1001 求和

对于某一非负整数 d 和任一非负整数 x ，由 f_x 无法得到 f_{x+d} ，但可由 $f_{[x-k+1,x]}$ 以暴力递推或更快的方式得到 $f_{[x+d-k+1,x+d]}$ ；记这种运算为 F_d ，并将 f 序列视为向量（可加），$F_d(f_{[x-k+1,x]}) = f_{[x+d-k+1,x+d]}$；

对于集合 S ，记其前 i 个数形成的集合为 S_i ；对于任意 $T \subseteq S_{i-1}$ ：

- 其 $T, y = \sum_{x\in T} x, f_y, f_{[y-k+1,y]}$ 分别与 S_i 中的
  - 一组 $T, y = \sum_{x\in T} x, f_y, f_{[y-k+1,y]}$
  - 和一组 $\{T, b_i\}, \sum_{x\in\{T,b_i\}} x = y + b_i, f_{y+b_i}, F_{b_i}(f_{[y-k+1,y]})$

  相对应；
- 记 $Vec_i = \sum_{T\subseteq S_i} f_{[y-k+1,y]}$ ，可得 $Vec_i = F_{b_i}(Vec_{i-1}) + Vec_{i-1}$ 。

由定义得 Vec_0 的最后一项 a_0 为 1 ，其他项为 0 ；Vec_n 的最后一项即要输出的答案；我们只需要关心如何进行 F_d ：

- 由于递推 Vec 需要调用 F 运算 n 次，矩阵乘法 $O(nk^3)$ 的做法是行不通的；
- 可以使用特征多项式：
  - 对任一非负整数 x ，若限制其 f_x 往后的递推贡献限制在范围 $[x+d-k+1, x+d]$ 内，对各项的贡献系数可由一个长 k 的特征多项式 g_d 反映；
  - 对于 $d = 0$ ：
  $$(g_d)_i = \begin{cases} 0, i \neq k-1 \\ 1, i = k-1 \end{cases}$$
  - 对于 $d > 0$ ：
  $$(g_d)_i = \begin{cases} (g_d)_{i+1} + (g_d)_0 \cdot a_{i+1}, i \neq k-1 \\ (g_d)_0 \cdot a_k, i = k-1 \end{cases}$$
  - 用 g_{d-1} 递推 g_d 用时 $\mathcal{O}(k)$；
  - 用 g_a 和 g_b 计算 g_{a+b} ，即对 g_a 的各项套用 g_b 向后贡献，再对齐到范围正确范围内，用时 $\mathcal{O}(k^2)$ ；对向量运算同理，递推 Vec 部分的复杂度为 $\mathcal{O}(nk^2)$ ；
  - 进一步地，用 g_d 倍增可得 g_{2d} ，以二进制构造可以 $\mathcal{O}(k^2 \log V)$ 得到每个 g_{b_i} ，但仍不足以通过本题；
  - 可以用更进一步的科技加速（不作展开）；或取一参数 B，预处理 $\{g_d \mid 0 \le d < B \vee d \,\%B = 0\}$，以 $\mathcal{O}(Bk + \frac{Vk^2}{B})$ 得到每个 g_{b_i} ；或进行其他方式的分块。

总复杂度 $\mathcal{O}(nk^2 + Bk + \frac{Vk^2}{B})$ （仅供参考）。

1002 小抹爱锻炼

当存在 $1 \le i < j \le n$ 使得 $b_i > c_j$ 时无解；

第 i 天训练量的下界为 $D_i = \max\limits_{j=1}^{i} b_j$ ，上界为 $U_i = \min\limits_{j=i}^{n} c_j$ ；

若训练量总和为 M ，需要每一天的 $D_i \le U_i$ ，且 $\sum\limits_{i=1}^{n} D_i \le M \le \sum\limits_{i=1}^{n} U_i$ 。

总复杂度 $\mathcal{O}(n)$ 。

1003 光线折射

暴力递推经过每条网格线的光强几无可能，试着研究每条路径。

对于沿着一条完整路径（从起点到终点）的光线，其经过的玻璃数固定为 $R=n+m+1$ ，设其被反射的次数为 k ，则被折射的次数为 $R-k$ ，这条路径的光强可以确定为 $bas_k=\frac{c^k d^{R-k}}{(c+d)^R}$ ；如果我们能再计算出对于每个 k 完整路径的个数 cnt_k ，即可得到答案 $Ans=\sum\limits_{k=0}^{R} bas_k \cdot cnt_k$ 。

k 也可视作路径拐弯的次数，在本题中显然 k 为奇数时 $cnt_k=0$ ，我们另记 $z=\frac{k}{2}$。

做法1:

路径拐弯 $2z$ 次相当于:

- 把跨过 $n+1$ 个整点的水平路径拆成 $z+1$ 个非空段，即在 $n+1$ 个分断点 $(x=0\sim n)$ 中选取 z 个，方案数为 C_{n+1}^{z} 个;
- 互不干扰地，把长为 m 的垂直路径拆成 z 个非空段，即在 $m-1$ 个分断点 $(y=1\sim m-1)$ 中选取 $z-1$ 个，方案数为 C_{m-1}^{z-1} 个。
  - 注意考虑 $z=0$ 的特殊情况，$m=0$ 时应定义 C_{m-1}^{z-1} 为 1 ，否则为 0 ；我们记这种加了特判的 C_a^b 为 D_a^b 。
  - 当然不合法的 C_a^b ，比如 $a<b$ ，也全部视作 0 。

因此有 $cnt_{2z}=C_{n+1}^{z}D_{m-1}^{z-1}$ 。

做法2:

递推从起点起，接下来将要经过的点为点 (i,j) ，且拐弯次数为 k（这时暂不忽视 k 为奇数的情况）的路径数 $A_k(i,j)$ 。考虑路径与最后一个拐弯点的组合，有:

$$A_k(i,j)=\begin{cases}0, i<0\ or\ j<0\ or\ k<0\\ [j=0], k=0\\ \sum\limits_{0\le h<j} A_{k-1}(i,h), k\text{为正奇数}\\ \sum\limits_{0\le w<i} A_{k-1}(w,j), k\text{为正偶数}\end{cases}$$

发现 i 和 j 之间是互不干扰的。拆解 $A_k(i,j)$ 为 $W_{\lfloor\frac{k}{2}\rfloor}(i)\cdot H_{\lceil\frac{k}{2}\rceil}(j)$ ，有:

$$W_c(i)=\begin{cases}0, i<0\ or\ c<0\\ 1, c=0\\ \sum\limits_{0\le w<i} W_{c-1}(w), c>0\end{cases}, H_c(j)=\begin{cases}0, j<0\ or\ c<0\\ [j=0], c=0\\ \sum\limits_{0\le h<j} H_{c-1}(h), c>0\end{cases}$$

更进一步地，设 $W'_c(i)=W_c(i+c)$，$H'_c(i)=H_c(i+c)$ ，有

$$W'_c(i)=\begin{cases}0, i<0\ or\ c<0\\ 1, c=0\\ \sum\limits_{0\le w\le i} W'_{c-1}(w), c>0\end{cases}, H'_c(j)=\begin{cases}0, j<0\ or\ c<0\\ [j=0], c=0\\ \sum\limits_{0\le h\le j} H'_{c-1}(h), c>0\end{cases}$$

其实在数值上，恒有$W'_c(v) = H'_{c+1}(v)$（仅有$c = -1$且$v = 0$时除外）；而 $W'_c(v)$ 实际上完全等价于 C_{c+v}^{c}，$H'_c(v)$ 的特判也与前文的 D_a^b 不谋而合。

于是有 $A_{2z}(i,j) = W_z(i) \cdot H_z(j) = W'_z(i-z) \cdot W''_{z-1}(j-z) = C_i^z \cdot D_{j-1}^{z-1}$；经过点 $(n+0.5, m)$ 的路径将经过点 $(n+1, m)$，因此有 $cnt_{2z} = A_{2z}(n+1, m) = C_{n+1}^{z} D_{m-1}^{z-1}$，与做法1中的式子完全一致（当然，特判也是一致的）。

最终整理得到：

$$Ans = \sum_{z=0}^{\lfloor \frac{R}{2} \rfloor} bas_{2z} cnt_{2z} = \sum_{z=0}^{\lfloor \frac{R}{2} \rfloor} \frac{c^{2z} d^{R-2z}}{(c+d)^R} \cdot C_{n+1}^{z} D_{m-1}^{z-1}$$

总复杂度 $\mathcal{O}(\sum n + \sum m)$。

1004 三带一

二分答案，判断某 Ans 是否有解：

设每个数有 b_i 个 "3" 被使用，则余 $c_i = a_i - 3b_i$ 个 "1"，使用个数待定；

由 c_i 不能为负有限制：

1. $\forall\ i \to b_i \le \lfloor \frac{a_i}{3} \rfloor$

另设 $\sum a_i = A, \sum b_i = B = Ans, \sum c_i = C = A - 3Ans$；

由于每个 "3" 都要匹配一个 "1"，有限制：

2. $B \le C \to Ans \le \lfloor \frac{A}{4} \rfloor$

另外，给定所有 b_i 和 c_i，存在方案使每个 "3" 匹配上不同数的充要条件是 $\max(b_i + c_i) \le C$：

- 当存在某 i 使得 $b_i > C - c_i$ 时，显然该 i 的 b_i 个 "3" 无法被匹配完；
- 否则当存在某 i 使得 $b_i + c_i = C$ 时，匹配的策略显然；
- 否则由于任意一次操作仅使 C 减1，且各 i 的 $b_i + c_i$ 不升，可随意匹配直至出现 $\max(b_i + c_i) = C$。

即 $\forall\ i \to b_i + c_i \le C$；又 $b_i + c_i = a_i - 2b_i$，得限制：

3. $\forall\ i \to b_i \ge \lceil \frac{a_i - A + 3Ans}{2} \rceil$

整理所有限制，得 $[Ans \le \frac{A}{4}] \wedge [\forall i, \lceil \frac{a_i - A + 3Ans}{2} \rceil \le b_i \le \lfloor \frac{a_i}{3} \rfloor]$

通过累加各 b_i 的上下界判该Ans的合法性（注意单个 b_i 的上下界之间也不能冲突）。

1005 平衡阵盘

先对边的分布进行一些分析：

- 可仅由黑/黄边相连的点之间不可能由单条红蓝边连通，说明黑/黄边形成若干双向完全图；
- 若钦定哪些边为红蓝边，哪些边不是：将黑/黄边完全图缩为一点，再去除所有蓝边，图中剩余的红边不可能形成回路，且此时所有点对都由一条红边相连，则各点存在由红边决定的拓扑序；反之确定各点（黑/黄边完全图）的拓扑序能反过来决定红蓝边的方向，两者情况一一对应，个数相等。

考虑以“按拓扑序排序后黑/黄边完全图大小的序列”分类合法阵盘的所有可能性；对于一序列 a （其长度为 m ，阵盘总点数为 $n=\sum\limits_{i=1}^{m} a_i$ ）：

- 符合该序列，忽略黑/黄边完全图结构的阵盘数为 $f'(a)=\frac{n!}{\prod\limits_{i=1}^{m} a_i!}$ ；
- 不忽略黑/黄边完全图结构的阵盘数为 $f(a)=f'(a)\cdot\prod\limits_{i=1}^{m} G(a_i)!$ ，其中 $G(x)$ 为大小为 x 的有标号黑/黄边完全图种数；
- 可另记 $H(x)=\frac{G(x)}{x!}$ ，得 $f(a)=n!\cdot\prod\limits_{i=1}^{m} H(a_i)!$

则 n 点有标号合法阵盘的个数 $Ans_n=\sum\limits_{a:\sum a=n} n!\cdot\prod\limits_{i=1}^{m} H(a_i)!$;

另考虑拓扑序最大黑/黄边完全图的添加（递推）与删去，记
$Ans'_n=\frac{Ans_n}{n!}=\sum\limits_{a:\sum a=n}\prod\limits_{i=1}^{m} H(a_i)!=\sum\limits_{i=1}^{n} H(i)\cdot Ans'_{n-i}$。

至此，最重要的是如何求得各 $G(x)$ ：

事实上，由于 n 条边在 n 点中必定成环，边双色而不出现回路的有标号双向完全图不可能很大；有：

$$G(x)=\begin{cases}1, x=1\\2, x=2\\6, x=3\\12, x=4\\0, x\geq 5\end{cases}$$

（可暴搜，也可直接手玩得到）

整理以上内容即可线性得出 $Ans_{1\sim 10^7}$ ，总复杂度 $\mathcal{O}(n)$ 。

1006 巨龙守卫

尝试从高位到低位确定每个士兵的力量值 a_i ，容易确定：

- 设当前操作位为 h ,将某数 x 的第 h 位及以上的部分记作 x_h ；
- 每个 $(a_i)_h$ 是否在区间 (l_h, r_h) 内或边界上（较高的若干位是否与 l_h 或 r_h 相等/在两者之间），或：在区间 (l_h, r_h) 内不可能再跑出边界 / 左边界 l / 右边界 r 的三部分位置各有多少个 a_i ；
- 现 $S_2=\oplus_{i=1}^{n} a_i$ 是否等于/小于 V_2 （小于则接下来不再受 V_2 的限制）；
- V_1 减去 现 $S_1=\sum_{i=1}^{n} a_i$ 的值；若大于则不合法，否则记刚操作完的位为第 k 位（从最低位第 0 位起计），记录 $\min(\lfloor\frac{V_1-S_1}{2^k}\rfloor, n)$ （前者达到 n 则接下来不再受 V_1 的限制）。

定义dp状态 $dp[h][cl][cr][slf][xt]$:

- h ：当前操作位，范围为 $0\sim 30$ ；初始状态可从 $h=30$ 开始，可用滚动数组去掉该维；
- cl ：当前有多少个 a_i 满足 $(a_i)_h=l_h$ ，范围为 $0\sim n$ ；
- cr ：当前有多少个 a_i 满足 $(a_i)_h=r_h$ ，范围为 $0\sim n$ ；
    - $l_h=r_h$ 的前缀部分处理方法因人而异；
- slf : 上文的 $\frac{V_1-S_1}{2^k}$ 或 $\min(\lfloor\frac{V_1-S_1}{2^k}\rfloor, n)$ ，允许在当前位再加上的 1 的个数，范围为 $0\sim n$；
- xt ：当前是否有 $(S_2)_h=(V_2)_h$，是则为 1 ，不是则为 0 ，范围为 $0\sim 1$。

由此将每个操作位（每层）的状态数压缩至 $\mathcal{O}(n^3)$ ；

每个状态往低位转移时，需确定在区间 (l_h, r_h) 内不可能再跑出边界 / 左边界 l / 右边界 r 的三部分位置的 a_i 各有多少个在下一位接 0（另一部分接 1 ），单个状态转移复杂度为 $\mathcal{O}(n^3)$ ，整层状态转移复杂度为 $\mathcal{O}(n^6)$ ——但实际上枚举得当可以除以 $5!$ （考虑将士兵分为 6 段的 5 个有序分界点），且有相当多的状态无法到达，可以剪枝。

单测的复杂度为极小常数 $\mathcal{O}(n^6 logV)$ ，总复杂度为极小常数 $\mathcal{O}(\sum n \cdot n^5 logV)$ ；复杂度不是本题的重点。

1007 性质不同的数字

关注（初始状态和）每个“有元素出现或消失的时间点后”的状态：

- 可以直接哈希，比如给每个元素赋随机哈希值，维护场上所有元素哈希值的异或和；注意哈希值的范围不宜低于 10^{12} ；
- 也可以用一个类似栈的结构在排序后以 $\mathcal{O}(n)$ 准确求得答案：
  - 栈内每个元素有一独立的位置（“高度”），可能不连续；时刻（在 `vector` 中用指针）追踪栈的最大高度 h 及其对应元素 id ，和栈内的元素数 cnt ；
  - 每个元素“入栈”时赋其一个“高度”，为当时栈内所有元素“高度”的最大值加1，即 $h+1$ ；
  - 对比不同时刻，栈的 $\{h, id, cnt\}$ 三元组不同时对应的状态肯定不同，而三元组相同时对应的状态肯定相同；$\{h, id, cnt\}$ 三元组与状态一一对应，只需求三元组种数；
  - 一个 id 只能占据某“高度”一段连续时间，且作为 h 时 cnt 单减；每当发现某高度作为 h 且 $\{id, cnt\}$ 二元组较上次不同，即认为出现新的三元组。

两种做法的总复杂度均为 $\mathcal{O}(n \log n)$ ，来自排序。

1008 01环

由显然的等效策略，大可将所有“翻转操作”放在最后；

钦定每个位置的最终颜色（只有2种组合，从某位起为0/1），要求：

进行任意次“交换操作”（a 次），并记交换操作结束后颜色不符预期的位置数为 b ，最小化 $a+b$ 。

由于两个有重叠，且两时间点间无其他“交换操作”干扰的“交换操作”可由一次“交换操作”和一次“翻转操作”或两次“翻转操作”替换，可限制每个位置最多被“交换操作”影响一次，将不符该条件的策略转化为符合该条件的策略不劣；

任取环上的两个相邻点，分别断环成链，跑线性dp即可；任意“交换操作”方案在至少一条链中没有任何交换操作被打断。

总复杂度 $\mathcal{O}(n)$ ，也有很多其他做法。

1009 线段染色

从左到右（从 1 到 n ）进行染色，记已尝试染色整点 $1 \sim i$ ，且 $1 \sim i$ 内的线段已全被染色，且当前最后一个被染色点为 j 的概率为 $dp_{i,j}$ 。

尝试从 $i-1$ 转移至 i：

1. 当点 i 被染色时，所有状态有同样的 p_i 的概率被改变（最后一个染色点被后移至 i ），则使全局的所有值乘以 $1-p_i$ ，再将损失的值加到位置 i：

$$dp_{i,j}=\begin{cases}dp_{i-1,j}\cdot(1-p_i), j<i\\ \sum_{k=0}^{j-1} dp_{i-1,k}, j=i\\ 0, j>i\end{cases}$$

1. 这之后关注右端点为 i 的线段，取其中最大的左端点为 mxl_i （没有线段则记其为 0 ），此时最后一个被染色点为 $j<mxl_i$ 则确定不符合“染色所有线段”的条件，使所有 $dp_{i,j<mxl_i}$ 变为 0 。

最后要求的 $Ans=\sum_{j=0}^{n} dp_{n,j}$ 。

易线性求得每个 mxl_i 和 $mxl'_i=\min_{j=0}^{i} mxl_j$；但用区乘线段树以 $\mathcal{O}(n\log n)$ 转移 dp 可能过慢。

优化：

维护“全局的所有值之和”为 sum ；

1. 将上文的第1步改为：
   - 若 $p_i\neq 1(a_i\neq 10)$ ，全局的值不变，仅赋位置 i 的值为 $sum\cdot\frac{p_i}{1-p_i}$ （最后再使答案乘回 $1-p_i$）；
   - 否则赋位置 i 的值为 sum，并视 mxl_i 为 i ；
2. 维护“存在有效值”的区间 $[mxl'_i,i]$ ；区间的左右端点都只会向右移动，移动时容易维护区间内的 " sum" ；

最后要求的 $Ans=sum\cdot\prod_{i=1\sim n,p_i\neq 1}(1-p_i)$ 。

注意预处理 $0\sim 10$ 关于 10^9+7 的逆元；总复杂度 $\mathcal{O}(\sum n+\sum m)$。

1010 坚船利炮

以任意点为根（以1为例），记每个点（点 i ）的儿子数为 ch_i ，其本身与其前 j 个儿子所对应子树形成的连通块为 $B_{i,j}$ ；特别地， $B_{i,0}$ 为点 i 本身形成的连通块。

试递推每个被定义连通块 $B_{i,j}$ ，内含 k 条被打断的边时的状态：

- 打断边的方案数 $c_{i,j,k}$；
- 各方案中未确定大小（与子树根相连）的连通块大小之总和 $sz_{i,j,k}$；
- 各方案中未确定大小（与子树根相连）的连通块大小平方之总和 $a_{i,j,k}$；
- 各方案中已确定大小（与子树根不相连）的连通块大小平方之总和 $fx_{i,j,k}$。

当连接两连通块 U 和 D（U 含树根，D 的子树根紧邻 U 的树根）形成新连通块时，根据所用边是否被打断，4个参数的组合计算略有区别，具体见代码。

最后，k 条边被打断时的答案即 $\frac{a_{1,ch_1,k}+fx_{1,ch_1,k}}{c_{1,ch_1,k}}$ 。
由树上背包的结论，总复杂度 $\mathcal{O}(nk)$ 。

1011 难度调整

记 $c_i = i - a_i$ ，题意转化为：

- 对于每个 $1 \le i \le n$ ，允许进行0或1次操作使 $[i, n]$ 范围内的所有 c_j 减1；
- 记 $c_0 = 0$ ，$\forall\, 1 \le i \le n \to c_i - c_{i-1} \le 1$ ；
- 最小化 $Ans = \sum_{i=1}^{n} c_i$ 。

考虑进行操作的位置序列 p 及位置对应的初始 c 序列（记为 S 序列） S_p 和最终 c 序列（记为 T 序列） T_p （显然有 $S_{p_i} - T_{p_i} = i$ ），并进行一些贪心：

- 在任意操作方案中，某 p_i 使得 $T_{p_i} = h < 0$ 时更劣： p_i 往后极长的已操作位置连续段内，（记段内任意位置为 j ）都有 $T_j \le T_{p_i} < 0$ ；大可转而操作往后第一个未被操作的位置，或若无该位置则去掉该操作，使答案减小；
    - 如此最优操作方案中必有 $(S_{p_i} >\ )\ T_{p_i} \ge 0$；
- 在任意操作方案中，某 p_i 使得 $S_{p_i} = h > 1$ ，而 $h - 1$ 在 S_{p_j} 中不出现时更劣优：找到 p_i 往前最近的位置 j 使得 $S_j = h - 1$ ，将操作从 p_i 替换至位置 j 可使答案减小；
    - 如此最优操作方案中有 $S_{p_i} = h > 1$ 则必有 $S_{p_{j<i}} = h - 1$ ，且 S_p 形成的不重集合一定是从1起的连续正整数（或为空）；
    - 且这意味着 $S_{p_i} \le S_{p_{i-1}} + 1 \le \cdots \le i$，和 $T_{p_i} = S_{p_i} - i \le 0$，结合前文 $T_{p_i} \ge 0$ 有 $T_{p_i} = 0$!
- 确定 $T_{p_i} = 0$ 和 $S_{p_i} = i$ 后，容易分析：
    - 选择操作 p_i 需 $c_{p_i} = i$ ，且将使答案减小 $f_i = \sum\limits_{j=i}^{n} [c_j \ge c_i] - [c_j < c_i]$
    - 需要找出一个可空位置序列 p 使 $c_{p_i} = i$ 且最大化 $\sum\limits_{x \in p} f_x$

计算各 f_i 可使用树状数组等，也可利用 $\forall\, 1 \le i \le n \to c_i - c_{i-1} \le 1$ 的特性均摊 $\mathcal{O}(1)$ 调整线性求得；最后进行简单的线性dp即可；复杂度 $\mathcal{O}(n \log n)$ 或 $\mathcal{O}(n)$ 。

1012 核心共振

以任意核心为坐标原点，并任选一个正参数 d ，画出 $\max(|x|, |y|) = d$ 所对应的曲线，再将坐标轴旋转45度，你会发现图形 $\max(|x|, |y|) = d$ 变成了图形 $|x| + |y| = \sqrt{2}\, d$ ，意味着你可以将 $\max$ 斜着拆成两维独立的绝对值；

或者有 $\max(|x|, |y|) = \frac{|x-y|+|x+y|}{2}$ ，即
$\max(|x_i - x_j|, |y_i - y_j|) = \frac{|(x_i-y_i)-(x_j-y_j)|+|(x_i+y_i)-(x_j+y_j)|}{2}$

我们以 $x + y$ 为例，计算 $Ans' = \sum\limits_{1 \le i \le j \le n} (a_i + a_j) \cdot |(x_i + y_i) - (x_j + y_j)|$: 将各点按 $w = x + y$ 升排，得 $Ans' = \sum\limits_{1 \le i \le j \le n} (a_i + a_j) \cdot (w_j - w_i)$ ；随后自由拆解，各前后缀的 $\sum 1, \sum a, \sum w, \sum a \cdot w$ 等易求。

镜像处理 $x - y$ 即可求出 Ans ；总复杂度 $\mathcal{O}(n \log n)$ ，主要来自排序。