《基础 01? 练习题》解题报告

浙江省衢州第二中学 郑钧

Oct 2024

目录

题目大意

记 $g(S)$ 表示 S 中有多少子串同时也是 Thue-Morse 序列 [1] 的子串。

记 $h(S)$ 表示将 S 中所有 `?` 替换为 `0` 或 `1` 后得到的串 S' 的 $g(S')$ 之和。

给定一个仅包含 `0,1,?` 的字符串 S,每次查询给出 l,r,求 $h(S_lS_{l+1}\cdots S_r) \bmod 998244353$。

数据范围

子任务	$n \le$	$m \le$	特殊性质	分值
1	15	15	A	10
2	20	2×10^5	无	10
3	5×10^4	2×10^5	A	5
4	5×10^4	1	BC	5
5	5×10^4	1	C	15
6	500	10^3	B	5
7	10^3	2×10^3	BC	5
8	5×10^3	10^5	C	10
9	2×10^4	10^5	无	15
10	5×10^4	2×10^5	无	20

特殊性质 A：$r-l+1\le 15$；

特殊性质 B：S 中 `?` 的个数不超过 8；

特殊性质 C：$l=0$。

时间限制：4s，空间限制：512MB。

解题过程

Subtask 1

观察到，判断一个串 S 是否是 Thue-Morse 序列的子串时，首次匹配成功的位置下标不会很大，事实上，这个上界为 $4 \times 2^{\lceil \log_2 |S| \rceil}$。

所以，只需要保留 Thue-Morse 序列的前 $O(|S|)$ 项，在此基础上按照题意模拟即可。

时间复杂度：$O(2^n \operatorname{poly}(n, m))$。

Subtask 2

在 Subtask 1 的基础上精细实现，记录每个串是否是 Thue-Morse 序列的子串。

记 $c(l, r)$ 表示将 $S_l S_{l+1} \cdots S_r$ 中的 `?` 替换成 `0` 或 `1` 得到 T，所有 T 的 $f(T)$ 之和。

这样，每一个 $c(l, r)$ 都可以通过枚举计算出来。

那么，在查询 $h(S_l S_{l+1} \cdots S_r)$ 时，只需要计算每一个 $c(l, r)$ 对答案的贡献。

具体地，记 $p(l, r) = \prod_{i=l}^{r} (1 + [S_i = \texttt{?}])$，那么 $c(l', r')$ 的贡献即为 $p(l, l' - 1) \times p(r' + 1, r)$。

时间复杂度：$O(2^n + n^2 m)$。

Subtask 3（特殊性质 A）

考虑预处理所有长度 ≤ 15 的 `01` 串的 f 值，并求出所有长度 ≤ 15 的包含 `0,1,?` 串的 h 值，可以通过 dp 快速转移。

时间复杂度：$O(15 \cdot 3^{15} + 15^2 m)$，精细实现可以做到 $O(3^{15} + 15^2 m)$。

正解

方便起见，约定：

- $a \oplus b$ 表示非负整数 a, b 按位异或的结果；
- $\mathrm{bit}_k(a)$ 表示非负整数 a 在二进制下第 k 位的值（从最低位开始为第 0 位）；
- $w(i)$ 表示 i 在二进制下 1 个数的奇偶性。
- 字符串的下标均从 0 开始，且字符串 S 的长度为 $|S|$；
- 记 $A + B$ 表示字符串 A 和字符串 B 拼接得到的字符串；
- 记 $S[l : r]$ 表示字符串 $S_l S_{l+1} \cdots S_r$；
- $\mathrm{rev}(S)$ 表示将 S 翻转得到的字符串；
- 记 Thue-Morse 序列形成的 `01` 字符串为 P，P 将 `01` 取反得到的字符串为 P'；
- 若非空字符串 S 是 P 或 P' 的前缀，则称 S 为 T.M. 串；
- 对于 `01` 字符串 S，我们称满足 $A + B = S$ 且 $\mathrm{rev}(A)$ 和 B 都是 T.M. 串的二元组 (A, B) 为 S 的一组 T.M. 拆分。

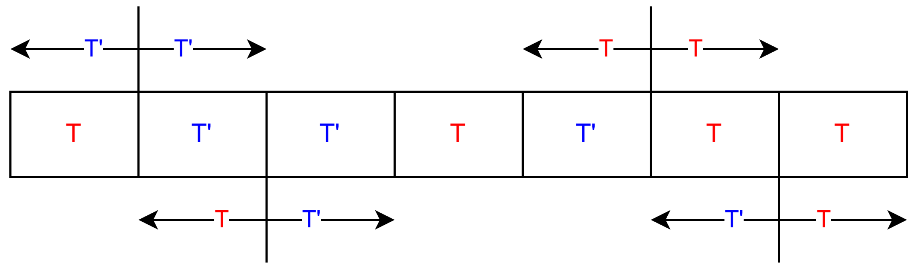
由归纳法可得，P 的第 i 位表示的字符即为 $w(i)$。

Lemma 1 *若 $|S| > 1$，则 S 为 Thue-Morse 序列的子串当且仅当 S 存在至少一组 T.M. 拆分。*

Proof 1 *一方面，若 $S(|S| > 1)$ 为 Thue-Morse 序列的子串，不妨设 $S = P[l : r]$，在 $(l, r] \cap \mathbb{N}$ 找到 lowbit 最大的数 i，则 $\mathrm{rev}(P[l : i-1])$ 和 $P[i : r]$ 均为 T.M. 串，故必要性得证；*

另一方面，若 $S = A + B$，且 $\mathrm{rev}(A)$ 和 B 均为 T.M. 串，则找到一个 k 满足 $k \in \mathbb{N}, 2 \nmid k, 2^k \ge \max(|A|, |B|)$，设 $T = P[0 : 2^k - 1], T' = P'[0 : 2^k - 1]$，得出 $T = \mathrm{rev}(T')$，且 A 和 B 均为 S 或 T 的前缀，$P[0 : 7 \times 2^k - 1] =$

$T+T'+T'+T+T'+T+T$，所以无论 $\mathrm{rev}(A), B$ 是 T 还是 T' 的前缀，$A+B$ 一定是 $P[0:7\times 2^k-1]$ 的子串，故充分性得证。



由 Proof 1 也可以分析得到 S 在 P 中匹配的位置上界。

Lemma 2 若 $S, \mathrm{rev}(S)$ 均为 $T.M.$ 串，则 $|S|$ 为 2 的非负整数幂。

Proof 2 反证法，若 $|S|$ 不为 2 的非负整数幂。

则 $S_i = x\oplus w(i) = y\oplus w(|S|-1-i)$，故 $\forall i\in[0,|S|)\cap\mathbb{N}$，$w(i)\oplus w(|S|-1-i)$ 为定值 $w(|S|-1)$。

由于 $|S|$ 不为 2 的非负整数幂，那么一定可以找到非负整数 k 使得 $\mathrm{bit}_k(|S|-1)=0$ 且 $\mathrm{bit}_{k+1}(|S|-1)=1$。

则当 $i=2^k$ 时，$w(|S|-1)=w(i)\oplus w(|S|-1-i)=1\oplus w(|S|-1)$，导出矛盾，故 $|S|$ 一定为 2 的非负整数幂。

Lemma 3 若 $S=A+B+C(|A|,|B|,|C|>0)$ 且 $(A,B+C),(A+B,C)$ 均为 $T.M.$ 拆分，那么 $|B|$ 为 2 的非负整数幂。

Proof 3 发现 B 和 $\mathrm{rev}(B)$ 都是 $T.M.$ 串，由 *lemma 2* 可得 $|B|$ 是 2 的非负整数幂。

Lemma 4 对于任意 *01* 字符串 $S(|S|>1)$，S 的 $T.M.$ 拆分个数不超过 2。

Proof 4 若 *01* 字符串 $S=A+B+C+D(|A|,|B|,|C|,|D|>0)$ 有三个 $T.M.$ 拆分：$(A,B+C+D),(A+B,C+D),(A+B+C,D)$。

则 $|B|+|C|,|B|,|C|$ 均为 2 的非负整数幂，得到 $|B|=|C|$。

由于 $B+C+D$ 为 $T.M.$ 串，故 $C_0=B_0\oplus 1$ 且 $D_0=B_0\oplus 1$；

由于 $C+D$ 为 $T.M.$ 串，故 $D_0 = C_0 \oplus 1$。

所以 $B_0 \oplus 1 = D_0 = C_0 \oplus 1 = B_0 \oplus 1 \oplus 1 = B_0$，导出矛盾，故 *01* 字符串 S 至多只有两个 $T.M.$ 拆分。

Lemma 5 若 $|A|, |B|, |C| > 0$，则 $(A, B+C), (A+B, C)$ 为 $A+B+C$ 的 $T.M.$ 拆分当且仅当 $|A|, |C| \le |B|$，且 $\mathrm{rev}(A+B)$ 和 $B+C$ 均为 $T.M.$ 串。

Proof 5 一方面，若 $|A|, |B|, |C| > 0$ 且 $(A, B+C), (A+B, C)$ 为 $A+B+C$ 的 $T.M.$ 拆分，则 $|B|$ 为 2 的非负整数幂，假设 $|C| > |B|$，则和 *Proof 4* 类似，有 $B_0 \oplus 1 = C_{|B|} = C_0 \oplus 1 = B_0 \oplus 1 \oplus 1 = B_0$，导出矛盾，故 $|C| \le |B|$，类似地 $|A| \le |B|$ 同样成立。所以，必要性得证。

另一方面，若 $|A|, |C| \le |B|$，$|A|, |B|, |C| > 0$，且 $\mathrm{rev}(A+B)$ 和 $B+C$ 均为 $T.M.$ 串，则 $|B|$ 为 2 的非负整数幂，由于 $|A|, |C| \le |B|$，故 $\mathrm{rev}(A), C$ 同样为 $T.M.$ 串，故 $(A, B+C), (A+B, C)$ 为 $A+B+C$ 的 $T.M.$ 拆分。所以，充分性得证。

接下来考虑如何求出 $g(S)$，发现只需计算【所有子串的 T.M. 拆分个数之和】减去【所有包含 2 个 T.M. 拆分的子串个数】。前者只需要枚举拆分的分界点在 S 中对应的位置，向左向右求出最长的 T.M. 串长并计算贡献；后者需要先枚举中间串 $|B|$ 的长度 2^k，然后枚举该串位置，再向左向右匹配最多 2^k 个位置，计算贡献。

然后，接着 Subtask 2 的 idea，考虑计算拆分点对每一个 $c(l,r)$ 的贡献。在求 $g(S)$ 的做法上，枚举一下拆分点或 B 两边匹配 P 还是 P'，那么对 c 的贡献就是一次矩形加。

于是，现在的问题转化为，存在 $O(n \log n)$ 次对 c 的矩形加，最终查询：

$$\sum_{l \le l' \le r' \le r} p(l, l'-1) c(l', r') p(r'+1, r)$$

至此，问题已经完全转化为一道数据结构问题，考虑扫描线，可以使用线段树、矩阵乘法解决（std 使用了历史和的写法，常数较小），时间复杂度 $O(n \log^2 n + q \log n)$。

其他部分分解法

- 取出 Thue-Morse 序列的前 $O(|S|)$ 项，使用 SAM 匹配带 ? 字符串计算贡献，运行时间与 ? 个数相关很大，能够获得不少的分数；
- 对于特殊性质 C，无需使用扫描线加线段树，可以直接差分计算贡献，较大程度降低代码量；
- 对于 n 较小的数据点，对于 c 的矩形加可以使用二维差分解决，再 $O(n^2)$ 递推出所有 $h(S[l:r])$ 即可；
- 对于一些基于枚举 ? 替换方式或复杂度与 ? 个数相关较大的做法，可以获得特殊性质 B 的分数；
- 对于一些复杂度正确但常数略大的做法，留了一档子任务 9 的分数，但 std 的运行时间约为 1s，并无刻意卡常。

总结与致谢

相关题目

本题在命题上的部分 idea，由该题 [2] 得到，在数据结构的使用上，与该题 [3] 类似。

总结

总体上，解决本题过程的思路流畅，各个结论需要选手循序渐进地推导，本题综合考察了选手对 Thue-Morse 序列的理解、分析合法字符串性质以及使用数据结构解决问题的能力，所用到的算法难度不高，但重在考察选手的思维能力。

致谢

感谢方心童同学、刘海峰同学、林卓铖同学在本题命制过程中提出的诸多富有价值的意见并为本文审稿。

参考文献

[1] Wikipedia - Thue-Morse sequence

[2] SDOI2019 d2t3 连续子序列

[3] NOIP2022 t4 比赛