CCPC 2022 桂林站参考题解

主办方：桂林电子科技大学
题目作者：上海交通大学

2022 年 10 月 30 日

总览

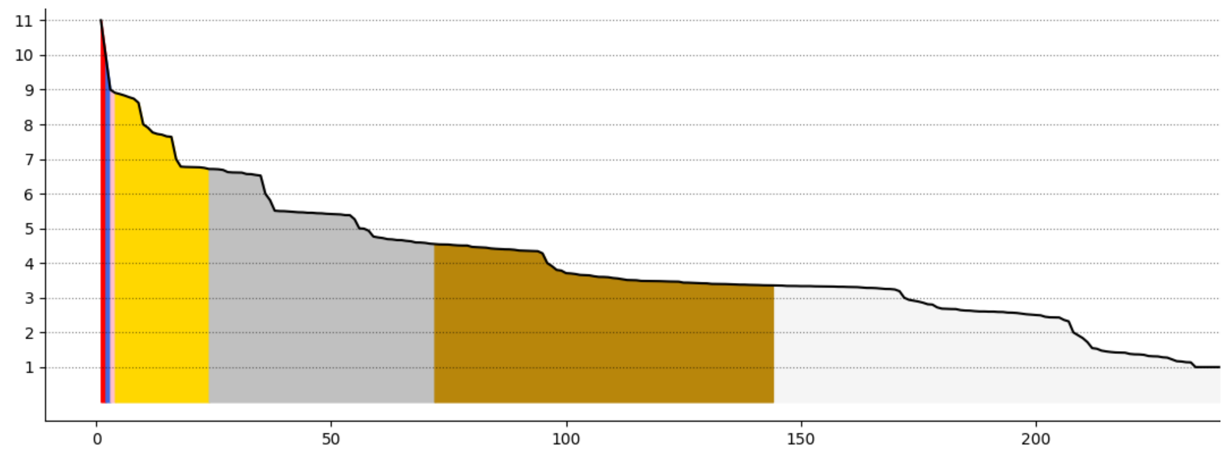

图 1: 排名分布情况

题目难度估计

Easy	A, M
Easy-Mid	C, E, G
Medium	<u>B</u>, H, J, L
Mid-Hard	D, $\overline{\text{K}}$
Hard	*F*, *I*

排名分布

- 冠军：11 题 北京大学 逆十字
- 亚军：10 题 浙江大学 *nameless story*
- 季军：9 题 北京理工大学 *ddl* 战神
- 金牌：7 题
- 银牌：5 题
- 铜牌：4 题

A. Lily

作者 fstqwq　**一血** 北京邮电大学 三只蒟蒻

题解　所有不在 `L` 旁的字符替换为 `C` 即可。

B. Code With No Forces

作者 fstqwq　**一血** 北京大学 逆十字

题解　我们考虑对于每个题用 3 个位来表示：是否状态满足、时间满足与空间满足。随后，我们考虑依次接受某个测试点。

- 如果一个非 **correct** 提交状态没有被满足，那么仅能接受 **correct** 和自己对应状态的提交，并且时间和空间不能超过最终结果。
- 非 **correct** 提交接受对应状态时，应当在接受结果后满足时间与空间限制。
- 如果一个非 **correct** 提交状态已经被满足了，那么该测试点已经与他无关了。
- **correct** 提交仅需要考虑时间和空间。

将上述内容写为压位 DP，使用位运算加速判断可以做到 $O(n2^{3m})$。当然，时限比较宽裕，$O(nm2^{3m})$ 亦存在通过可能性。

C. Array Concatenation

作者 relyt871　**一血** 浙江大学 *Phantom Threshold*

题解

做法 1　可以发现，一旦使用了一次第二种操作，整个数组会变成回文的，于是之后两种操作就没有区别了。枚举在第几次操作时第一次使用第二种操作即可，最终只有 $m+1$ 种不同的前缀和之和。

对于一个长度为 n，总和为 s，前缀和之和为 ss 的数组，执行 k 次第一种操作后的前缀和之和为

$$ss \cdot 2^k + n \cdot s \cdot \frac{2^k \cdot (2^k - 1)}{2}$$

这个式子可以 $O(1)$ 计算，于是上述算法可以 $O(m)$ 模拟。

做法 2　进一步观察性质，可以发现，只要使用了第二种操作，无论什么时候第一次使用，最终的前缀和之和都是相同的，所以只有两种本质不同的前缀和之和。时间复杂度 $O(1)$。

D. Alice's Dolls

作者 AntiLeaf　**一血** 华东师范大学 *SEI1*

题解　方便起见令 $p = \frac{a}{b}, q = 1 - p$。

做法 1　由二项式定理可知

$$E\left((x+y)^k\right) = \sum_{i=0}^{k} \binom{k}{i} E\left(x^i\right) E\left(y^{k-i}\right)$$

那么如果能求出 $n = 1$ 的答案，只需借用一个类似快速幂的过程，通过 $O(\log n)$ 次上面的卷积合并即可得到最终的答案。

设 $n = 1$ 时 x^i 的期望是 f_i，则有：

$$\begin{aligned}
f_k &= \sum_{i\ge 1} i^k q^{i-1} \\
q f_k &= \sum_{i\ge 1} i^k q^i \\
(1-q) f_k &= 1^k + q \sum_{i\ge 1} \left((i+1)^k - i^k\right) q^{i-1} \quad \text{（错位相减）} \\
(1-q) f_k &= 1 + q \sum_{i\ge 1} q^{i-1} \sum_{j=0}^{k-1} \binom{k}{j} j^k \\
(1-q) f_k &= 1 + q \sum_{j=0}^{k-1} \binom{k}{j} f_j \\
f_k &= 1 + q \sum_{j=0}^{k} \binom{k}{j} f_j \\
\frac{f_k}{k!} &= \frac{1}{k!} + q \sum_{j=0}^{k} \frac{f_j}{j!} \times \frac{1}{(k-j)!} \\
\text{Let } F(x) &= \sum_{i\ge 0} \frac{f_i}{i!} x^i \\
\text{Then } F(x) &= e^x + q F(x) e^x \\
F(x) &= \frac{e^x}{1 - q e^x}
\end{aligned}$$

用多项式求逆求出指数生成函数即可。后面再用倍增得到答案，复杂度 $O(m \log m \log n)$。

其实最开始提到的合并就是 x 和 y 的指数生成函数直接相乘，因此要得到最终的答案实际上只需要快速幂。如果用多项式 exp 做快速幂的话也可以把理论复杂度优化成一个 log，不过因为常数问题没什么提升。

做法 2　由于 n 不大，可以考虑直接从最终答案入手。

$$\text{ans}_k = \sum_{i\ge n}\binom{i-1}{n-1}p^n q^{i-n} i^k$$

显然对任意的常数 C，$\binom{x+C}{n-1}$ 都可以被表示为一个不超过 $n-1$ 次的关于 x 的多项式，因此不妨令

$$A(x) = \sum_{i=0}^{n-1} a_i x^i = \binom{x+n-1}{n-1}$$

那么就有

$$\begin{aligned}\text{ans}_k &= p^n \sum_{i\ge 0} A(i) q^i (i+n)^k \\ &= p^n \sum_{t=0}^{k}\binom{k}{t} n^{k-t} \sum_{i\ge 0} A(i) q^i i^t \\ &= p^n \sum_{t=0}^{k}\binom{k}{t} n^{k-t} \sum_{j=0}^{n-1} a_j \sum_{i\ge 0} q^i i^{j+t}\end{aligned}$$

令

$$f_k = \sum_{i\ge 0} q^i i^k$$

经过和做法 1 差不多的推导可以得到指数生成函数

$$F(x) = \frac{1}{1-qe^x}$$

继续考虑如何计算答案：

$$\text{ans}_k = p^n \sum_{t=0}^{k}\binom{k}{t} n^{k-t} \sum_{j=0}^{n-1} a_j f_{j+t}$$

其中

$$\sum_{j=0}^{n-1} a_j f_{j+t}$$

可以将 a 翻转后用一次 FFT 解决，记结果为 c_t，则有

$$\text{ans}_k = p^n \sum_{i=0}^{k}\binom{k}{i} n^{k-i} c_i$$

再用一次 FFT 求出即可。

回过头来考虑如何计算

$$A(x)=\sum_{i=0}^{n-1}a_ix^i=\binom{x+n-1}{n-1}$$

考虑到

$$\binom{x+n-1}{n-1}=\frac{1}{(n-1)!}\prod_{i=1}^{n-1}(x+i)$$

后面的乘积可以直接用分治 FFT 求，也可以用一个倍增的 FFT，后者的复杂度是 $O(n\log n)$ 的。

总的复杂度是 $O(n\log n+m\log m)$（如果用倍增求 a_i）或者 $O(n\log^2 n+m\log m)$（如果用分治）。

实际上上面的推导过程也是为了借用 $\sum q^i i^k$ 可以用多项式求逆求出的结论，其他部分只是在将这个结果转化到答案。

E. Draw a Triangle

作者 desprado2 **一血** 清华大学 写得不好，草草睡辽

题解 假设三个点分别为 A,B,C，其中 A 和 B 坐标已知。则面积可以用叉积表示：$S=\frac{1}{2}|\overrightarrow{AB}\times\overrightarrow{AC}|$。

设已知的 $\overrightarrow{AB}=(x,y)$, 未知的 $\overrightarrow{AC}=(u,v)$，则 $S=\frac{1}{2}|xv-yu|$。不难发现绝对值符号内是一个经典的 exgcd 的式子，其能取得的最小值为 $\gcd(x,y)$。直接套用 exgcd 求出 $xv-yu=\gcd(x,y)$ 的任意一组解即可。

注意对符号和 0 的情况稍作讨论。

F. Union of Circular Sectors

作者 desprado2 **一血** N/A

题解 可以发现，最后会形成若干连通块，而每个连通块外轮廓一定是由若干条圆弧和线段围成的。

对于这类图形求面积，一个经典方法是使用**格林公式**，令 $Q = x$, $P = -y$：

$$S = \iint_D dxdy = \frac{1}{2}\iint_D (\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y})dxdy = \frac{1}{2}\oint_L xdy - ydx$$

其中 D 表示该连通块的面，而 L 表示逆时针围绕该面的边缘曲线。

对于一条线段 $(x_1, y_1) \rightarrow (x_2, y_2)$，积分的结果是 $x_1y_2 - x_2y_1$。

对于一条弧线，假设其参数方程为：

$$\begin{cases} x = x_0 + r\cos\theta \\ y = y_0 + r\sin\theta \end{cases} \quad \theta = \alpha \rightarrow \beta$$

则有：

$$\begin{aligned} S = \frac{1}{2}\oint_L xdy - ydx &= \frac{1}{2}\int_\alpha^\beta (x_0 + r\cos\theta)d(y_0 + r\sin\theta) - (y_0 + r\sin\theta)d(x_0 + r\cos\theta) \\ &= \frac{1}{2}\int_\alpha^\beta r\cos\theta(x_0 + r\cos\theta)d\theta + r\sin\theta(y_0 + r\sin\theta)d\theta \\ &= \frac{1}{2}r(x_0\sin\theta + y_0\cos\theta + r\theta)|_\alpha^\beta \end{aligned}$$

题意保证了外轮廓线的每一段都唯一属于某个扇形。因此，可以枚举每个扇形 i，再枚举其他的扇形 j，计算出扇形 j "覆盖" 了扇形 i 外轮廓的哪些部分。对覆盖部分求并（方法类似于区间求并，排序后打标记），剩下的部分就是扇形 i 被 "暴露" 在外面成为外轮廓的部分。对每一段按照上面的式子求出积分，将积分值求和即为答案。

总时间复杂度 $O(n^2 \log n)$。

另外，不要使用辛普森积分来求与圆有关的面积问题。本题在造数据时着重卡了辛普森积分。

G. Group Homework

作者 fstqwq　**一血** 清华大学 写得不好，草草睡辽

题解　考虑两条链的关系，我们可以证明至多交于一点。若交于两点以上，我们可以修改为更大的答案。

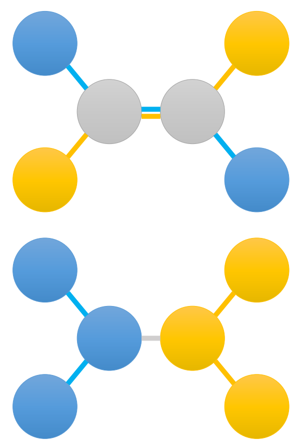

图 2: 两条交大于 1 的路径不如两条独立的路径。

因此要么是两条独立的路径，要么是交于一点。对于前者，我们可以考虑拆开一条树边两边用换根 DP 求带权直径；对于后者，我们可以枚举交点，用换根 DP 维护到叶子最长链，前四大加起来即可。时间复杂度 $O(n)$；考虑到时限充裕，为了实现方便可以直接 sort，为 $O(n \log n)$。

对于有交的情况，亦可以忽略掉至多交于一点的限制，使用状态压缩的方法直接维护出所有可能的答案，当然实现起来会非常麻烦。

H. Hysteretic Racing

作者 czjxyz　**一血** 北京大学 逆十字

题解　将环拉成一个无限长的序列后，询问操作即是询问满足

$$\sum_{i=s}^{r} A_i \cdot \max_{\mathrm{j=l}}^{\mathrm{i}}\{A_j\} > t$$

的最小的 r。不妨令

$$S_1(l,r) = \sum_{i=l}^{r} A_i \cdot \max_{\mathrm{j=l}}^{\mathrm{i}}\{A_j\}$$
$$S_2(l,r) = \sum_{i=l}^{r} \max_{\mathrm{j=l}}^{\mathrm{i}}\{A_j\}$$
$$S_3(l,r) = \sum_{i=l}^{r} A_i$$

考虑使用线段树在每一个节点 x 的区间 $[l_x, r_x]$ 维护这三个值；还要额外维护最大值 mx。

令 $f(v,x)$ 表示在线段树节点 x 里，传入前缀最大值 v 时 S_1 的值为多少，即：

$$f(v,x) = \sum_{i=l_x}^{r_x} A_i \cdot \max(v, \max_{\mathrm{j=l}}^{\mathrm{i}}\{A_j\})$$

$f(v,x)$ 可以在 x 的子树内二分求得，即：

- 若 $\mathrm{mx}_{\mathrm{lson}} \le v$，则左子树的最大值将被 v 覆盖，即 $f(v,x) = v \cdot S_3(l_{\mathrm{lson}}, r_{\mathrm{lson}}) + f(v, \mathrm{rson})$；
- 若 $\mathrm{mx}_{\mathrm{lson}} > v$，则 v 的影响不会波及右子树，我们可以用当前点的 S_1 减去左子树的 S_1 来得到在左子树影响下右子树的 S_1 值，即 $f(v,x) = f(v, \mathrm{lson}) + S_1(l_x, r_x) - S_1(l_{\mathrm{lson}}, r_{\mathrm{lson}})$。

$S_1(l_x, r_x) = S_1(l_{\mathrm{lson}}, r_{\mathrm{lson}}) + f(\mathrm{mx}_{\mathrm{lson}}, \mathrm{rson})$，故我们可以利用 f 来进行线段树的 push up，单次复杂度为 $O(\log n)$。S_2 的维护同理。

当区间整体加上 v 时 $S_1'(l,r) = S_1(l,r) + v(S_3(l,r) + S_2(l,r)) + v^2(r-l+1)$，$S_2'(l,r) = S_2(l,r) + v(r-l+1)$，区间加单次复杂度为 $O(\log^2 n)$。

当区间修改为 v 时 $S_1'(l,r) = v^2(r-l+1)$，$S_2'(l,r) = v(r-l+1)$，区间赋值单次复杂度为 $O(\log^2 n)$。

询问操作是利用 f 在线段树上进行二分，而走过一整个环之后最大值就不变了，故询问最多进行 3 次线段树二分，询问单次复杂度为 $O(\log^2 n)$。

总时间复杂度为 $O(n \log n + q \log^2 n)$。

I. Invincible Hotwheels

作者 desprado2, AntiLeaf　**一血** N/A

题解　我们称三元组 (i, j, k) 中的三个串分别为 *最短串*，*中间串*，*最长串*。

枚举三元组中的*最长串* k，枚举 k 中的每一个前缀 $s_k[1..len]$，则有一些其他字符串满足其为这个前缀的后缀；将满足条件的字符串中的最长者和次长者放入集合 S 中（下文将这二者简称为某个前缀的"最长后缀"和"次长后缀"）。不难发现，最后有可能对答案产生贡献的最短串 i，必然满足 $i \in S$。

为了快速找到一个前缀的最长后缀和次长后缀，可以对所有串建立 AC 自动机 Σ。对于串 k 的某个前缀，我们可以找到其在 Σ 中的对应节点 u，则不难发现最长后缀和次长后缀，是从节点 u 跳 fail 链接过程中遇到的第一个和第二个结束符对应的字符串。这一步在建 fail 树后一边 dfs 预处理即可。

现在我们可以观察集合 S 中的某个串可以作为*最短串* i 对答案产生贡献的条件。需要注意的是，集合 S 中的每一个元素可能在 s_k 中出现多次，且部分出现位置可能并不是对应前缀的前二长后缀。我们不妨把 s_k 中每个前缀对应的前二长后缀各视作一个区间，并为其染上对应的颜色。**以题目中样例三为例子**，字符串 cbcb 长为 3 的前缀的最长后缀为 cbc，且其对应于第四个字符串，则其可以被视作区间 $[1, 3]$ (1-based)，并为其染上颜色 4；其次长后缀为 bc，且其对应于第一个字符串，则其可以被视作区间 $[2, 3]$，并为其染上颜色 1。那么集合 S 中某个串可以作为*最短串* i 的条件是：

1. 对于所有该串对应的区间（因为出现了多次所以可能有多个），至少有一个区间包含了它们中的某一个；且所有包含过它们的区间，颜色相同。

2. 不存在一个前缀 $s_k[1..len]$，使得该串是这个前缀的后缀，但是该串的长度小于其*次长后缀*的长度。

对于第一条，可以将问题转化为一个二维数点问题：包含区间 $[l, r]$ 的区间 $[L, R]$ 满足 $L \le L$, $R \ge r$。因此，离线后用树状数组来维护这一信息即可。

对于第二条，在 AC 自动机的 fail 树上，一旦找到了次长后缀，则将次长后缀对应节点的父节点到根的路径全部打上一个"非法"标记，表示该路径上的所有串均不能作为*最短串*。该过程同样可以用树状数组维护，另外可以用 set 等其他数据结构来维护。

本题一个有趣的地方是，虽然需要统计的是三元组的数量，但是实际上只有 $O(n)$ 个合法答案，不需要开 long long。

假设串长总和为 m，则总时间复杂度为 $O(m \log m)$。

此外，使用广义 SAM 来替代 AC 自动机，同样可以求出答案。

J. Permutation Puzzle

作者 relyt871　**一血** 北京理工大学 *ddl* 战神

题解　根据所给限制建图，将 $p_u < p_v$ 的限制视为一条 $u \to v$ 的单向边，题目保证会得到一个 DAG。

设在 DAG 上存在一条从 u 到 v 的边数为 k 的路径，若 p_u 已知而 p_v 未知，则可以推出 $p_v \geq p_u + k$；若 p_u 未知而 p_v 已知，则可以推出 $p_u \leq p_v - k$。

首先我们通过上述规则推导出所有位置的取值范围 $[L_i, R_i]$。对于已知未知，显然 $L_i = R_i = p_i$；对于未知位置，可以用拓扑排序 + dp 来优化。先正向拓扑排序求出 L：对于边 (u, v)，做转移 $L_v = \max(L_v, L_u + 1)$；然后反向拓扑排序求出 R：对于边 (u, v)，做转移 $R_u = \min(R_u, R_v - 1)$。

于是转化为如下问题：给定 k 个区间和 k 个互不相同的数，我们需要给每个数匹配一个包含它的区间，此外每个区间匹配的数还要满足一些拓扑关系。如果暂时不考虑拓扑关系的话是一个经典问题，存在一个简单的贪心做法：从小到大枚举所有数，当枚举到 x 时，从所有左端点 $\leq x$ 且还没被匹配的区间中，选择右端点最小的那个匹配给 x，这个过程用优先队列优化。

然后分析一下区间的性质：如果存在边 (u, v)，根据转移方程可得 $L_u + 1 \leq L_v$，$R_u + 1 \leq R_v$，按照上述贪心做法，$[L_u, R_u]$ 一定比 $[L_v, R_v]$ 更早被匹配到，即一定满足 $p_u < p_v$。所以直接贪心求出来的就是原问题的合法解，如果贪心无解则原问题一定无解。

总时间复杂度 $O(m + n \log n)$。

K. Barrel Theory

作者 Timsei,desprado2 **一血** 北京大学 逆十字

参考题解 1 - Timsei

- 首先处理一些简单的情况，$n = 1$ 和 $n = 2$ 可以通过简单讨论解决。
- 我们构造的思路是试图对于 m 为偶数的情况，使得异或值为 0。对于 m 为奇数的情况，使得异或值为 1。这是我们对于这样的 m 能构造的可能能最小的异或值，因为 m 为奇数时，最终异或值也一定是奇数。
- 对于 m 为偶数的情况：

如果 n 为偶数 我们构造 $n - 2$ 个 1，和两个 $(m - n + 2)/2$. 这是最简单的情况。

如果 n 为奇数 我们可以先填 $n - 3$ 个 1，观察最后 $n - 3$ 个数的和 k.

  - 如果和的二进制位中至少有两个 bit, 那么我们可以构造 $k/2, k/2 - \text{lowbit}(k), \text{lowbit}(k/2)$.
  - k 是 2^p 的形式。$k = 4$ 那么无解。
    * 如果 $n \geq 5$ 那么我们可以将前面的任意两个 1 变成两个 2，这样就转化成至少有两个 bit 的情况。
    * 如果 $n = 3$
      · $k = 8$ 则无解
      · $k \geq 16$ 则构造 $3k/16, 6k/16, 7k/16$

- 对于 m 为奇数的情况, 我们异或的最小值为 2，所以当 $m < 2n + 1$ 时无解。我们沿用类似 m 为偶数时的方法讨论。
  - 如果 n 为偶数 ($n = 2$ 已经讨论)，那么我们先放 $n - 2$ 个 2，剩下的两个数平均的放，然后把前面的一个 2 改成 3 即可。
  - 如果 n 为奇数
    * 我们首先放 $n - 3$ 个 2 , 假设剩下的数的和为 k。我们观察在沿用上面的方法的时候，何时会出现问题。
      · k 不为 $2^p + 1$ 或者 $2^p + 3$ 的形式, 此时，我们可以使用上面的方法，构造 $k/2, k/2 - \text{lowbit}(k/2), \text{lowbit}(k/2)$。然后在其中的偶数上加一个 1，因为后两个数中一定存在一个偶数。
      · $k = 2^p + 1$ 当 $k \leq 9$ 无解。

如果 $n = 3$ 那么 $k \leq 17$ 时无解, 否则我们构造 $3*(k-1)/16, 3*(k-1)/16*6+1, (k-1)/16*7$

否则 我们将前面的两个 2 改为两个 3，就可以按照上面的方法处理

· $k = 2^p + 3$, 当 $k \le 11$ 无解

如果 $n = 3$ 那么 $k \le 19$ 时无解，否则我们构造 $7, (m-7)/2-1, (m-7)/2+1$

否则 我们将前面的两个 2 改为两个 4，就可以按照上面的方法处理

参考题解 2 - desprado2 直接对奇偶性进行分类讨论。讨论方法如下：

- 首先排除掉一些必定无解的情况。首先 $n = 1$ 必定无解。若 m 为奇数，则异或和至少为 1，因此每个数都至少为 2。则当 $m < 2n$ 时必然无解；若 m 为偶数，则 $m \le n+1$ 时必然无解。
- 若 n 和 m 均为偶数，则可以先放 n 个 1，然后将其中两个 1 加上 $(m-n)/2$。
- 否则，先讨论掉 n 较小的几种情况：
  - 若 $n = 2$，则无解的情况当且仅当 m 的形式为 $2^k - 1$。否则直接构造答案 $[\lfloor \frac{m}{2} \rfloor, \lceil \frac{m}{2} \rceil]$
  - 若 $n = 3$，这是本题中最难讨论的情况：
    * 若 m 为偶数，则 $m = 4, 8$ 时无解，其他情况下：
      · m 是 2^k 的形式，即 $\text{popcount}(m) = 1$。此时直接构造答案 $[3,\ \lfloor \frac{m-3}{2} \rfloor, \lceil \frac{m-3}{2} \rceil]$
      · 否则，$\text{popcount}(m) \ge 2$，则 $m/2$ 可以被分为两个数 $\text{lowbit}(m/2)$ 和 $m/2 - \text{lowbit}(m/2)$。因此，直接构造答案 $[m/2,\ \text{lowbit}(m/2), m/2 - \text{lowbit}(m/2)]$
    * 若 m 为奇数，则 $m = 3, 5, 7, 9, 11, 17, 19$ 时无解。其余情况下，对 $m \bmod 8$ 和 $m \bmod 16$ 进行讨论：
      · $m \bmod 8 = 5\ or\ 7$，则意味着三个数的最后两个二进制位可以构造成 $[00,\ 10,\ 11]$ 或者 $[01,\ 11\,11]$ 的形式。因此，直接构造答案 $[(m-5)/2,\ (m-5)/2+2,\ 3]$ 或者 $[(m-7)/2+1,\ (m-7)/2+3,\ 3]$
      · 否则，对 $m \bmod 16$ 做讨论，此时构造最后三个二进制位。假设用 (t, a, b, c) 来表示最后的答案为 $[(m-t)/2+a,\ (m-t)/2+b,\ c]$，则不同的 $m \bmod 16$ 的答案如下：

        9: $(9, 0, 4, 5)$

        11: $(11, 1, 5, 5)$

        1: $(17, 3, 7, 7)$

        3: $(19, 5, 7, 7)$
  - 若 $n = 4$，则只有 m 为奇数尚未被讨论。有解条件是 $m \ge 9$ ，构造方法为 $[2,\ 3,\ (m-5)/2,\ (m-5)/2]$
  - 若 $n = 5$：
    * m 为偶数，有解条件是 $m > 6$，构造方法为 $[1,\ 2,\ 3,\ (m-6)/2,\ (m-6)/2]$

    * m 为奇数，有解条件是 $m > 15$，构造方法为 $[2, 4, 7, (m-13)/2, (m-13)/2]$

- 讨论掉 $n \le 5$ 的情况后，所有 $n > 5$ 的情况均可以转化为 $n = 4\ or\ 5$ 的情况：
  - n 奇 m 偶，则可以放 $n-5$ 个 1 后转化为 $n = 5$ 且 m 为偶数的情况。有解条件是显然的：$m > n+1$。
  - n 偶 m 奇，则可以放 $n-4$ 个 2 后转化为 $n = 4$ 且 m 为奇数的情况。有解条件也是显然的：$m \ge 2n+1$。
  - n 奇 m 奇。首先证明有解条件是 $m > 2n+5$：若 $m \le 2n+5$ 且有解，则由于每个数都至少为 2，所以解里面只有至多 $m-2n = 5$ 个数大于 2，即至少有 $n-5$ 个数等于 2。由于 $n-5$ 为偶数，所以剔除 $n-5$ 个 2 以后，问题就转化为将 $m-2(n-5)$ 分给 5 个数的问题。然而，当 $n = 5, m \le 15$ 且为奇数时，问题是无解的，因此出现矛盾。当 $m > 2n+5$ 时，构造方法时先放 $n-5$ 个 2，然后问题转化为 $n = 5$ 且 m 为奇数的情况。

备注　讨论比较繁琐，但是思路其实比较自然。解法 1 是通过讨论二进制的最后几位处理特殊情况来解决问题，而解法 2 则是首先讨论几个 n 较小的情况，之后推理整体做法，这种做法可以由打表找规律的方式帮助解决。

此外，本题允许 $O(m)$ 的方法通过，所以一些带有枚举的做法也是可以通过的。尤其是最复杂的 $n = 3$ 的情况，事实上打表可得最小数小于等于 7 即有解（该结论可以在上面的讨论中被证明），因此可以用 $O(7m)$ 的两重循环暴力枚举。

L. Largest Unique Wins

作者 fstqwq　**一血** 南京大学 明明是我先来的

题解　本题的难点在于发现这是一道构造题：存在一种非对称纳什均衡，其中每个玩家的策略是一个纯策略（非随机策略）。

我们给出这个题可能的观察方法：

- 当 $n = 2m$ 时，如果每种恰好有 2 个玩家选择，任意玩家不变的收益是 0，而变的收益是 -1。
- 当 $n > 2m$ 时，前 $2m$ 名玩家以上述方法构造，之后的玩家无法造成影响，任意选择即可。
- 当 $n < 2m$ 时，如果所有玩家都是纯策略，那么直观来说所有玩家都想选大的，除非大的已经被选了两次。因此，我们可以考虑玩家一个一个一个来，此时：
  - 第 1 名玩家：m
  - 第 2 名玩家：m
  - 第 3 名玩家：$m-1$
  - 第 4 名玩家：$m-1$
  - 第 5 名玩家：$m-2$
  - ...

  我们发现按照这个构造方法构造的任意 m 均是可行的，因此这就是本题的参考解法。

出题组已知的范围内，没有非常好的计算对称纳什均衡的方法。此题数据范围很小，是因为 checker 使用了 $O(nm \cdot 3^m)$ 的状压 DP。如果对于猜测题目做法带来了困扰，我们深表歉意。

M. Youth Finale

作者 fstqwq　**一血** 中国科学院大学 果壳家的龙 *ACMer*

题解　首先我们需要数一下逆序对，使用归并排序或者树状数组即可。

随后，我们可以发现：当进行 Shift 操作时，对逆序对的影响只和这个数是谁有关系：将 i 从头移动到尾时，有 $i-1$ 个 (i,j) s.t. $j < i$ 关系减少了，而有 $n-i$ 个 (i,j) s.t. $j > i$ 增加了。因此我们维护需要 Shift 谁即可完成操作。Reverse 操作会改变 Shift 的方向，同时将 #inv 改为 $\binom{n}{2} - \#\text{inv}$，因为一共有 $\binom{n}{2}$ 个关系，原先所有逆序对变为顺序对，顺序对变为逆序对。

因此，仅需数一次逆序对，回答询问是 $O(1)$ 的，总复杂度 $O(n \log n)$。当然，时限宽裕，直接 $O(\log n)$ 也是应当可以通过的。

尾声

热身赛 B. Underdetermined

作者　fstqwq

题目　给一个 01 矩阵，问是否能修改一个子集的 1 为 0，使得原矩阵行列式不为 0。$n \le 100$。

题解　充要条件是存在一个排列乘积非 0，也即存在行与列的完备匹配。方案即保留匹配中的边。

热身赛 C. Reversing

作者　Timsei

题目 & 题解　在本场比赛的验题赛结束的那天晚上，desprado2 刷知乎看到了这个：

一道小清新的后缀数组题目 - 严格鸽的文章 - 知乎 `https://zhuanlan.zhihu.com/p/576417417`

本题有 114514 种做法：Lyndon 分解，SA，SAM，等等。挺好一个题，怎么就被人抢先出了呢。当然，特别感谢严格鸽老师在知乎上分享了做法，避免了惊天巨锅。

G. Group Homework

本题实现需要注意正确实现换根 DP。一个常错的细节是：在求两条独立的链的时候，第二遍换根 DFS 需要从其他儿子那里传递最长直径的答案。出题人、验题人，以及逆十字这个地方都写挂了。特别感谢郭吉舟同学：如果没有和他聊题目时想到这个地方，数据就寄了。

K. Barrel Theory

原本要求是异或小于等于最小值，是由一道读错的 CF (Round 819B) 题改编的。但是，验题时发现很多人也读错过，所以加强到了小于。在这里，特别感谢浙江大学 *Solitary Dream* 队指出这一点。

H. Hysteretic Racing

`text2img` 生成的 prompt 是：`best quality, 4K, anime, sloth in a car, sloth driver`，Seed `2174826620`，Model Hash `925997e9`，大约能生成下图，然后再 inpaint 改改表情和细节就好了。