[AGC057C] Increment or Xor

> 给定正整数 N，以及一个长度为 2^N 的序列，$A = A_0, A_1, \cdots, A_{2^N-1}$。满足每个数都是 $[0, 2^N - 1]$ 里的整数，且**互不相同**。每次你可以对序列进行两种操作：
>
> - 操作一：**每个数**加一，再对 2^N 取模。
> - 操作二：选取一个 $[0, 2^N - 1]$ 中的整数 x，**每个数**异或上 x。
>
> 最终要使得 $\forall i, A_i = i$。输出方案或报告无解。可以证明，若有解，一定能在 10^6 次操作内实现。可以给出任意合法方案，但你给出的方案不应超过 10^6 次操作。若有解，输出中有一行整数依次描述你的每次操作：输出 -1 表示该次你选择操作一；输出 $[0, 2^N - 1]$ 中的一个整数 x 表示你选择用它进行操作二。

首先这题的做法非常的构造性。首先我们考虑 SPJ 怎么写的，这肯定是个 trie，bit reverse 后：

- 异或相当于把某一层的全部儿子翻转。
- +1 相当于把最右边的一整条链翻转，事实上也可以做到翻转任一叶子节点上方的所有结点。

对于每个点，记录它被翻转了几次和目标翻转几次，我们从下向上递归构造就好。

【北大集训2021】末日魔法少女计划

> 对于给定的 n, k，你需要构造一个只含 $0, 1$ 的矩阵 $A_{i,j}$，$0 \le i, j \le n$，满足：
>
> 1. $A_{i,i} = 1$。
> 2. $A_{i,i+1} = 1$。
> 3. 对 $i > j$ 有 $A_{i,j} = 0$。
> 4. 若 $A_{i,j} = 1, j - i > 1$，则存在 $i < t < j$，满足 $A_{i,t} = A_{t,j} = 1$。
> 5. 对 $i \le j$ 有 $(A^k)_{i,j} > 0$。
>
> 你需要输出满足 $A_{i,j} = 1$ 且 $j - i > 1$ 的每个 (i, j)，设这样的 (i, j) 共有 m 个。
>
> 若输出不满足要求，则不能得到该测试点的任何分数。若输出满足要求，则根据 m 进行评分。

首先这是一道理论数据结构题。

我们设 $f_{x,i}$ 表示 x 个数的区间，要求任一区间都得在 i 步以内达成，那么我们枚举它分成了 m 段，中间的 $m-2$ 段都需要前缀和和后缀和，第一段只要前缀和，最后一段只要后缀和；然后对于整段的和，可以用 $f_{m-2,i-2}$ 解决。

【北大集训2021】简单数据结构

> 小 D 是一位数据结构大师，他特别喜欢研究形式简单的数据结构，今天他想到了这样一道题目：
>
> 你有一个长度为 n 的序列 a，下面你要进行 q 次修改或询问。
>
> 1. 给定 v，将所有 a_i 变为 $\min(a_i, v)$。
> 2. 将所有 a_i 变为 $a_i + i$。
> 3. 给定 l, r，询问 $\sum_{i=l}^{r} a_i$。
>
> 顶级数据结构大师小 D 轻松的解决了这个问题，现在他打算来考考即将参加 IOI2022 的你，相信你也可以轻松解决这个问题。

考虑这个取 min 操作的影响怎么合并，设当前进行了 t 次操作二，v_i 表示 i 次操作二时最小的 v，令 $a'_i = a_i - it$，那么 $a'_i = \min_j\{v_j - ij\}$，也就是和一个下凸包取 min。每次给下凸包新增一条斜率递减的直线，然后找到所有和它相交的点，把它删掉。拿线段树套 Splay 维护凸包，来求凸包和某个斜率的切线和删点即可。

Everybody Lost Somebody

> "But there's nothing I wouldn't do to wake up and remember it."
>
> Jonathan is fan of string problems. He is learning lexicographic order and suffix array these days.
>
> String x is lexicographically less than string y, if either x is a prefix of y (and $x \neq y$), or there exists such i $(1 \le i \le \min(|x|,|y|))$, that $x_i < y_i$, and for any j $(1 \le j < i)$, $x_j = y_j$. Here $|a|$ denotes the length of the string a. The lexicographic comparison of strings is implemented by operator < in modern programming languages. For example, everybody is lexicographically smaller than somebody.
>
> Let $\mathrm{suf}\ i$ be $x_i x_{i+1} \cdots x_n$ for string x of length n. In suffix array problems, there are two commonly used arrays: sa of length n and $height$ of length $n-1$. Formally, sa_i $(1 \le i \le n)$ is the starting position of the i-th lexicographically smallest suffix $\mathrm{suf}\ j$, which means $sa_i = j$. And $height_i$ $(2 \le i \le n)$ is the length of longest common prefix between $\mathrm{suf}\ sa_{i-1}$ and $\mathrm{suf}\ sa_i$. For example, the sa and $height$ for remember is $\{6,4,2,7,5,3,8,1\}$ and $\{0,2,1,0,1,0,1\}$ respectively.
>
> As we all know, Little Y is a Riddler. One day, Little Y got a string S of length n consisting of only lowercase letters. He used suffix-array algorithms to get the array sa and $height$. He erased several numbers in $height$ and gave the two modified array to Jonathan.
>
> Curiously, Jonathan wants to know what the string S is. Please help him to figure out a possible answer. Since there may be multiple answers, you only need to print the lexicographically smallest one. It is guaranteed that the answer exists.

对于 $ht(a,c)$ 已知，那么就是连 $s_{a,b} = s_{c,d}$ 这样的区间相同的边，然后连一条小于号的边；如果 $ht(i,j) = -1$，那么比较 rk_{i+1}, rk_{j+1}，若 $rk_{i+1} < rk_{j+1}$，那就连小于等于的边；否则连小于的边。

为了优化连边，那么可以用分层并查集维护；也可以考虑其性质，对于 $ht > 0$，我们依然考虑 $ht(i+1, j+1)$，只需要连一条 $i - j$ 的边就够了；ht 从大到小扫一遍就好了。$O(n \log n) \sim O(n)$。

Upside Down

> 给出一棵 n 个节点的树，每一条边上有一个小写字母。给定 m 个字符串 s。
>
> 有 q 组询问，每组询问包含三个数：i, j, k，表示如下询问：
>
> 以从 i 到 j 的路径上每一条边的字母连接起来所构成的字符串为文本串，s_k 为模式串，求 s_k 在文本串中出现的次数。$n, q \le 10^5$。

考虑 $u - lca - v$ 这条链，$u - lca$ 的部分显然，$v - lca$ 的部分也显然；难在于跨过去的部分。

实际上，也就要求 $u - lca$ 的一个后缀，恰好是 s_k 的前缀；$lca - v$ 的一个前缀，恰好是 s_k 的后缀，然后拼起来是 s_k。假如求出 $u - lca$ 最长的一个后缀 W，那么 W 的所有 border 也就构成了 $u - lca$ 的后缀 $\cap$ s_k 的前缀。而 border 是 log 个等差数列，可以 $q \log^2$ 求。

抽象一下求 $|W|$ 的问题：给定 S 和 T，求最长的 S 的后缀恰好是 T 的前缀：

我们求出 $\max \operatorname{lcp}(S[i,|S|),T)$，那么 W 是这个 lcp 的一个 border。怎么求 lcp？可以把 S 后缀排序了，然后二分 T 在 S 中的排名就好了。

Almost Multiplication Table

> There are positive integers N, M, and an $N \times M$ positive integer matrix $A_{i,j}$. For two **(strictly) increasing** positive integer sequences $X = (X_1, \ldots, X_N)$ and $Y = (Y_1, \ldots, Y_M)$, we define the penalty $D(X, Y)$ as $\max_{1 \le i \le N, 1 \le j \le M} |X_i Y_j - A_{i,j}|$.
>
> Find two **(strictly) increasing** positive integer sequences X and Y such that $D(X, Y)$ is the smallest possible.

我们二分一个 D，设 $[L_{i,j}, R_{i,j}]$ 表示 $x_i y_j$ 必须处在的范围。一开始，令 $x_i = i, y_j = 10^9 + D + j$，也就是分别是 x 能取到的最小值，y 能取到的最大值，我们反复迭代直到找到答案：

$$\begin{aligned} y_j &\leftarrow \min\left(y_{j+1} - 1, \left\lfloor \frac{R_{i,j}}{x_i} \right\rfloor\right) \\ x_i &\leftarrow \min\left(x_{i-1} + 1 \left\lceil \frac{L_{i,j}}{y_j} \right\rceil\right) \end{aligned} \tag{1}$$

正确性显然，因为求的都是必要条件，而最后满足条件时就很充分！复杂度 $nm(n+m) \cdot V^{0.5} \cdot \log V$。

使用条件？二分图吧。