KMP

2022 年 1 月 15 日

基础定义

字符串

S：无特殊说明，字符串仅由 26 个小写字母'a'-'z' 构成，并用大写字母表示一个字符串。
$|S|$：表示一个字符串的长度。
$S[i]$：表示字符串 S 第 i 个位置的字母，下标从 1 开始。

子串

$S[l, r]$：表示字符串 S 从第 l 到第 r 个字母顺次连接而成的新字符串。
$Prefix_S[i]$：表示字符串 S 的长度为 i 的前缀，$Prefix_S[i] = S[1, i]$
$Suffix_S[i]$：表示字符串 S 的长度为 i 的后缀，$Suffix_S[i] = S[|S| - i + 1, |S|]$

注意，如果语境中只存在一个字符串，则可以简写为 $Prefix[i]$ 与 $Suffix[i]$

重要定义

Border

如果字符串 S 的同长度的前缀和后缀完全相同，即 $Prefix[i] = Suffix[i]$ 则称此前缀（后缀）为一个 Border（根据语境，有时 Border 也指长度）。

特殊地，字符串本身也可以是它的 Border，具体是不是根据语境判断。

例子

若 $\mathrm{S} = \mathrm{bbabbab}$，试求所有 Border

重要定义

定义

周期和循环节

对于字符串 S 和正整数 p，如果有 $S[i] = S[i-p]$，对于 $p < i \leq |S|$ 成立，则称 p 为字符串 S 的一个周期。

特殊地，$p = |S|$ 一定是 S 的周期

定义

若字符串 S 的周期 p 满足 $p \mid |S|$，则称 p 为 S 的一个循环节

特殊地，$p = |S|$ 一定是 S 的循环节

重要性质

Border vs 周期

p 是 S 的周期 $\Leftrightarrow |S| - p$ 是 S 的 Border

证明.

p 为 S 的周期 $\Leftrightarrow S[i-p] = S[i]$
q 为 S 的 Border $\Leftrightarrow S[1,q] = S[|S|-q+1,|S|] \Leftrightarrow$
$S[1] = S[|S|-q+1], S[2] = S[|S|-q+2], \ldots, S[q] = S[|S|]$
易得：$p+q=|S|$ $\square$

因此，字符串的周期性质等价于 Border 的性质，
求周期也等价于求 Border。

警告：Border 不具有二分性。

Border 的 Naive 求法

暴力

枚举 $1 \leq i \leq |S|$，暴力验证是否有 $Preffix[i] == Suffix[i]$ 。

复杂度 $O(N^2)$

优雅的暴力

使用 $Hash$ 验证 $Prefix[i] == Suffix[i]$

复杂度 $O(N)$，常数很大，容易构造 $Hash$ 冲突

Border 的性质

传递性

S 的 Border 的 Border 也是 S 的 Border

证明.

设 p 为 S 的 Border，则有 $Preffix_S[p] == Suffix_S[p]$，即 $S[1,p] == S[|S|-p+1,|S|]$

设 q 为 $S[1,p]$ 的 Border，则有 $Prefix_{S[1,p]}[q] == Suffix_{S[1,p]}[q]$，即 $S[1,q] == S[p-q+1,p]$，进而 $S[1,q] == S[|S|-q+1,|S|]$，因此 q 也是 S 的 Border。

$\square$

求 S 的所有 Border 等价于求所有前缀的最大 Border

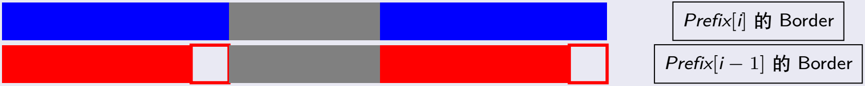
KMP

Next 数组

$next[i] = Preffix[i]$ 的非平凡的最大 *Border*

$next[1] = 0$

考虑 $Prefix[i]$ 的所有（长度大于 1 的）Border，去掉最后一个字母，就会变成 $Prefix[i-1]$ 的 Border。



因此求 $next[i]$ 的时候，可以遍历 $Prefix[i-1]$ 的所有 Border，即 $next[i-1], next[next[i-1]], \ldots, 0$，检查后一个字符是否等于 $S[i]$。

这看着也太 $O(N^2)$ 了 ??

KMP

复杂度分析

考虑使用势能分析进行讨论：

- 如果 $next[i] = next[i-1] + 1$，则势能会增加 **1**
- 否则势能会先减少到某个 $next[j]$，然后有 $next[i] = next[j] + 1$，势能也会增加 **1**，在寻找 $next[j]$ 的过程中，势能会减少，每次至少减少 **1**。
- 还有一种情况，$next[i] = 0$，势能清空，且不会增加。

综上，势能总量为 $O(N)$，因此整体的复杂度也是 $O(N)$，常数为 2 左右（很小）。空间复杂度也为 $O(N)$。

例题 1

牛客 15165

字符串 S 长度不超过 10^6，求一个最长的子串 T，满足：

- T 为 S 的前缀。
- T 为 S 的后缀。
- T 在 S 中至少出现 3 次。

例题 1

牛客 15165

字符串 S 长度不超过 10^6，求一个最长的子串 T，满足：

- T 为 S 的前缀。
- T 为 S 的后缀。
- T 在 S 中至少出现 3 次。

题解

首先用 KMP 求出 S 的所有 Border，答案为 $next[n]$ 或者 $next[next[n]]$。

例题 3

某谷 3375 - 模板

给出两个字符串 S，和 T，求出 T 在 S 中所有出现位置。

例如：$S = abababc$, $T = aba$，则 T 在 S 的所有出现位置为 1 和 3。

字符串匹配

Naive 的匹配

枚举起始位置，然后暴力匹配。复杂度 $O(N^2)$

优雅的暴力

枚举起始位置，然后用 *Hash* 检查。复杂度 $O(N)$，常数极大。字符集很大时的处理比较繁琐。

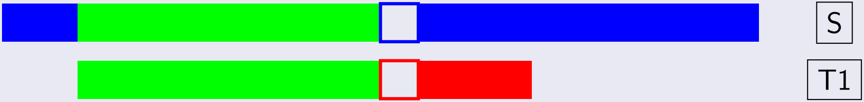
KMP 匹配

KMP 充分利用前缀匹配的有效信息，即 *next* 数组（Border 的性质），进行快速转移。

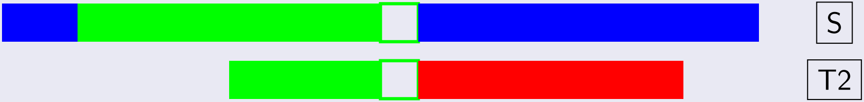
字符串匹配

KMP 匹配

假设在暴力匹配的过程中发生了如下情况：



由于红蓝方块位置的字符不匹配，因此需要合理向右移动 *T* 字符串，在成功匹配了绿色方块位置的字符之后，才可以继续向后匹配：



此时可以清晰的看到，*T2* 绿条部分，恰好是 *T1* 绿条部分的 Border。也就是说，当遇到匹配失败的字符时，只需要考虑 Border 所有的长度即可，非 Border 长度一定不会匹配的更“远”。

字符串匹配

KMP 匹配的复杂度分析

使用 KMP 进行字符串匹配时，利用势能分析，不难看出总势能为 $|S|$，再加上预处理 T 的 $next$ 数组，复杂度为 $O(|S| + |T|)$。

例题 4

牛客 14694

给出两个正整数数组 A 和 B，长度分别为 $n \le m \le 2 \cdot 10^5$，求 A 有多少个长度为 m 的区间 A' 满足：
$(A'[1] + B[1])\%k = (A'[2] + B[2])\%k = \ldots (A'[m] + B[m])\%k$

例题 4

牛客 14694

给出两个正整数数组 A 和 B，长度分别为 $n \le m \le 2 \cdot 10^5$，求 A 有多少个长度为 m 的区间 A' 满足：

$(A'[1] + B[1])\%k = (A'[2] + B[2])\%k = \dots (A'[m] + B[m])\%k$

题解

要求满足的条件为：

(1) $(A'[1] + B[1])\%k = (A'[2] + B[2])\%k$

(2) $(A'[2] + B[2])\%k = (A'[3] + B[3])\%k$

$\dots$

(m) $(A'[m-1] + B[m-1])\%k = (A'[m] + B[m])\%k$

例题 4

题解

移项得到：

(1) $(A'[1]-A'[2])\%k=-(B[1]-B[2])\%k$

(2) $(A'[2]-A'[3])\%k=-(B[2]-B[3])\%k$

$\ldots$

(m) $(A'[m-1]-A'[m])\%k=-(B[m-1]-B[m])\%k$

因此答案等于 $-Diff_B$ 数组在 $Diff_A$ 数组中的出现次数。
进而问题转化为字符串匹配问题，可以使用 KMP 解决。

拓展

Border 的性质

周期定理：若 p, q 均为串 S 的周期，则 (p, q) 也为 S 的周期。

一个串的 Border 数量是 $O(N)$ 个，但他们组成了 $O(logN)$ 个等差数列。

KMP 的推广

拓展 KMP(a.k.a Z 算法)

KMP 自动机，Border 树

AC 自动机，即 KMP 的多串模式。

Trie 图，即 KMP 自动机的多串模式。