杂题选讲

ASDFZ LXM

November 22, 2020

JOISC2019 馕

有一个总长度为 L 厘米的馕，每一厘米是一种口味。有 N 个人，每个人吃不同口味的馕会获得不同的快乐度。现在需要将馕切成 N 份并分给 N 个人，切的位置可以是分数。你需要构造一种方案，使得每个人获得的快乐度都不小于他们独吞整个馕所获得的快乐度的 $\frac{1}{N}$ 。

$1 \le N, L \le 2000$

JOISC2019 馕 Solution

直接做不好做，可以先对每个人求出馕上的 $N-1$ 个分界点，使得馕被分成 N 份，每份的快乐度恰好为 $\frac{1}{N}\times$ 总快乐度，然后要求最后的方案至少包含完整的一份

JOISC2019 馕 Solution

直接做不好做，可以先对每个人求出馕上的 $N-1$ 个分界点，使得馕被分成 N 份，每份的快乐度恰好为 $\frac{1}{N}\times$ 总快乐度，然后要求最后的方案至少包含完整的一份

切馕时可以直接贪心，每次选最靠前的分界点切开，并把这块馕分给对应的人。

JOISC2019 馕 Solution

直接做不好做，可以先对每个人求出馕上的 $N-1$ 个分界点，使得馕被分成 N 份，每份的快乐度恰好为 $\frac{1}{N}\times$ 总快乐度，然后要求最后的方案至少包含完整的一份

切馕时可以直接贪心，每次选最靠前的分界点切开，并把这块馕分给对应的人。

证明：可以发现，如果一个人这次分到了馕，那么它上一个分界点一定比上次切的位置靠后，所以每个人的快乐度都至少是 $\frac{1}{N}$

JOISC2017 幽深府邸

有 N 个房间，编号为 $1 \sim N$ ，编号为 i 和 $i+1$ 的房间之间有门，需要类型为 C_i 的钥匙打开。每个房间有 B_i 个钥匙，不同房间中可能出现相同的钥匙。q 组询问，每次询问一个初始没有钥匙的人能否从编号为 x 的房间走到编号为 y 的房间。

$n, \sum B_i \leq 5 \times 10^5$

JOISC2017 幽深府邸 Solution

显然从每个房间出发，能到达的是一段连续的房间，设编号为 $[l_i, r_i]$

JOISC2017 幽深府邸 Solution

显然从每个房间出发，能到达的是一段连续的房间，设编号为 $[l_i, r_i]$ 考虑分治。设当前已经处理好了 $[L, m]$ 和 $[m+1, R]$ 的区间内部答案，对于所有 $[L, m]$ 中能到达 m 的房间，一定是编号越小范围越大，所以直接从 mid 到 L 枚举，动态扩大范围即可。$[mid+1, R]$ 同理。

JOISC2017 幽深府邸 Solution

显然从每个房间出发，能到达的是一段连续的房间，设编号为 $[l_i, r_i]$

考虑分治。设当前已经处理好了 $[L, m]$ 和 $[m+1, R]$ 的区间内部答案，对于所有 $[L, m]$ 中能到达 m 的房间，一定是编号越小范围越大，所以直接从 mid 到 L 枚举，动态扩大范围即可。$[mid+1, R]$ 同理。

这样就可以处理出 $[L, R]$ 内所有点的 $[l_i, r_i] \cap [L, R]$ 了。

CF1110G Tree-Tac-Toe

两个人在一棵树上下三子棋，先手执白。初始时已经有一些位置有白子。假设两人绝顶聪明，问最后结果如何。

$n \leq 500000$

CF1110G Tree-Tac-Toe Solution

因为先手有优势，所以只可能是先手必胜或平局。

CF1110G Tree-Tac-Toe Solution

因为先手有优势，所以只可能是先手必胜或平局。

首先假设没有位置有白子。

CF1110G Tree-Tac-Toe Solution

因为先手有优势，所以只可能是先手必胜或平局。

首先假设没有位置有白子。

如果有点的度数 ≥ 4 ，那么显然先手必胜。

CF1110G Tree-Tac-Toe Solution

因为先手有优势，所以只可能是先手必胜或平局。

首先假设没有位置有白子。

如果有点的度数 ≥ 4 ，那么显然先手必胜。

如果有点的度数 $= 3$ 且有至少两个邻点不是叶子，那么仍然先手必胜。

CF1110G Tree-Tac-Toe Solution

因为先手有优势，所以只可能是先手必胜或平局。

首先假设没有位置有白子。

如果有点的度数 ≥ 4 ，那么显然先手必胜。

如果有点的度数 $=3$ 且有至少两个邻点不是叶子，那么仍然先手必胜。

否则树的形态只可能有三种：一条链，一条链 $+$ 链的一个端点处连接两个叶子，一条链 $+$ 链的两个端点处都连接两个叶子。前两种情况可以发现一定平局，第三种情况的结果与点数的奇偶性有关。

CF1110G Tree-Tac-Toe Solution

因为先手有优势，所以只可能是先手必胜或平局。
首先假设没有位置有白子。
如果有点的度数 ≥ 4 ，那么显然先手必胜。
如果有点的度数 $=3$ 且有至少两个邻点不是叶子，那么仍然先手必胜。
否则树的形态只可能有三种：一条链，一条链 $+$ 链的一个端点处连接两个叶子，一条链 $+$ 链的两个端点处都连接两个叶子。前两种情况可以发现一定平局，第三种情况的结果与点数的奇偶性有关。
如果初始有白子，那么可以通过在这里挂一个三个点的子树转化成没有白点的情况。

UOJ529 114514

你有一个长度为 $6N$ 的字符串，你要将它拆分成 N 个大小为 6 的子序列，使得每个子序列都是 114514 。保证有解。
$n \leq 6 \times 10^5$

UOJ529 114514 Solution

5 是唯一确定的。5 前后都有 14 ，可以考虑将 14 配对。

UOJ529 114514 Solution

5 是唯一确定的。5 前后都有 14 ，可以考虑将 14 配对。

如果一个 14 包含了另外一个 14 ，那么将它们改为相交一定不会更劣。

UOJ529 114514 Solution

5 是唯一确定的。5 前后都有 14 ，可以考虑将 14 配对。
如果一个 14 包含了另外一个 14 ，那么将它们改为相交一定不会更劣。
所以可以按下标从大到小，如果遇到一个 1 则找到最靠后的一个 4 与它配对。如果没有这样的 4 ，则把这个 1 当作某个子序列开头的 1 。这样同时能保证开头的 1 尽量靠前。

UOJ529 114514 Solution

5 是唯一确定的。5 前后都有 14 ，可以考虑将 14 配对。

如果一个 14 包含了另外一个 14 ，那么将它们改为相交一定不会更劣。

所以可以按下标从大到小，如果遇到一个 1 则找到最靠后的一个 4 与它配对。如果没有这样的 4 ，则把这个 1 当作某个子序列开头的 1 。这样同时能保证开头的 1 尽量靠前。

配对好 14 后需要将 5 和前后的 14 配对，此时 5 已经和最开头的 1 配对过了。仍然可以贪心，从小到大，对每个 5 找到最靠前的没被使用过的 14 与它配对（要保证在开头的 1 之后），从大到小对每个 5 找到最靠后的没被使用过的 14 与它配对。

ARC102D Revenge of BBuBBBlesort!

给定长度为 n 的排列，可以进行无限次操作，问最终能否将其排成升序.
其中，一次操作定义为：选择 i 使得 $p_{i-1} < p_i < p_{i+1}$ 并且交换 p_{i-1} 和 p_{i+1} 。
$n \le 3 \times 10^5$

ARC102D Revenge of BBuBBBlesort! Solution

有一个重要的性质：被操作的点被操作后就不可能被移走。

ARC102D Revenge of BBuBBBlesort! Solution

有一个重要的性质：被操作的点被操作后就不可能被移走。

证明：如果 p_i 被操作了一次，那么满足 $p_{i-1} < p_i < p_{i+1}$ 。假设之后某次操作从左边被换走了，那么 $p_{i-2} > p_{i-1} > p_i$ ，所以要先把 p_{i-1} 换成更大的。但换 p_{i-1} 之后一定满足 $p_{i-3} < p_{i-2} < p_{i-1}$ ，此时就不可能 $p_{i-2} > p_{i-1}$ 了。

ARC102D Revenge of BBuBBBlesort! Solution

有一个重要的性质：被操作的点被操作后就不可能被移走。
证明：如果 p_i 被操作了一次，那么满足 $p_{i-1} < p_i < p_{i+1}$ 。假设之后某次操作从左边被换走了，那么 $p_{i-2} > p_{i-1} > p_i$ ，所以要先把 p_{i-1} 换成更大的。但换 p_{i-1} 之后一定满足 $p_{i-3} < p_{i-2} < p_{i-1}$ ，此时就不可能 $p_{i-2} > p_{i-1}$ 了。
所以只可能操作 $p_i = i$ 的点。

ARC102D Revenge of BBuBBBlesort! Solution

有一个重要的性质：被操作的点被操作后就不可能被移走。
证明：如果 p_i 被操作了一次，那么满足 $p_{i-1} < p_i < p_{i+1}$ 。假设之后某次操作从左边被换走了，那么 $p_{i-2} > p_{i-1} > p_i$ ，所以要先把 p_{i-1} 换成更大的。但换 p_{i-1} 之后一定满足 $p_{i-3} < p_{i-2} < p_{i-1}$ ，此时就不可能 $p_{i-2} > p_{i-1}$ 了。
所以只可能操作 $p_i = i$ 的点。
我们设 $b_i = [p_i = i]$ 。显然对于相邻且 b 值相等的两个位置无法进行操作（所以有 b 中有连续三个 0 一定无解），这样我们就把原序列分成了若干个 b 值 01 交替的子串。每个子串可以单独考虑。

ARC102D Revenge of BBuBBBlesort! Solution

首先，子串内部的点一定要是值域连续段（因为子串和子串之间无法交换）。

ARC102D Revenge of BBuBBBlesort! Solution

首先，子串内部的点一定要是值域连续段（因为子串和子串之间无法交换）。

其次，$p_i \neq i$ 的点如果有长度 ≥ 3 的下降子序列那么一定无解。

ARC102D Revenge of BBuBBBlesort! Solution

首先，子串内部的点一定要是值域连续段（因为子串和子串之间无法交换）。

其次，$p_i \neq i$ 的点如果有长度 ≥ 3 的下降子序列那么一定无解。

证明：如果存在三个位置使得 $x < y < z, p_x > p_y > p_z$ ，那么 p_x 移到 p_y 旁后交换 x, y 一定要用一个 w 使得 $p_x > p_w > p_y$ ，此时 p_z 就没有办法换到 p_x 前面了。

ARC102D Revenge of BBuBBBlesort! Solution

首先，子串内部的点一定要是值域连续段（因为子串和子串之间无法交换）。

其次，$p_i \neq i$ 的点如果有长度 ≥ 3 的下降子序列那么一定无解。

证明：如果存在三个位置使得 $x < y < z, p_x > p_y > p_z$ ，那么 p_x 移到 p_y 旁后交换 x, y 一定要用一个 w 使得 $p_x > p_w > p_y$ ，此时 p_z 就没有办法换到 p_x 前面了。

最长下降子序列 ≤ 2 的序列大致为 $1, 4, 2, 3, 9, 5, 6, 7, 8...$ （即从 $1 \sim n$ 的序列中选若干不交子区间循环右移），插入奇数位 $p_i = i$ 的点后大致为 $1, 2, 3, 8, 5, 4, 7, 6, 9, 18, 11, 13, 12, 15, 14, 17, 16$ ，所以就只需要把 $p_i > i$ 的点往后移就可以了。

CF1237H Balanced Reversals

给定两个长度为 N 的 01 串 A 和 B，满足 N 为偶数。你需要翻转若干次 A 的前缀（前缀长度也要是偶数），使 $A = B$ 。给出构造方案或判定无解。

$n \leq 4000, 2|n$

CF1237H Balanced Reversals Solution

如果把相邻两个分组，那么 00 和 11 无论怎么翻转都不会改变，所以 A 和 B 中如果 00 和 11 数量不同一定无解。

CF1237H Balanced Reversals Solution

如果把相邻两个分组，那么 00 和 11 无论怎么翻转都不会改变，所以 A 和 B 中如果 00 和 11 数量不同一定无解。
然后要先让 AB 中 01 和 10 的数量相同。这个可以通过翻转 01 和 10 差的绝对值大的那个串的某个前缀做到。假设 01 用 1 表示，10 用 -1 表示，那么依次翻转 A 的每个前缀得到的 A 的和一定是连续的，最终翻转整个 A 可以让 A 的和取相反数，所以在中间某个时刻 A 的和一定能和 B 的和相同。

CF1237H Balanced Reversals Solution

然后可以把 A 两位两位的匹配上 B 。为了方便，可以先倒过来匹配。假设前 $i-2$ 位已经匹配好了，B 中的第 $i-1$ 和 i 与 A 中的 $j-1$ 和 j 相同，那么只要翻转长度为 $j-2$ 和 j 的前缀即可把 $j-1$ 和 j 翻转到 A 的最前面并且其余字符顺序不变。最后翻转一次长度为 $N-2$ 的前缀即可（因为 $N-1$ 和 N 的位置一定是匹配好的）。

CF1237H Balanced Reversals Solution

然后可以把 A 两位两位的匹配上 B 。为了方便，可以先倒过来匹配。假设前 $i-2$ 位已经匹配好了，B 中的第 $i-1$ 和 i 与 A 中的 $j-1$ 和 j 相同，那么只要翻转长度为 $j-2$ 和 j 的前缀即可把 $j-1$ 和 j 翻转到 A 的最前面并且其余字符顺序不变。最后翻转一次长度为 $N-2$ 的前缀即可（因为 $N-1$ 和 N 的位置一定是匹配好的）。

总操作次数为 N 次。

CF gym102268 K

给定一个长度为 n ，只包含 ab 的串。你每次可以在字符串任意位置加或删掉 aa bbb ababab 中的一个。初始有一个串，问经过任意次操作后可以得到的长度为 x 的串有多少个。

$n \leq 300000, x \leq 10^9$

CF gym102268 K Solution

可以猜出本质不同的 ab 串不是很多。

CF gym102268 K Solution

可以猜出本质不同的 ab 串不是很多。

如果用置换群表示本质不同的 ab 串，A 表示字符 a ，B 表示字符 b ，那么需要满足 $A^3=B^2=(AB)^3=\epsilon$ ，可以凑出 $A=\begin{Bmatrix}1,2,3,4\\2,1,4,3\end{Bmatrix}$ ，$B=\begin{Bmatrix}1,2,3,4\\3,1,2,4\end{Bmatrix}$ 是一组合法的解。这样删掉一个串等同于除掉一个 ϵ ，加上一个串等同于乘一个 ϵ 。

CF gym102268 K Solution

可以猜出本质不同的 ab 串不是很多。

如果用置换群表示本质不同的 ab 串，A 表示字符 a，B 表示字符 b，那么需要满足 $A^3 = B^2 = (AB)^3 = \epsilon$，可以凑出 $A = \begin{Bmatrix} 1,2,3,4 \\ 2,1,4,3 \end{Bmatrix}$，$B = \begin{Bmatrix} 1,2,3,4 \\ 3,1,2,4 \end{Bmatrix}$ 是一组合法的解。这样删掉一个串等同于除掉一个 ϵ，加上一个串等同于乘一个 ϵ。

可以通过暴力验证出不能被完全删掉的串所代表的置换一定不是 ϵ。

CF gym102268 K Solution

可以猜出本质不同的 ab 串不是很多。

如果用置换群表示本质不同的 ab 串， A 表示字符 a ， B 表示字符 b ，那么需要满足 $A^3 = B^2 = (AB)^3 = \epsilon$ ，可以凑出 $A = \begin{Bmatrix} 1,2,3,4 \\ 2,1,4,3 \end{Bmatrix}$ ， $B = \begin{Bmatrix} 1,2,3,4 \\ 3,1,2,4 \end{Bmatrix}$ 是一组合法的解。这样删掉一个串等同于除掉一个 ϵ ，加上一个串等同于乘一个 ϵ 。

可以通过暴力验证出不能被完全删掉的串所代表的置换一定不是 ϵ 。

所以可以得到一个自动机，识别出所有串对应的 12 种基本串。注意到 AB 为偶置换，所以自动机的点数为 $\frac{4!}{2} = 12$ 。

CF gym102268 K Solution

可以猜出本质不同的 ab 串不是很多。

如果用置换群表示本质不同的 ab 串，A 表示字符 a ，B 表示字符 b ，那么需要满足 $A^3 = B^2 = (AB)^3 = \epsilon$ ，可以凑出 $A = \begin{Bmatrix} 1, 2, 3, 4 \\ 2, 1, 4, 3 \end{Bmatrix}$ ，$B = \begin{Bmatrix} 1, 2, 3, 4 \\ 3, 1, 2, 4 \end{Bmatrix}$ 是一组合法的解。这样删掉一个串等同于除掉一个 ϵ ，加上一个串等同于乘一个 ϵ 。

可以通过暴力验证出不能被完全删掉的串所代表的置换一定不是 ϵ 。

所以可以得到一个自动机，识别出所有串对应的 12 种基本串。注意到 AB 为偶置换，所以自动机的点数为 $\frac{4!}{2} = 12$ 。

给定串只需要在自动机上走就可以得到其代表的基本串。然后对自动机做矩阵快速幂，就可以得到长度为 x 的序列个数了。