空

考虑模拟子序列匹配的过程，有一个显然的DP做法，设$F_{i,j}$表示当前已经考虑了T的前i位，匹配了S的前j位的方案数，转移的话讨论一下就好了。

从这里也可以发现答案跟S的内容无关。

考虑这个DP的过程，我们可以发现实际上是在n个位置中钦点m个位置来匹配，在匹配完之前的一个失配位置有$c-1$种选法，后面的每个位置有c种选法。

那么有$ans = \sum_{i=m}^{n} \binom{i-1}{m-1}(c-1)^{i-m} \cdot c^{n-i}$。暴力模拟这个式子就可以$\mathcal{O}(n)$了。

考虑进一步优化这个式子，$ans = c^{n-m}(\sum_{i=0}^{n-m} \binom{i+m-1}{m-1}(\frac{c-1}{c})^{i+m-1})$。令$p = \frac{c-1}{c}$，可以将$ans$写成$f(n,k) = \sum_{i=0}^{n} \binom{i+k}{k} p^i$的形式，$ans$即为$c^{n-m} \cdot f(n-m, m-1)$。

考虑从$f(n,0)$递推出$f(n,k)$，那么有

$$\begin{aligned} f(n,k) &= (\sum_{i=0}^{n} \binom{i+k-1}{k-1} p^i) + p(\sum_{i=0}^{n} \binom{i+k}{k} p^i) - \binom{n+k}{k} p^{n+1} \\ &= f(n,k-1) + p \cdot f(n,k) - \binom{n+k}{k} p^{n+1} \end{aligned}$$

即

$$f(n,k) = \frac{f(n,k-1) - \binom{n+k}{k} \cdot p^{n+1}}{1-p}。$$

可以在$\mathcal{O}(m)$的复杂度内解决。

银

翻转棋子不好处理，我们可以考虑另一个游戏：一开始在每个黑子的位置有一枚棋子，每个位置可以放置任意数目的棋子，每次操作变为取一个区间$[l,r]$且l处有棋子，从l处取走一枚棋子，再在$(l,r]$的所有位置各放一枚棋子。这样的两个游戏可以归纳证明是等价的，即初始状态对应的SG值相同，原因是一个位置如果有>1枚棋子，总是能从中取走两枚，且根据SG定理，局面的SG值不变。

现在只用考虑这个新游戏，计算初始局面的SG值只需要先算出每个黑子对应的SG值了，接下来我们默认从右往左。

可以归纳证明出$SG(i) = 2^{lowbit(i)}$，这样就可以$\mathcal{O}(nm)$了。

我们可以将SG值看做二进制数，做一个可逆的变换，每一位变成后面（大于等于它）的位的异或和，这个事实上定义了一个双射f。它显然满足一些性质，例如$f(0) = 0, f(a \bigoplus b) = f(a) \bigoplus f(b)$。因为是否必胜只跟$SG$是否为0有关，所以只用研究$f$了。那么$f(SG(i)) = 2^{lowbit(i)+1} - 1$，也就是$f(SG(n))$的第$i$位为1当且仅当$2^{i-1} | n$。

因此有$f(SG(1)) \bigoplus f(SG(2)) \bigoplus \ldots \bigoplus f(SG(n)) = n$。

于是考虑一个右端点r是否合法，也就是是否存在$l \le r$满足$f(SG(l)) \bigoplus f(SG(l+1)) \bigoplus \ldots \bigoplus f(SG(r)) = f(SG)$，即$f(SG) = r \bigoplus (l-1)$。那么$l \le r$当且仅当$f(SG)$的最高位（也是$SG$的最高位）在$r$的二进制表示中为1。

直接拿个数组维护每一位为1的黑子个数即可，复杂度$\mathcal{O}((n+m)\log n)$。

子

根据CRT，模不同质数的幂的乘积的结果显然等于模单个质数幂的结果的积，接下来只用考虑单个质数的幂。

对于奇质数p，考虑$x^k \mod p^a$有多少种不同的结果。先考虑$\gcd(x,p)=1$的，它们是整数模p^a乘法群的元素，它是一个阶为$\phi(p^a)=p^{a-1}\cdot(p-1)$的循环群，因此$x^k \mod p^a$有$\frac{\phi(p^a)}{\gcd(k,\phi(p^a))}$种可能。如果$gcd(x,p)>1$，那么设$x=p\cdot y$，$x^k=p^k\cdot y^k$，即$x^k \mod p^a=p^k\cdot(y^k \mod p^{a-k})$。于是可以将$a$减去$k$迭代计算。注意到$a$可能很大，需要加速迭代的过程。可以发现迭代时除了最后一项外，贡献是一个公比为p^k的等比级数，因此可以快速计算前面的项的和。

对于2^a也可以类似做。但是注意到当$a>2$时，整数模2^a乘法群的分解是一个阶为2^{a-2}的循环群与一个2阶群的直积，因此当k为奇数时$x^k \mod 2^a$显然取遍所有奇数，否则贡献为$\frac{2^{a-2}}{\gcd(k,2^{a-2})}$。迭代的过程同样可以用等比级数加速计算。

算gcd的时候需要分解$p-1$，因为限定了$p\leq 10^7$，可以线性预处理最小因子。不算预处理的复杂度为$\mathcal{O}((n+m)\log 10^9)$。

细节可能有些多，不清晰的地方详见标程。