

Knuth–Morris–Pratt(KMP) 算法 *

张晴川
qzha536@aucklanduni.ac.nz

April 18, 2020

1 题意

设文本串为 $T(1 \leq |T| \leq 10^5)$ ，模式串为 $S(1 \leq |S| \leq 10^5)$ ，求 S 在 T 中的所有出现位置。下标均从 1 开始。

例： $S = 14, T = 114514$ ，那么匹配的末尾位置为 $\{3, 6\}$

2 算法

首先我们考虑用一个没有出现过的字符 # 把 S 和 T 拼成一个串 $S\#T$ ，例如 $14\#114514$ 。

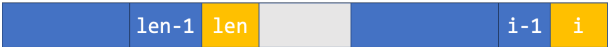
现在考虑对于位置 i ，有哪些以它结尾的串等于相同长度的前缀，以 i 结尾的前缀本身除外。例如对于串 $aba\#ababa$ ，最后一个位置 $i = 9$ 的话，匹配的长度为 $\{3, 1, 0\}$ ，分别对应 $\{aba, a, \epsilon\}$ ¹，定义这样的串为以 i 结尾的 border。

问题转化 我们用 $match(i, len) = true/false$ 表示以 i 结尾，长度为 len 的串是否是 border，用 $borders[i]$ 表示以 i 结尾，所有 border 的集合。原问题可以转化为：对于多少位置 i ， $match(i, |S|) = true$ 。

.....

如何求 $borders[i]$ 首先我们可以发现 $match(i, len) = true$ 成立有两个条件：

- 1. $match(i-1, len-1) = true$
- 2. $S[len] = S[i]$

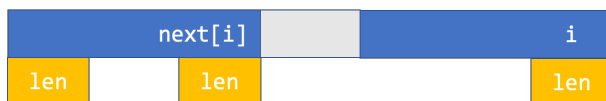


*更多内容请访问：<https://github.com/SamZhangQingChuan/Editorials>
¹ ϵ 表示空串

于是我们可以枚举 $\text{borders}[i-1]$ 里的元素, 如果下一位恰好和 $S[i]$ 匹配, 那么就 $+1$ 就可以得到 $\text{borders}[i]$ 里的一个元素。最后加入 0 即可。设 $n = |S| + |T|$, 由于一共有 n 个位置, 每次最多枚举 n 个元素, 复杂度为 $O(n^2)$, 需要加速。

.....

更高效的表示方法 我们用 $\text{next}[i]$ 表示 $\text{borders}[i]$ 里最长的 border。假设 len 也是 i 的一个 border, 不难发现 len 也是 $\text{next}[i]$ 的 border。(参考下图, 想想为什么)



所以除了 $\text{next}[i]$ 外, 其余元素都是 $\text{next}[i]$ 的 border, 于是可以得到:

$$\text{borders}[i] = \{\text{next}[i]\} \cup \text{borders}[\text{next}[i]]$$

展开写的话就是:

$$\text{borders}[i] = \{\text{next}[i], \text{next}[\text{next}[i]], \dots, 0\}$$

另外根据定义, $\text{borders}[1] = \{0\}$ 。

.....

递推求解 由上述解释, 只要求出 $\text{next}[i]$, 就可以得到整个 $\text{borders}[i]$ 。如果知道了 $\text{borders}[i]$, $\text{next}[i+1]$ 就等于 $\text{borders}[i]$ 里从大到小第一个可以转移到 $i+1$ 的元素再加 1。如果全部无法转移, 那么 $\text{next}[i+1] = 0$ 。

.....

完全解决 在求出所有 $\text{next}[i]$ 之后, 只需要计算有哪些位置的 $\text{next}[i] = |S|$ 即可。

3 习题

1. 实现 KMP 算法
2. 「NOI2014」动物园
3. 以同样思路分析 AC 自动机的算法。

4 核心代码

```
1  int KMP(string S, string T) {
2      string s = " " + S + "#" + T;
3      vector<int> next(s.size());
4      for(int i = 1; i < s.size(); i++) {
5          for(int len = next[i]; len = next[len]) {
6              if(s[len + 1] == s[i + 1]) {
7                  next[i + 1] = len + 1;
8                  break;
9              } else {
10                 if(len == 0) {
11                     break;
12                 }
13             }
14         }
15     }
16     int count = 0;
17     for(int i = 0; i < T.size(); i++) {
18         count += next[i + S.size() + 1] == S.size();
19     }
20     return count;
21 }
```