PKU Campus 2019 **Summer 2019**

Problem Solution

PKU ACM Team

A Ball

我们先做这样的建模: 把每个篮子看成图上的一个点，把球看成连接两个篮子代表的点的边。我们就变成了要计算一个这样的问题：给一张图 $G = (V, E)$, 可以把每条边分配给一个相邻的结点，每个结点只能有一条边分配给它，问有多少种分配边的方案数?

显然，不同的连通分量之间互不影响，我们只需要计算每一个连通分量的答案，再把它们的答案按乘法原理分别乘起来即可。

考虑一个连通分量 $G' = (V', E')$, 首先我们不难发现当 $|E| > |V|$ 时，显然结点并不够装下所有的边，所以答案为 0。又因为 G' 是连通分量，只需要考虑 $|V| = |E|$ 与 $|V| - 1 = |E|$ 两种情况即可。

A.1 $|V| = |E| - 1$

不难发现这种情况下，连通分量的形态是一棵树。因为 $|V| = |E| - 1$，所以有且仅有一个点不拥有一条边，我们枚举是哪个点没有边，把这个点拎到根后，不难发现这种情况有且仅有一组解（即每条边属于它下面的那个点），所以这种情况的方案数为 $|V|$。

A.2 $|V| = |E|$

不难发现这种情况下，连通分量的形态是一棵环套树。因为 $|V| = |E|$，所以每个点都恰好拥有一条边。当环上的点拥有哪些边确定之后，剩下外面的树上的也唯一确定了。而环上的点只能拥有环上的边，只不过方向有区别 (顺时针和逆时针), 所以当环不是自环时，方案数为 2，否则为 1。

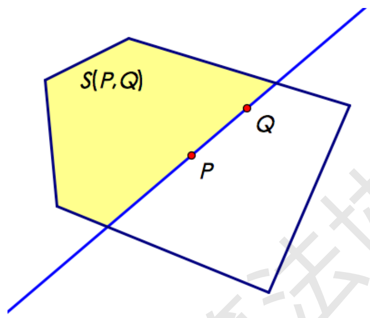
所以我们只需要写一个 DFS 就能解决本题，时间复杂度 $O(n)$。

B Seal

设 $f(P)$ 为点 $P=(p_x,p_y)$ 不在凸包 B 内的概率，那么答案就是：

$$S(A)-\iint f(p_x,p_y)\ \mathrm{d}p_x\mathrm{d}p_y$$

如果 P 不在凸包 B 内，那么 P 到 B 一定有两条切线 l_1,l_2，其中凸包 B 在 l_1 的逆时针方向，在 l_2 的顺时针方向。因为点到凸包的切线一定过凸包的顶点，设 l_1 经过的顶点是 Q，那么此时其他的所有点一定都在 Q 的逆时针方向（相对 P）。



如图所示，在 P,Q 的位置一定给定的情况下，剩下的 $m-1$ 个点必须分布在黄色区域内。设这个区域的面积是 $S(P,Q)$，那么答案就是（其中 P,Q 必须在凸多边形 A 内）：

$$\iiiint \frac{m}{S(A)}S(P,Q)^{m-1}\ \mathrm{d}p_x\mathrm{d}p_y\mathrm{d}q_x\mathrm{d}q_y$$

首先进行一个简单的还原，设直线 PQ 为 $u(\cos\theta,\sin\theta)+t(-\sin\theta,\cos\theta)$，其中 u 表示原点到这条直线的距离，θ 表示斜率，t 表示直线上一个点的位置。不妨设 P 的位置是 a，Q 的位置是 b，枚举直线的方向从 $-\pi$ 到 π，那么 a,b 需要满足 $b>a$。计算对应的雅克比行列式为：

$$\left|\begin{bmatrix}\cos\theta & \sin\theta & \cos\theta & \sin\theta\\ -u\sin\theta-a\cos\theta & u\cos\theta-a\sin\theta & -u\sin\theta-b\cos\theta & u\cos\theta-b\sin\theta\\ -\sin\theta & \cos\theta & 0 & 0\\ 0 & 0 & -\sin\theta & \cos\theta\end{bmatrix}\right|=b-a$$

因此得到（其中 $a,b-$ 必须保证 P,Q 在凸多边形 A 内）：

$$\iiiint \frac{m}{S(A)}S(P,Q)^{m-1}|b-a|\ \mathrm{d}u\mathrm{d}\theta\mathrm{d}a\mathrm{d}b$$

先对 a,b 进行积分，设 $L(P,Q)$ 为直线 PQ 在凸多边形 A 内的长度，那么就有：

$$\iint |b-a|\mathrm{d}a\mathrm{d}b=\frac{L(P,Q)^3}{6}$$

于是得到一个关于 u,θ 的二重积分：

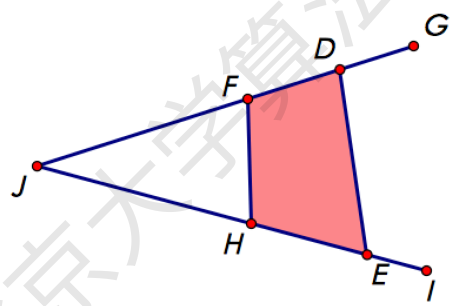
$$\iint \frac{m}{6S(A)}S(P,Q)^{m-1}L(P,Q)^3\ \mathrm{d}u\mathrm{d}\theta$$

一个问题是，$L(P,Q)$ 并不是有理的：平面上两点之间的距离会引入根号。所以需要想办法把 $L(P,Q)$ 的次数变成偶数。不难发现 $\mathrm{d}S = L(P,Q)\mathrm{d}u$，其中 S 表示 $S(P,Q)$。利用它进行换元就能成功改变 $L(P,Q)$ 次数的奇偶性：

$$\iint \frac{m}{6S(A)}S^{m-1}L(P,Q)^2\ \mathrm{d}S\mathrm{d}\theta$$

现在我们回到积分区域：凸多边形 A 上来。显然 $L(P,Q)$ 是一个关于 S 和 θ 的分段函数，它在直线 PQ 与凸多边形正好交在顶点上的时候进行分段，因此 $L(P,Q)$ 是一个 $O(n^2)$ 段的分段函数。一般来说对分段函数的处理方法是，对每一段分别积分，然后把答案累加。

设 PQ 和凸多边形 A 的两个交点是 D 和 E，它们分别在边 FG 和 HI 上。先假设 FG 和 HI 不平行，于是可以延长 FG 和 HI，设交点为 J。在这张图中，$S(P,Q)$ 为 $S(DFHE)$ 再加上凸多边形 A 中 F 到 H 这一段的面积，$L(P,Q)$ 为 $|DE|$ 。不失一般性地，我们假设 J 和四边形 $DFHE$ 在同侧，另一种情况只需要进行换元，变成对 $S(A)-S$ 积分即可。



首先进行一些换元，我们让 S 表示 $S_{\triangle JDE}$，θ 表示 $\angle JDE$：对 S 的换元是简单的平移，被积分的函数并不会发生改变；而对 θ 的换元可能会改变符号，这由 FG 和 HI 的位置决定，需要对每一段都用计算几何的方法进行界定。在这个例子中，θ 的变化方向不变，因此积分负号不变。接着定义一些常量，令 α 表示 $\angle FJH$，那么 $\angle JED = \pi - \alpha - \theta$。

为了进行积分，首要任务是把 $L(P,Q)^2$ 用 S 和 θ 进行表示。我们可以利用 $\triangle JDE$ 的内角、边长以及面积信息，列出如下的方程：

$$\begin{cases} |JD||JE|\sin\alpha = 2S \\ \dfrac{|JD|}{\sin(\alpha+\theta)} = \dfrac{|JE|}{\sin\theta} = \dfrac{|DE|}{\sin\alpha} \end{cases}$$

因此有：

$$L(P,Q)^2 = |DE|^2 = \frac{|JD||JE|\sin^2\alpha}{\sin(\alpha+\theta)\sin\theta} = \frac{2S\sin\alpha}{\sin(\alpha+\theta)\sin\theta}$$

注意这儿的 $\sin\theta$ 和 $\sin(\theta+\alpha)$ 是有可能等于 0 的：发生这种情况当且仅当 $D=F=H=E=J$。幸运的是此时 $S=0$，在假设 $0\ln 0=0$ 的情况下，这样的小瑕疵并不会造成大问题。但是在接下来的推导的某些地方，我们需要针对性的处理这种情况。

在忽略掉一些无关项，首先对 θ 计算一下不定积分：

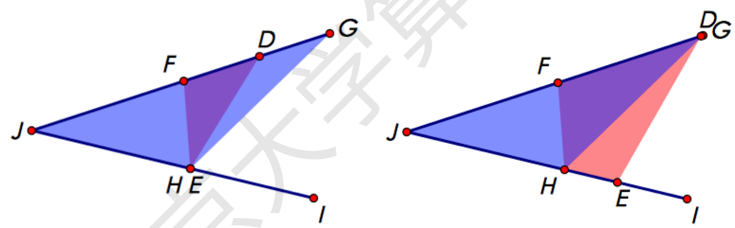
$$\int L(P,Q)^2\mathrm{d}\theta=\int\frac{2S\sin\alpha}{\sin(\alpha+\theta)\sin\theta}\mathrm{d}\theta=2S\ln\left(\frac{\sin\theta}{\sin(\theta+\alpha)}\right)+C$$

现在我们需要来确认 θ 的取值范围，遗憾的是 θ 的上下界都是关于 S 的分段函数。设 θ 的上界是 $R_\theta(S)$，下界是 $L_\theta(S)$，那么答案就是：

$$\frac{m}{6S(A)}\left(\int 2S(S+\Delta S)^{m-1}\ln\left(\frac{\sin R_\theta(S)\sin(L_\theta(S)+\alpha)}{\sin L_\theta(S)\sin(R_\theta(S)+\alpha)}\right)\mathrm{d}S\right)$$

注意因为我们对 S 已经进行了换元，所以在带回到原式子的时候，需要给原来的 S 加上偏移量：这个偏移量等于凸包上 F 到 H 这一段的面积再减去 $S_{\triangle JHF}$。这一个偏移量是一个常数，可以先计算出来。

因为上界部分和下界部分有着几乎一样的形式，所以我们先考虑下界。考虑 θ 大小达到下界的情况：



当 $S\le S_{\triangle JGH}$ 时 $E=H$，而当 $S\ge S_{\triangle JGH}$ 时，$D=G$。根据余弦定理：$\sin\theta/\sin(\theta+\alpha)=|JE|/|JD|$。因此可以得到：

$$\frac{\sin(L_\theta(S)+\alpha)}{\sin L_\theta(S)}=\begin{cases}\dfrac{2S}{|JH|^2\sin\alpha} & S\le S_{\triangle JGH}\\ \dfrac{|JG|^2\sin\alpha}{2S} & S\ge S_{\triangle JGH}\end{cases}$$

同理，关于上界，有：

$$\frac{\sin R_\theta(S)}{\sin(R_\theta(S)+\alpha)}=\begin{cases}\dfrac{2S}{|JF|^2\sin\alpha} & S\le S_{\triangle JFI}\\ \dfrac{|JI|^2\sin\alpha}{2S} & S\ge S_{\triangle JFI}\end{cases}$$

不妨设 $S_{\triangle JGH}\le S_{\triangle JFI}$，那么需要积分的函数是一个三段函数，分界点为 $S_0=S_{\triangle JFH},S_1=S_{\triangle JHG},S_2=$

$S_{\triangle JFI}, S_3 = S_{\triangle IJG}$，那么把每一段分开来就如下图所示：

$$\int_{S_0}^{S_1} 2S(S+\Delta S)^{m-1} \ln\left(\frac{4S^2}{|JF|^2|JH|^2\sin^2\alpha}\right) \mathrm{d}S = \int_{S_0}^{S_1} 4S(S+\Delta S)^{m-1} \ln\left(\frac{S}{S_0}\right) \mathrm{d}S$$
$$\int_{S_1}^{S_2} 2S(S+\Delta S)^{m-1} \ln\left(\frac{|JG|^2}{|JF|^2}\right) \mathrm{d}S$$
$$\int_{S_2}^{S_3} 2S(S+\Delta S)^{m-1} \ln\left(\frac{|JI|^2|JG|^2\sin^2\alpha}{4S^2}\right) \mathrm{d}S = \int_{S_2}^{S_3} 4S(S+\Delta S)^{m-1} \ln\left(\frac{S_3}{S}\right) \mathrm{d}S$$

为了避免接下来出现一些偏差，我们先把先前提到的 $\sin\theta = 0$ 的情况处理掉。之前提到，$\sin\theta = 0$ 当且仅当 $D = F = H = E = J$，因此 $|JF| = |JH| = 0$。幸运的是这时 $S_0 = S_1 = S_2 = 0$，因此我们并不需要考虑前两个积分，只需要计算最后一个就行了，此时：

$$\begin{aligned}
&\int_{0}^{S_3} 4S(S+\Delta S)^{m-1} \ln\left(\frac{S_3}{S}\right) \mathrm{d}S \\
=&\sum_{i=0}^{m-1}\left(4\Delta S^{m-1-i}\binom{m-1}{i}\int_0^{S_3} S^{i+1}(\ln S_3 - \ln S)\mathrm{d}S\right) \\
=&\sum_{i=0}^{m-1}\left(4\Delta S^{m-1-i}\binom{m-1}{i}\left(\frac{S^{i+2}(\ln S_3 - \ln S)}{i+2} + \frac{S^{i+1}}{(i+2)^2}\right)\Bigg|_0^{S_3}\right) \\
=&\sum_{i=0}^{m-1}\left(4\Delta S^{m-1-i}\binom{m-1}{i}\frac{S_3^{i+2}}{(i+2)^2}\right)
\end{aligned}$$

这个式子可以在 $O(m)$ 的时间复杂度内计算得到。因此在接下来的讨论过程中，我们默认 $\sin\theta \neq 0$。重新整理上面的三部分积分，可以得到：

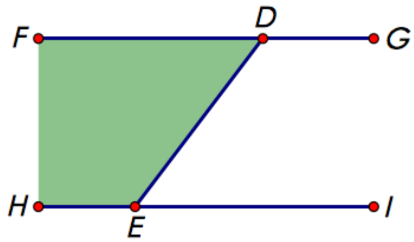
$$\begin{aligned}
&\int_{S_0}^{S_1} 4S(S+\Delta S)^{m-1} \ln\left(\frac{S}{S_0}\right) \mathrm{d}S + \int_{S_1}^{S_2} 2S(S+\Delta S)^{m-1} \ln\left(\frac{|JG|^2}{|JF|^2}\right) \mathrm{d}S + \int_{S_2}^{S_3} 4S(S+\Delta S)^{m-1} \ln\left(\frac{S_3}{S}\right) \mathrm{d}S \\
=&\sum_{i=0}^{m-1}\left(4\Delta S^{m-1-i}\binom{m-1}{i}\left(\int_{S_0}^{S_1} S^{i+1}\ln\left(\frac{S}{S_0}\right)\mathrm{d}x + \int_{S_1}^{S_2} S^{i+1}\ln\left(\frac{|JG|}{|JF|}\right)\mathrm{d}S + \int_{S_2}^{S_3} S^{i+1}\ln\left(\frac{S_3}{S}\right)\mathrm{d}S\right)\right) \\
=&\sum_{i=0}^{m-1}\left(4\Delta S^{m-1-i}\binom{m-1}{i}\left(\frac{S_0^{i+2} - S_1^{i+2} + S_3^{i+2} - S_2^{i+2}}{(i+2)^2} + \frac{S_1^{i+2}}{i+2}\ln\left(\frac{S_1}{S_0}\times\frac{|JF|}{|JG|}\right) + \frac{S_2^{i+2}}{i+2}\ln\left(\frac{S_2}{S_3}\times\frac{|JG|}{|JF|}\right)\right)\right) \\
=&\sum_{i=0}^{m-1}\left(\frac{4}{(i+2)^2}\Delta S^{m-1-i}\binom{m-1}{i}(S_0^{i+2} - S_1^{i+2} + S_3^{i+2} - S_2^{i+2})\right)
\end{aligned}$$

最后一步等号成立是因为 S_1, S_0 和 S_2, S_3 的面积比值正好是着两条边长度的比值，因此乘积为 1，取对数后为 0。于是在这种情况下，我们可以在 $O(m)$ 的时间内积分得到答案。

最后，我们来处理 FG 和 HI 平行的情况，如下图所示：

回顾最开始的积分形式：

$$\iint \frac{m}{6S(A)} S^{m-1} L(P,Q)^2\ \mathrm{d}S\mathrm{d}\theta$$



和前一种情况类似，先对 S 和 θ 作简单的换元，让 S 变成绿色部分的边际，θ 变成 $\angle FDE$，注意在对 θ 换元的时候同样可能改变正负号。在这个例子中，正负号不会发生改变。

接着尝试表示表示 $L(P,Q)$。因为 FG 和 HI 平行，所以长度的表示就变得容易了很多。设平行线之间的距离是 L，那么有 $L(P,Q)^2 = L^2 \sin^{-2}\theta$。对 $L(P,Q)$ 求不定积分得：

$$\int L(P,Q)^2 \mathrm{d}\theta = \int L^2 \sin^{-2}\theta = -\cot\theta + C$$

令 $L_\theta(S)$ 和 $R_\theta(S)$ 分别表示 S 给定时 S 的上界和下界，那么它们仍然还是两段的分段函数：

$$\cot L_\theta(S) = \begin{cases} \dfrac{2S}{L^2} & S \le S_{\triangle FHG} \\ \dfrac{2(|FG|L - S)}{L^2} & S \ge S_{\triangle FHG} \end{cases} \quad \cot R_\theta(S) = \begin{cases} -\dfrac{2S}{L^2} & S \le S_{\triangle FIH} \\ -\dfrac{2(|HI|L - S)}{L^2} & S \ge S_{\triangle FIH} \end{cases}$$

不妨假设 $S_{\triangle FHG} \le S_{\triangle FIH}$。令 $S_1 = S_{\triangle FHG}, S_2 = S_{\triangle FIH}, S_3$ 为四边形 $FHIG$ 的面积，在提出公共的常数项 $\frac{2m}{3S(A)L^2}$ 后，积分可以写成三段：

$$\int_{S_0}^{S_1} 2(S+\Delta S)^{m-1} S \mathrm{d}S + \int_{S_1}^{S_2} |FG|L(S+\Delta S)^{m-1} \mathrm{d}S + \int_{S_2}^{S_3} 2(S+\Delta S)^{m-1}(S_3 - S)\mathrm{d}S$$

这一部分积分涉及到的全是有理数，并没有什么困难的地方：直接把 $m-1$ 次幂二项式展开然后分别积分就可以了。这一步积分同样可以在 $O(m)$ 的时间内完成。

综上所述，对于每一段的积分，时间复杂度是 $O(m)$ 的。因为一共有 $O(n^2)$ 段，所以总的时间复杂度为 $O(n^2m)$。

C Parade

虽然本题题目里规定了每个格子不能有多个点，但实际上这是无所谓的，就算方案中某个时刻某个格子有重复的点，我们也可以通过调整，使得他变成一个合法的方案，所以本题我们在做的时候可以认为没有这条规定。

具体的调整方法大概是：某个时刻 A 的点如果要走到 B（有一个点）再走到 C（没有点）的话，等价于让 B 的点走到 C，再让 A 的点走到 B。

所以我们只要给每个点分配好终点，那么总的答案就是每个点的位置和终点的曼哈顿距离之和。

那么我们一开始先进行一些处理：首先我们可以把所有点都先走到矩形的边缘，例如 $(114514, -5)$ 可以先走到 $(N, 1)$，这样我们就把所有点集中到矩形内部了，我们设 $a_{i,j}$ 表示 (i, j) 上有几个点，设 $b_{i,j}$ 表示 (i, j) 还需要几个点，即 $b_{i,j} = 1 - a_{i,j}$

然后我们考虑一个答案的下界：$\sum_{i=1}^{n} |\sum_{j=1}^{i}(b_{j,1} + b_{j,2})|$

这个下界是什么意思呢？考虑第 k 列到第 $k+1$ 列的这条竖边，我们可以发现至少有 $|\sum_{i=1}^{k}(b_{i,1} + b_{i,2})|$ 个点要经过。所以上面那个式子就是答案的下界，准确来说是棋子进行左右移动的次数的下界。

那么我们能否让这个下界就变成答案呢？可以发现，对于一条边，肯定不会既有从左往右的点，又有从右往左的点的情况。所以上面的式子就是我们左右移动的最少次数。

考虑在满足左右移动次数最少的情况下去最小化上下移动次数。

我们考虑 $b_{1,1}$ 和 $b_{1,2}$ 设他们为 A, B

当 $A \le 0, B \le 0$ 时，相当于这一列有多余的点要往右移动，所以我们可以等移完了之后再进行上下移动，没必要在第一列就上下移动。

当 $A \ge 0, B \ge 0$ 时，相当于这一列两个位置都要从右边借点，那么我们可以在移来之前就完成上下移动，所以没必要在第一列上下移动。

当 A, B 异号时，我们可以直接用上下移动来让一个去补另一个，否则第 1 列和第 2 列之间的边肯定会既有从左到右，又有从右到左的点（因为异号，所以不内部解决的话会导致又要出口又要进口），所以这种情况下上下移动的次数就是 $\min(|A|, |B|)$

于是我们解决了第一列，之后的操作都不会涉及第一列，所以递归成了 $n-1$ 列的子问题。

时间复杂度：$O(n)$

D Circuit

首先考虑第一部分：求出数组 A 和 B。这是一个经典的区间加的问题，考虑用差分的方法。以 A 为例：令 A' 为 A 的差分数组，定义为 $A'_i = A_i - A_{i-1}$。那么在对 A 的下表区间 $[l, r]$ 加上 x 之后，A'_l 加上了 x，A'_{r+1} 减去了 x，其余位置不变。因此对于每一个操作，用 $O(1)$ 的时间就可以维护差分数组的变化。最后用前缀和的方法 $A_i = A_{i-1} + A'_i$ 就可以还原出数组 A。

同样对于最后一部分，询问一个数组的区间和，也可以用前缀和的方法。令 C' 为 C 的前缀和数组：$C'_i = C'_{i-1} + C_i$。那么 C 区间 $[l, r]$ 中所有数组的和就等于 $C'_r - C'_{l-1}$。这样每一组询问可以用 $O(1)$ 的时间回答。

这两部分都非常 trivial，问题的核心在于中间的部分：从 A 和 B 得到 C。这是一个异或卷积问题，可以用快速沃尔什变换（Fast Walsh-Hadamard Transform）在 $O(n2^n)$ 的时间复杂度内解决。下面给出针对异或卷积的一个简单推导，其他的情况（其他二进制操作的卷积）几乎相同。

令 $N = 2^{n-1}$，$A(x) = \sum_{i=0}^{2N-1} A_i x^i$，$A_l(x) = \sum_{i=0}^{N-1} A_i x^i$，$A_r(x) = \sum_{i=0}^{N-1} A_{i+N} x^i$。顾名思义，$A_l(x)$ 和 $A_r(x)$ 分别是 $A(x)$ 的前半段和后半段。类似地，可以定义 $B(x), B_l(x), B_r(x)$。为了方便，我们同样用操作符 $\oplus$ 来表示多项式之间的异或卷积。

在这些定义的基础上，我们可以进行如下推导：

$$
\begin{aligned}
C(x) = A(x) \oplus B(x) &= \sum_{i=0}^{2N-1} \sum_{j=0}^{2N-1} A_j \times B_{i \oplus j} \times x^i \\
&= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (A_j B_{i \oplus j} + A_{j+N} B_{(i \oplus j)+N}) \times x^i + \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (A_j B_{(i \oplus j)+N} + A_{j+N} B_{(i \oplus j)}) \times x^{i+N} \\
&= (A_l(x) \oplus B_l(x) + A_r(x) \oplus B_r(x)) + (A_l(x) \oplus B_r(x) + A_r(x) \oplus B_l(x)) \times x^N
\end{aligned}
$$

因此只要能计算出 $(A_l(x) \oplus B_l(x) + A_r(x) \oplus B_r(x))$ 和 $(A_l(x) \oplus B_r(x) + A_r(x) \oplus B_l(x))$，就可以合并出 $C(x)$ 的所有系数。通过简单的构造，令 $A_+(x) = A_l(x) + A_r(x), A_-(x) = A_l(x) - A_r(x)$，$B_+(x), B_-(x)$ 同理，则可以得到：

$$
\begin{aligned}
A_l(x) \oplus B_l(x) + A_r(x) \oplus B_r(x) &= (A_+(x) \oplus B_+(x) + A_-(x) \oplus B_-(x))/2 \\
A_l(x) \oplus B_r(x) + A_r(x) \oplus B_l(x) &= (A_+(x) \oplus B_+(x) - A_-(x) \oplus B_-(x))/2
\end{aligned}
$$

因此只要递归计算 $A_+(x) \oplus B_+(x)$ 和 $A_-(x) \oplus B_-(x)$ 就可以了，相当于从一个规模为 n 的问题递归到了两个规模为 $\frac{n}{2}$ 的问题。解 $T(n) = O(n) + 2T(\frac{n}{2})$ 得，时间复杂度为 $O(n \log n)$。

如果你熟悉分治乘法的话，可以发现这个做法和分治乘法非常像。唯一的区别在于它利用了异或卷积长度不变的性质，只递归到了两个子问题，因此复杂度从 $O(n^{\log_2 3})$ 变成了 $O(n \log n)$。

E Coprime

考虑我们要求的东西，即 $\sum_{i=l}^{r}[\gcd(a_i,x)==1]$。

首先先进行非常套路化的推导

$$\begin{aligned}&\sum_{i=l}^{r}[\gcd(a_i,x)==1]\\&=\sum_{i=l}^{r}\sum_{d|\gcd(a_i,x)}\mu(d)\\&=\sum_{d|x}\mu(d)\sum_{i=l}^{r}[d|a_i]\\&=\sum_{d|x}\mu(d)(cnt_r(d)-cnt_{l-1}(d))\end{aligned}$$

，其中 $cnt_i(d)$ 表示 a 的前 i 个数中是 d 的倍数的数的个数。

不难发现，我们在预处理 μ 数组后，只需要能快速地求出 cnt_i 数组，我们就可以在 $O(\sqrt{x})$ 的时间内回答一组询问。

其次，我们不难发现，cnt_i 这个数组相比于 cnt_{i-1} 这个数组，一定只有至多有 $O(\sqrt{a_i})$ 个地方的值会 +1。（因为 $cnt_i(d)$ 相比于 $cnt_{i-1}(d)$ 会 +1 当且仅当 $d|a_i$。）所以对于每个 $cnt(d)$, 我们都可以开一个 set S_d 来记录在哪些位置 i,$cnt_i(d)$ 加了一。而当询问 $cnt_i(d)$ 的时候，我们只需要在 S_d 里找 $\leq i$ 的最大的位置，采用那个位置时候的值即可。(熟悉数据结构这一套理论的同学不难发现，这其实就是用 set 来可持久化了数组 cnt_i, 来把一个原来离线的做法拓展到了在线)。

而至于复杂度，由以上的分析易知复杂度为 $O(n\sqrt{a_i}\log n)$，可以通过本题。

F Graduation

首先一共最多只有 10 门课，所以我们可以枚举所有的选课方案，这样大概只需要枚举 2^{10} 种方案，是可以接受的。

之后根据题目描述，我们要检查以下几种情况：

1. 英语课，政治课，体育课是否最多只选了一门
2. 所选的课的学分是否小于等于 25
3. 所选的课之间有没有时间上的冲突
4. 是否满足毕业要求

这些都是可以直接枚举的。

时间复杂度：$O(N^2 2^N)$

G Go and Oreo

首先我们思考一下，Oreo 的本质是什么

竖着的一个 Oreo，等价于一个黑子上面第一个是白子，且他上面还有黑子。

也就是说，(i,j) 是一个竖着的 Oreo 的底部的充要条件是：

1. $(i-1,j)$ 是白子，(i,j) 是黑子

2. $(1..i-1,j)$ 中有至少一个黑子

同样的，(i,j) 是一个横着的 Oreo 的右边的充要条件是：

1. $(i,j-1)$ 是白子，(i,j) 是黑子

2. $(i,1..j-1)$ 中有至少一个黑子

我们一行一行地去填棋子，每行从左到右填：

假设我们现在需要填 (i,j)，那么我们事先已经知道了 $(i,1..j-1)$ 和前 $i-1$ 行所有棋子的颜色，然后根据上面两种 Oreo 的充要条件，可以计算出 (i,j) 贡献了几个 Oreo。

这样我们可以得到一个非常 trivial 的状压动态规划：按照上面的顺序填子，把已经填过的子的颜色都给记下来，时间复杂度：$O(n^2 2^{n^2})$

然而这个复杂度显然是不能接受的。

其实我们可以发现，对于每一列，上面几个的棋子颜色我们是不 care 的，我们只 care 最后一个棋子的颜色，以及这些棋子中是否有黑色的棋子。

所以对于每一列，我们记录两个东西：

1. 这一列已经填的棋子中，是否有黑色棋子

2. 这一列已经填的棋子中，最下面的棋子的颜色

看上去对于每一列有 4 种情况，所以状态数是 4^n，但是可以发现最后一个棋子是黑色时，这一列一定有黑色棋子，所以一共只有 3 种状态。

于是我们得到了一个 $O(n^2 3^n)$ 的算法，可以通过本题。

H Homomorphism

定义 G_n 表示所有 n 个点无向图的集合。

定义 Φ 表示所有 $[n] \to [m]$ 的映射构成的集合。

定义 $H(G,\varphi)$ 表示 φ 是不是 G 到 T 的同态，如果是同态那么 $H(G,\varphi)=1$, 否则 $H(G,\varphi)=0$。

问题就是要计算:

$$\begin{aligned}&\sum_{G\in G_n}\sum_{\varphi\in\Phi} H(G,\varphi)p^{|E(G)|}(1-p)^{\binom{n}{2}-|E(G)|}\\ =&\sum_{\varphi\in\Phi}\sum_{G\in G_n} H(G,\varphi)p^{|E(G)|}(1-p)^{\binom{n}{2}-|E(G)|}\end{aligned}$$

这意味着，我们可以枚举 φ，然后计算所有可以让 φ 成为同态的图的概率和 $S(\varphi)$。假设 φ 已知。定义 $W_u=\{i\in[n]:\varphi(i)=u\}$。对于树上没有直接相连的两个点 u,v，$\forall a\in W_u, b\in W_v$, G 中必然不存在 a 到 b 的边。因此在 φ 固定的情况下，$S(\varphi)$ 可以写成：

$$\sum_{u,v\notin E(T)}(1-p)^{|W_u||W_v|}=(1-p)^{\binom{n}{2}}\sum_{u,v\in E(T)}(1-p)^{-|W_u||W_v|}$$

我们发现，$S(\varphi)$ 只和每个 W_i 集合的大小有关。同时，固定每个 W_i 集合的大小 a_i, 对应的 φ 个数是 $\frac{n!}{\prod_{i=1}^{n} a_i!}$。

因此我们可以用类似于树上背包的动态规划解决这个问题：用 $f[u][j][k]$ 表示考虑 T 中 u 的子树，$|W_u|=j$，u 子树内 W 大小的和 k 的概率和。初始 $f[u][j][j]=\frac{1}{j!}$，和子树合并时的转移方程是 (u 是父亲,v 是孩子):

$$f_{new}[u][i][j]=\sum_{a=0}^{j}\sum_{b=0}^{a} f_{old}[u][i][j-a]\cdot f[v][b][a]\cdot(1-p)^{-ib}$$

最后答案就是 $n!(1-p)^{\binom{n}{2}}\sum_{i=0}^{n} f[1][i][n]$(假设根是 1)。

时间复杂度 $O(n^5)$。

I Chamber of Braziers

问题为求数列 a 中，长度为 k 的，且其中数恰好为 $1 \sim k$ 的连续子序列，要求输出最大的满足条件的 k。

首先，问题要求所选的连续子序列不包含相同的数，所以我们可以先通过一次预处理求出数组 f_i 表示以 i 为结尾的，最长的不包含相同数的连续子序列的开头位置。这部分只要用一个数组 g_j 维护数 j 上次出现的位置，并每次用 $\max(f_{i-1}, g_{a_i})$ 来更新 f_i 即可。

如果有一段长度为 k 的连续子序列不包含相同的数，且其中最大的数大小不超过 k，那么其中的数一定恰好是 $1 \sim k$，即为满足题目要求的连续子序列。现在我们可以 $O(1)$ 判断一个序列是否包含相同的数，问题的难点就转移到了如何保证序列中最大数的大小不超过 k。

容易想到我们可以从小到大枚举数列中的数。当枚举到 k 时，我们可以用并查集将所有连续的，满足 $a_j \leq k$ 的数并入一个集合中。

每当一个新集合 $[l, r]$ 产生，我们根据如下规则判断其中是否有合法的连续子序列：

- 首先序列长度 $r - l + 1 \geq k$，如果不满足则显然不包含，跳过以下两步。
- 判断是否有 $f[l + k - 1] \geq k$，如果有，则说明该区间前 k 项不包含相同的数，为合法的连续子序列。如果不满足，则根据第三条规则决定。
- 判断是否存在 $t \in [l + k, r]$，满足 $f[t] = k$，如果有，则说明区间 $[l, r]$ 中包含一个长度为 k 的子序列其中数互不相同，为合法连续子序列。同时，我们注意到显然不会有满足 $f[t] > k$ 的情况，因为区间中只包含 $1 \sim k$ 中的数，所以不可能有连续超过 k 个数互不相同。

前两种情况我们都可以在 $O(1)$ 的时间内快速判定，难点在于第三种情况。

对于第三种情况，我们可以先将所有满足 $f[t] = k$ 的 t 按照从小到大的顺序挂到链表 $vec[k]$ 上，之后我们在进行并查集时，保证按照位置从小到大的顺序枚举每个满足 $a[j] = k$ 的数，这样可以保证我们在 k 确定的情况下生成的新集合 $[l, r]$ 的右边界 r 也一定是递增的，即保证了区间 $[l + k, r]$ 的单调性。所以我们可以用 two pointer 的思想在做并查集的同时枚举 $vec[k]$ 中的所有元素，保证第三步也能在均摊 $O(1)$ 的复杂度内完成。

最后我们发现并查集算法其实可以优化成两个数组，即对于每个集合 $[l, r]$ 我们只要用两个数组 LL, RR 记录左右端点，分别表示为 $RR[l] = r, LL[r] = l$，在合并时更新这两个数组即可。这样就能将算法优化到严格 $O(n)$ 的时间复杂度。

时间复杂度 $O(n)$，空间复杂度 $O(n)$。

J Matrix of Determinants

这个问题等价于求矩阵 A 的伴随矩阵 A^*, 求出 A^* 以后, 将其转置后, 把位置 (i,j) 上的值乘上 $(-1)^{i+j}$ 后输出即可。

如果 A 可逆（$rank(A)=n$），那么 $A^*=\det A\cdot A^{-1}$，此时使用高斯消元计算行列式和逆矩阵即可。

如果 $rank(A)\le n-2$，此时 A 的每个 $n-1$ 阶余子式都是 0。因此 $A^*=\mathbf{0}$。

如果 $rank(A)=n-1$ ，此时 $rank(A^*)=1$。我们引入一些记号:

设 A 的行向量依次是 $a_1,a_2,\ldots,a_n$, A^* 的行向量依次是 $\alpha_1,\alpha_2,\ldots,\alpha_n$, A^* 的列向量依次是 $\beta_1,\beta_2,\ldots,\beta_n$。

注意到只需求出 A^* 的其中一个行向量 α_i 和其中一个列向量 β_j，就可以解出 A^* 剩下位置的值。

下面介绍怎么计算出其中一个行向量 α_i，计算 β_j 同理。

利用高斯消元找到 A 的一个行向量 a_i，使得 a_i 能被剩下的行向量线性表出。接着把矩阵 A 的第 i 行替换成一个随机向量 a_i'（每个元素在 $[0,P-1]$ 之间随机）。

我们将证明: 随机替换以后, 矩阵 A 以高概率变成一个可逆矩阵。这是因为 $rank(A)=n-1$，因此至多只有 P^{n-1} 个向量能够被 A 中剩下的行向量线性表出，因此一个随机向量能被线性表出的概率至多是 $\frac{1}{P}$，在本题中由于 P 很大，这个错误率可以忽略不计。

于是我们在只替换一行的前提下，成功地把 A 变成了一个可逆矩阵。然后对 A 运行可逆矩阵的伴随矩阵计算方法，就可以得到一个错误的伴随矩阵 A'。由于我们没有对矩阵 A 其他行的元素进行任何修改，A' 的第 i 个行向量 α_i' 和 α_i 是一样的。于是我们就求出了 A^* 的一个行向量 α_i。

时间复杂度 $O(n^3)$。

K Winner, Winner, Chicken Dinner

这道题实际上主要的问题是求出面积最大的四边形，剩下的几个问题都十分简单，是非常基础的计算几何：

1. 求出四边形的面积：将四边形划分成两个三角形，然后根据叉积去计算三角形的面积（$(0,0),(x_1,y_1),(x_2,y_2)$ 组成的三角形的面积是 $|x_1y_2 - x_2y_1|/2$）

2. 求出四边形内有几个点：考虑射线法，从点 x 垂直向上射出一条射线，如果这条射线穿过了奇数条边，则该点在多边形内，否则在多边形外

3. 求点 x 到四边形的距离：等价于点 x 到四边形的四条边的最小距离，所以变成了求点到线段的距离，这有两种情况：如果点到直线的垂足在线段上的话，就直接跑到垂足，否则一定是跑到线段的两个端点中的一个

那么现在最重要的问题，如何求出面积最大的四边形 $ABCD$

考虑一个暴力做法：我们枚举四边形的对角线 AC，因为 $S_{ABCD} = S_{ACB} + S_{ACD}$，所以我们相当于要选一个点 B 来最大化 S_{ACB}，然后选一个点 D 来最大化 S_{ACD}

这非常简单，因为根据三角形的面积公式，我们只要找两侧离直线 AC 最远的点即可，也就是在底边固定的情况下最大化高。

这个算法的时间复杂度是 $O(n^3)$ 的，显然是不能接受的，但这个算法让我们发现了一个结论：因为 BD 是离直线 AC 最远的点，所以显然 BD 一定在整个点集的凸包上，然后也可以发现 AC 也在点集的凸包上，所以四个点都在凸包上。

然后我们研究一下凸包的性质：

1. 如果枚举对角线 AC 的话，根据在他的哪侧可以把凸包分成两个区间，可以发现这两个区间到对角线的距离是单峰的。

2. 当我们固定 A 时，随着 C 在凸包上顺时针枚举过去，AC 一侧的最远点在凸包上也是顺时针变化的，也就是具有单调性。

这两个性质都很好证明，画画图也很容易发现。

所以我们固定 A，按顺时针顺序枚举 C，然后维护两侧的最远点 X, Y，每次 C 往顺时针转时，X, Y 都检查一下顺时针顺序下的后面一个点是否比他们优，优的话就替代掉 X, Y，重复到直到无法替代为止。

可以发现，因为 C, X, Y 都最多往顺时针方向动 n 次（也就是最多转一圈），所以时间复杂度是 $O(n^2)$