QOJ#7894

考虑先求一组变括号方案，并将其树形结构建出。假如某个节点拥有一个儿子的括号种类与自己相同，则一定存在至少两种方案（`(...()...)` 和 `(...)(...)`）。在此基础上，如果存在两个儿子，则同样至少有两种方案（`([...][...])` 和 `([...[]...])`）。

则此时只剩下一种情况：`([([...])]) [([(...)])]`，即至多两条根节点种类不同的圆方相间的链，易验证此时变括号方案唯一，复杂度 $O(n)$。

CF402E

对于每个 $a_{i,j} > 0$，我们令 i 向 j 连一条有向边。

那么 A^k 的 (i,j) 这一项大于 0 当且仅当图上有一条 i 到 j 的恰好经过 k 条边的路径。

那么 $\exists k$，使得 $A^k_{i,j} > 0$ 当且仅当图上 i 可以到达 j。

假如对于一组 i,j，上面 k 不存在，那么原题要求不可能满足，否则，现在整张图都是一个强连通块，因为 $\sum_{i=1}^n a_{i,i} > 0$，所以原题要求必然满足。所以我们只需要判定整张图是否强连通即可。

总复杂度 $O(n^2)$。

AGC017D

考虑求以 i 为根的子树 sg 值为 f_i。

考虑已经求出 f_i，那么考虑只给 i 加上一个父亲，此时的 sg 值为多少，发现就是 f_i+1。证明可以考虑归纳，假设对于 i 的真子树这个结论都是对的，那么由于 i 删去一个子树后残余局面的 sg 可以遍取 $0 \dots f_i - 1$，那么由于对于真子树这个结论是对的，所以加上一个父亲后残余局面的 sg 可以遍取 $1 \dots f_i$，而删去父亲和 i 的边后残余局面 $sg = 0$，所以可以发现加上父亲后 $sg = \operatorname{mex}(0,1,\dots,f_i) = f_i + 1$。

那么考虑计算 f_i，发现就是 $\oplus_{v\in son_i}(f_v+1)$。

复杂度 $O(n)$。

PTZ winter 2020 Day2 G

下面设 $w = 16$。

我们需要找到一个串，使得它有两种不同的凑法。

那么我们设 $f_{i,j}$ 表示，我们凑出来了两个串，一个串长度是 x，另一个是 $x+j$，两个串的前 x 个字符完全相等，第二个串的后 j 个字符是第 i 个模式串的长度为 j 的后缀，这种情况下 x 的最小值。

我们把每对 (i,j) 看作点，那么就是一个最短路的形式，注意到边数 $m \propto wn^2$，最短路的复杂度是 $O(m\log n)$，看起来不太能过，但注意到边权很小，每个点最多被更新 w 次，所以复杂度其实是 $O(w^2 n\log n + m)$，能过。

PTZ winter 2020 Day3 F

考虑每个数一定是你先操作到 $\bmod b \le a$，然后等对手减到 $\le b$，然后补刀。那么考虑算出 $a_i = \lceil \frac{(h_i-1)\%b+1}{a} \rceil, b_i = \lceil \frac{h_i}{b} \rceil$。那么如果我们不去抢一个数，它可以送我们 b_i 次机会，否则会提供我们 $b_i - a_i - 1$ 次机会（如果是负数就是要消耗 $1 - b_i + a_i$ 次机会），由于我们是先手，多一次机会，所以我们的要求就是前缀和都大于等于 -1。

那么每次我们都贪心地拿数，如果打完之后前缀和小于 -1，就选择原本选择打的那些怪中 a_i+1 最大的那只怪不打。复杂度 $O(n\log n)$。

CF1667E

设 f_i 为以 i 为根的子树大小超过 $m=\frac{n+1}{2}$ 的方案。

那么有：

$$f_i=\sum_{j=m}^{n}\binom{n-i}{j-1}(i-1)(n-j-1)!(j-1)!$$

其中 $\binom{n-i}{j-1}(j-1)!$ 为选择 $j-1$ 个点挂在 i 的子树内的方案，$(n-j-1)!$ 为子树外的点选父亲的方案，$i-1$ 为 i 选父亲的方案。

$$\begin{aligned}f_i&=(n-i)!(i-1)\sum_{j=m}^{n}\frac{(n-j-1)!}{(n-i-j+1)!}\\&=(n-i)!(i-1)!\sum_{j=m}^{n}\binom{n-j-1}{i-2}\\&=(n-i)!(i-1)!\sum_{j=i-2}^{n-m-1}\binom{j}{i-2}\\&=(n-i)!(i-1)!\binom{n-m}{i-1}\end{aligned}$$

考虑设 g_i 为 i 为重心的方案数。考虑某个点 $j(j>i)$ 向父亲跳的过程，那么 i 是 j 的祖先的概率为 $\frac{1}{i}$。

那么有：

$$g_i=f_i-\frac{\sum_{j=i+1}^{n}g_j}{i}$$

复杂度 $O(n)$。

AGC019F

考虑我们在状态 (n,m) 的决策，假如 $n\ge m$，那么我们选择回答 `yes`，否则是 `no`。

那么我们考虑对于点 (n,m)，假如 $n\ge m$，那么染黑 $(n-1,m),(n,m)$ 之间的边，否则染黑 $(n,m-1),(n,m)$ 之间的边。一种答案情况可以看作一条 (n,m) 走到 $(0,0)$ 的折线，折线上黑边的数量就是答对数量。

对于 $n\neq m$ 连出来的边，这部分可以轻松发现总和永远为 $\max(n,m)$，这部分可以比较容易理解，因为每次 $\max(n,m)$ 减小一时就会让你答对一题。

对于 $n=m$ 连出的边，考虑这需要你到达 (n,m) 后第一步向左，那么 (i,i) 的贡献就是 $\frac{\binom{2i}{i}}{2}\binom{n+m-2i}{n-i}$。

复杂度 $O(\max(n,m))$。

QOJ#7737

如果是一般图，则有一个 $O(nk)$ 个点，$O((n+m)k)$ 条边的网络流做法，不能通过。

但是本题图为平面图，考虑平面图最大流等于对偶图最短路，平面图的对偶图的对偶图皆为其本身，我们将对偶图建出，每次操作相当于给某条边容量加一，最小操作数使最大流增加 k，给每条边多建一个费用为 1，容量无限的复制即可，复杂度 $O((nm)^3+knm\log(nm))$。