \min a_i$。元素在 $1 \leq a_i \leq m$ 范围内, 长度在 $1 \leq r \leq n$ 范围内。 ($n, m, k \leq 100$)</p>	<p>考虑这样一个序列的产生过程, 按数字大小从小到大插入, 可以将若干个数字 $x+1$ 插在原先两个数字 x 的空隙中。若原先有 c 个空隙, 插入了 p 个数字 $x+1$, 则 $p \geq c$, 插入后形成新的空隙 $p - c$ 个, 这一步骤的方案数为 $\binom{p-1}{c-1}$。于是可以据此进行 DP。</p>
时空复杂度	空间 $O(n^4)$ (滚动数组 $O(n^3)$), 时间在 $O(n^4)$ 和 $O(n^5)$ 之间?

题 101

试题编号	Codeforces 248E
试题名称	Piglet's Birthday
题目大意	算法讨论
<p>n 个架子上, 每个架子初始有 a_i 个装满了的蜜罐。每次从第 u 个架子上随机取 k 个罐, 喝完放到 v 架子上。每次操作后输出只有空罐的架子个数的期望。 ($n \leq 10^5, a_i \leq 100, k \leq 5$)</p>	<p>可以知道每个架子上装满了的罐子的数量是不会增加的。对每个架子 i 维护 $f[i][j]$, 表示这个架子上留有 j 个装满了的罐子的概率。每次操作后更新这个数组即可 (根据组合数的定义)。</p>
时空复杂度	空间 $O(na)$, 时间 $O(kna)$