

A

由于所有字符串长度都相等，所以对于循环同构的偏移量 d ，如果 $d \geq n$ ，那么必然可以对某个字符串序列进行重排找到一种满足 $d < n$ 的方式来满足同样的循环同构。

所以只需要讨论 $d < n$ 的情况，固定 A 中的字符串，观察到此时 B 中的字符串相当于拆成一个长度为 d 的前缀和一个长度为 $n - d$ 后缀，同时这个长度为 d 的前缀会与 A 中对应位置的字符串一个长度为 d 的后缀配对，长度为 $n - d$ 的后缀会与 A 中下一个位置的字符串的一个长度为 $n - d$ 的前缀配对。

所以对于一个 d ，只需要将 A, B 中所有的字符串按照上述方式拆分成一个前缀和一个后缀，拆分后相同的字符串代表同一个点，可以用哈希来判断字符串是否相同。

对于一个长度为 d 的字符串，它会和 B 中一个长度为 $n - d$ 的后缀配对，同时还要和 A 中一个长度为 $n - d$ 的前缀配对。同样的，一个长度为 $n - d$ 的字符串也需要和两个长度为 d 的字符串配对。

将配对方式抽象成边，所有的不同前后缀抽象成点，那么就会得到一张包括少于 $2m$ 个点和恰好 $2m$ 条边的图。一个循环同构可以看作一条不重复的经过这 $2n$ 条边的路径，即欧拉回路。

所以枚举偏移量 d ，每次判断是否存在欧拉回路的复杂度为 $O(m)$ ，复杂度 $O(nm)$ 。

B

后面的操作会覆盖掉前面的操作，所以倒着做这些操作，就可以保证一个位置的值确定后不会更改。

- 全为 2 的操作肯定要先做，全为 1 的操作肯定要最后做，问题只剩下考虑其中一个是 1 一个是 2 的操作的操作顺序。
- 在第一步之后，考虑存在的一个被固定为 2 的位置 p ，且有一个操作形如将 p 位置的数改为 1，同时将 q 位置的数改为 2，那么这个操作也是可以先做的，因为它确定了 q 位置的数在最后为 2。
- 第二步操作完之后，场上已经不存在上述情形，必须要确定一个位置的值为 1，那么对于剩余的所有形如把 u_i 改为 1，把 v_i 改为 2 的操作，如果将每一个位置看作一个点，连有向边 $u_i \rightarrow v_i$ ，那么在让 $u_i = 1$ 之后所有 u_i 可以直接达到的位置都可以被锁定为 2，操作顺序按照dfs序即可。

在连边后，考虑每一个连通块，对其进行强连通分量的缩点之后可以得到一张有向无环图。只需要在入度为 0 的每一个强连通分量中选取一个点就可以将这个连通块上其余的点全部锁定为 2。显然这是最优的情况，因为这些强连通分量的本身就只能通过其中的某一个点来将其他的点锁定为 2。

第二步本身可以合并入第三步来一起操作，只需要在选择强连通分量中的点时优先选择被 $(l_i, 2, r_i, 2)$ 的操作固定为 2 的点即可，最后按照dfs序排列好顺序输出方案即可。

C

考虑利用题目中给出的限制条件，设 s_i 代表 $[1, i]$ 中的左括号数量，那么有

- $s_i - s_{i-1} = 0 \text{ 或 } 1$
- $s_i \geq i - s_i$
- $s_n = \frac{n}{2}$
- 对于每个限制条件，有 $s_r - s_{l-1} - (r - l + 1 - s_r + s_{l-1}) = c$

利用上述条件构建差分约束系统，有

差分约束系统有解 \iff 原括号序列存在合法解

本题数据较弱，直接差分约束即可，不过极端情况下直接差分约束可能会被卡成 $O(nq)$ 的复杂度。

考虑将一个限制条件看作 $l-1$ 和 r 之间连边，那么在处理完一些限制条件后会得到一些连通块。对于每个连通块而言，一个新的限制条件如果连接了这个连通块内的两个点，那么这个条件在合法的情况下是没有意义的。

利用带权并查集判断连通性以及合法性，就可以将有效限制条件的个数限制在 $O(n)$ 个，运行差分约束求解的复杂度为 $O(n^2)$ 。

D

先不花钱，那么最终只有两种可能的情况，就是两条路径交汇或者不交汇。

不交汇的情况非常简单，就是分别求出最短路径长度。

交汇的情况可以发现必然有一种最优情况是两人路径的公共部分只有一段，于是枚举这一段的起点 s 和终点 t ，然后第一个人的路径就是 $s_1 \rightarrow s \rightarrow t \rightarrow t_1$ ，第二个人的路径就是 $s_2 \rightarrow s \rightarrow t \rightarrow t_2$ 或者 $s_2 \rightarrow t \rightarrow s \rightarrow t_2$ 。

对于相同长度的重合路径，非重合部分的路径长度肯定越短越好，记长度为 i 的重合路径对应的非重合部分的最短路径长度为 $minn_i$ 。

上述部分要求预处理出全源最短路，利用01bfs即可在 $O(n^2 + nm)$ 的时间复杂度内求出。

再考虑花钱使路径长度减小的情况，对于重合的路径，花 1 元可以减小的总时间依次为 $1, \frac{1}{3}, \frac{1}{6}, \frac{1}{10}, \dots$ ，对于非重合的路径，花 1 元可以减小的总时间依次为 $\frac{1}{2}, \frac{1}{6}, \frac{1}{12}, \dots$ ，容易发现每次花钱肯定是贪心选择减小总时间最多的一条路。

对于所有重合的路径和非重合的路径，其改造的次数分别满足最大值减去最小值不超过 1。

对于每一个 $(i, minn_i)$ 的组合，可以二分重合的路径改造的次数的最小值，进而求得非重合的路径改造的次数的最小值，从而在 $O(\log k)$ 的时间复杂度内求得每条路径上应该花费的钱，再求得答案。

上一步也可以通过三分或者其他方式实现，时间非常充裕，只要是非暴力的方法一般都可以通过此题。

E

将最后的AND和记为 S ， $S > V$ 意味着 S, V 的二进制位存在一个相同的前缀，且下一位满足 $V_i = 0, S_i = 1$ 。

枚举这个前缀，最多 $O(\log n)$ 个，这样就会确定了 S 的某些位置必须为 1，其它位置可以任意，根据AND和的性质可以确定路径上的所有边在这些位置上也必须是 1，将所有合法的边提取出来建一张新图，问题来到了判断 x, y 是否联通，可以用并查集实现。

由于 V 提前给定，这 $\log n$ 个新图也可以提前建好，然后对于每组 (x, y) 进行 $O(\log n)$ 的判断即可。