海亮1月25日

题单

本人很菜，复盘如果出锅还请轻喷（

QOJ7894

link

题意

给定一个只由圆括号和方括号（即字符集为 $()[]$）的字符串，你现在可以任意地把若干个左括号变成右括号、把若干个右括号变成左括号，但保持括号的种类（圆或方）不变。求是否唯一存在一个变括号的方案，使得括号序列合法。
合法的括号序列由如下过程递归定义：

- $\emptyset$是合法的。
- S 是合法的，则 (S) 和 $[S]$ 是合法的。
- S,T 是合法的，则 S,T 是合法的。

多组测试，字符串长度之和不超过 10^6 。对于每组测试保证至少存在一种变括号的方案使得括号序列合法。

题解

发现一个事情：所有的 $()$ 可以看作相同的（$[]$ 同理）

然后先扫一遍整个序列，如果栈顶两个种类相同就分别赋值成左右括号，否则就先入栈。

然后发现一个性质：如果存在合法序列 A,B,C，那么 $(A(B)C)$，$(A)B(C)$ 一定不唯一（$[A[B]C]$ 和 $[A]B[C]$ 同理），证明显然。

然后发现合法的序列一定形如 $[A](B)$（$(),[]$ 可以 $swap$），其中 A,B 是合法序列且一定是形如 $[([\dots])]$ 的形式。

然后判定一下就好了。

CF402E

link

题意

给出一个矩阵 A，问是否存在一个正整数 k 使得 A^k 的所有元素都是正数。

$2 \leq n \leq 2000, 0 \leq a_{i,j} \leq 50, \sum_{i=1}^{n} a_{i,i} > 0$

题解

我们看一下矩阵乘法的式子：

$$C_{i,j}=\sum_{k=1}^{n}(A_{i,k}\times B_{k,j})$$

然后我们发现，数字的大小没有意义，只需要知道每个数字是不是大于零即可。

然后转化一下式子：

$$C_{i,j}=\begin{cases}1 & \exists k\in[1,n],A_{i,k}=1\ \&\ B_{k,j}=1\\ 0 & \nexists k\in[1,n],A_{i,k}=1\ \&\ B_{k,j}=1\end{cases}$$

看看这个式子是什么？$Floyd$ 的式子！

我们发现，这个 A^k 其实就是恰好经过 k 条边，能否从 $i\to j$。

然后我们发现，k 是没有限制的，换句话说，这道题问的是，经过任意多条边之后，这张图是不是一张**竞赛图**。

然后就没了，拿 $Floyd$ 跑一下就没了，记得 $bitset$ 优化。

AGC017D

link

题意

有一棵 N 个节点的树，节点标号为 $1, 2, \cdots, N$，边用 (x_i, y_i)表示。 Alice 和 Bob 在这棵树上玩一个游戏，Alice先手，两人轮流操作：

选择一条树上存在的边，把它断开使树变成两个连通块。然后把不包含 1 号点的联通块删除

当一个玩家不能操作时输，你需要算出：假如两人都按最优策略操作，谁将获胜。

$N \leq 2 \times 10^5$

保证给出的是一棵树。

题解

设 sg_u 表示子树 u 的 SG 函数值。

但是怎么转移啊？不会啊?

先考虑一种比较简单的情况，就是 u 的儿子的子树都是链。

那么显然可以将儿子的子树看作石子堆，然后发现 $sg_u = \oplus_{u\to v}(sg_v + 1)$。

但是这个结论能不能推到不是链的情况呢?

显然是可以的。你尝试将 u 给每个儿子复制一个出来，每个复制的 u 都只连接一个相对应的儿子。

现在已经分成若干个子游戏了，但是怎么证明每个子游戏的 $sg = sg_v + 1$ 呢?

如果直接断开 u 连向子节点的边，那么显然下一状态的 $sg = 0$。

否则，通过数学归纳法，我们发现这个状态一定能够走遍儿子状态+1的状态。

然后就没了，代码是很简单的。

PTZ winter 2020 Day2 G

link

题意

给你 n 个 01 串 $s_1, s_2, \dots, s_n$，问你是否存在两个下标序列 $p_1, p_2, \dots, p_x$ 和 $q_1, q_2, \dots, q_y$，使得 $S = s_{p_1} + s_{p_2} + \cdots + s_{p_x} = s_{q_1} + s_{q_2} + \cdots + s_{q_y}$（$A + B$ 表示将 01 串 A 和 B 拼接)

如果有解，需要你保证 $|S|$ 尽可能小，输出这个最小值。无解输出 0。

题解

设 $w = \max_{i=1}^{n} |s_i|$

我们设 $dis_{i,j}$ 表示现在两个串中，长度较长的那个结尾是字符串 i，且超出另一个字符串长度 j，当前长度较长的字符串的长度最小是 $dis_{i,j}$。

那么这个就是一个最短路了。

然后最多有 $O(n \times w)$ 中状态，每种状态最多可以转移到 $O(n)$ 中状态。

那么最终的时间复杂度就是 $O(n^2 w)$，能过。

PTZ winter 2020 Day3 F

link

题意

有 n 个小怪，第 i 个小怪的生命值是 h_i。

你的攻击力是 a，对手的攻击力是 b，你先手，双方轮流清小怪。

如果是你的回合，你可以选择打小怪或者不打。

由于对手是人机，所以在对手的回合，他会选择最左边没被打死的小怪，打上一刀。

一个小怪被你打死当且仅当你对小怪攻击后小怪死掉。

请问在最优条件下，你能够打死多少小怪。

题解

发现一个事情：第 i 个小怪如果满足 $h_i \mod b \le a$，那么显然这个小怪能够被你收掉。

我们设 $calc(a,b) = (a-1) \mod b + 1$。（其实就是找到血量为 a 的小怪在被攻击力为 b 的人最后一击打死之前剩下的血量，或者说，）

换句话说

$$calc(a,b) = \begin{cases} (a \mod b) & (a \mod b) > 0 \\ b & (a \mod b) = 0 \end{cases}$$

于是我们设 $x_i = \lceil \frac{calc(h_i, atkb)}{atka} \rceil, y_i = \lceil \frac{h_i}{atkb} \rceil$。

然后发现，如果抢第 i 个小怪，那么会增加 b_i 个机会，否则增加 $b_i - a_i - 1$ 个机会。

然后我们每次贪心，如果无法保证 $s \ge -1$（先手多一次机会），那么每次找到最大的 $a_i + 1$，反悔贪心即可。

然后就没了。

CF1667E

link

题意

对于所有点数为 n 的树，如果其满足 对于所有 $i \in [2, n]$，与 i 相连的 j 中有且只有一个点 j 满足 $j < i$ ，那么我们称其为好树。

对于 $1 \sim n$ 每个点求出来有多少好树满足重心为 i。

这里重心定义为删去这个点后形成的所有连通块大小均不超过 $\frac{n-1}{2}$。

数据范围 $3 \le n \le 2 \times 10^5$ 且 n 为奇数（所以不存在树有多个重心的情况）。

题解

首先让 1 为根，那么希望树是好树就必须使得每个点的父亲编号小于自己的。

然后问题转化为，求使得 i 子树的大小 $\ge m=\frac{n+1}{2}$ 且其他任意点的子树大小 $<m$ 的方案数。

设 f_i 表示子树 i 的大小 $\ge m$ 的方案数。

那么

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

解释：

- j 表示子树 i 的大小是 j 的方案数。
- $\binom{n-i}{j-1}$ 表示在比自己大的 $n-i$ 个点中选出 $j-1$ 个点作为自己子树的点的方案数。
- $(n-j-1)!$ 表示对于剩下的 $(n-j)$ 个点，第 k 大的点有 $(n-j-k)$ 种父亲的选法，那么乘起来就是 $(n-j-1)!$。
- $(j-1)!$ 表示给在子树中的 j 个点选择父亲，等同于上面这个。
- $(i-1)$ 表示需要给第 i 个点选择一个父亲，显然有 $i-1$ 种选法。

刚刚的式子中，对于每一种方案都给每一个点确定了一个父亲，可以构造出一个确定的树。

但是这个式子显然没办法 $O(1)$ 计算，需要再化简化简。

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

然后你发现可以用 $O(1)$ 求出 f_i 了。

但是这玩应也不是我们想要的答案啊！

我们发现，只要子树 i 中没有重心，那么点 i 就一定是重心。

我们设点 i 是重心的方案数是 g_i。

然后我们发现，可以通过从 f_i 中减去 $i < j \leq n$ 且 i 是 j 的祖先中 j 是重心的方案数即可。

我们知道 $i < j \leq n$ 中 j 是重心的方案数，但是不知道其中 i 是 j 的祖先所占的比例。

但是我们可以知道（bushi）

我们发现，对于每一个 j，如果不断的跳到父亲（编号单调递减），那么第一次跳到 $[1, i]$ 的概率是相同的，也就是说，有 $\frac{1}{i}$ 种方案使得 i 成为 j 的父亲，并且对于所有的 j 都是相同的。

那么最终的答案 $g_i = f_i - \frac{\sum_{j=i+1}^{n} g_j}{i}$。

AGC019F

link

题意

有 $N+M$ 个问题，其中有 N 个问题的答案是 `YES`，M 个问题的答案是 `NO`。当你回答一个问题之后，会知道这个问题的答案，求最优策略下期望对多少。

答案对 998244353 取模。

题解

设状态 (i,j) 表示现在还剩 i 个 Yes，j 个 No。将每个状态放在坐标轴上。

我们发现，Yes 和 No 没有本质区别，那么不妨钦定 $N \geq M$

然后我们发现，你一定能够保底猜对 N 个，问题就在于，你在直线 $y=x$ 上的时候对多少。

我们发现，如果将一种问题的正误画成折线，那么我们需要统计的是直线 $y=x$ 被折线经过的次数，显然是 $\sum_{i=1}^{M}\binom{2\times i}{i}\binom{N-i+M-i}{N-i}$。

然后算期望的话，每一步对的概率都是 $\frac{1}{2}$，总的路径数是 $\binom{N+M}{N}$，然后再加上保底的 N 个，就没了。

QOJ7737

link

题意

给定一个 $n \times m$ 的网格图以及正整数 k，即 $(x, y), (x', y')$ 之间有边当且仅当 $|x - x'| + |y - y'| = 1$，每条边有正边权。

你可以进行任意次如下操作：选择一条边，该边权加一。

记 $d = \min_{1 \le p, q \le n}\{dis((p, 1), (q, n))\}$，其中 $dis((x, y), (u, v))$ 为网格图上 $(x, y) \to (u, v)$ 最短路。

求最小操作次数使得 d 至少增加 k。

$2 \le n, m \le 500, 1 \le n \times m \le 500, 1 \le k \le 100$，边权范围 $w \in [1, 10^9]$

题解

首先这是个平面图，先转个对偶图。

然后先跑个最大流。

然后对于每条边，都连接一条容量是 $\inf$，费用是 1 的边，每一个流量表示对这条边操作一次。

然后在汇点后面再加一个超级汇点，容量是 k。

然后跑最小费用最大流即可。

然后就做完了。