# 字符串进阶：扩展 kmp

Ebola

Institute of Mathematics,
Zhejiang University.

Jan, 2024

1. 基础回顾
2. 扩展 kmp （Z 函数）

1. 基础回顾
2. 扩展 kmp （Z 函数）

# kmp 算法

kmp 算法解决的是单文本串、单模式串匹配问题，即：给定文本串 $S$ 和模式串 $T$，求 $T$ 在 $S$ 中完整出现的所有位置。

复杂度是 $O(n)$.

# kmp 算法

我们来回顾一下 kmp 算法的流程。首先有一个 next 数组，我这里把它写成 $\pi$，定义如下：

$$\pi(i) = \max\{k \mid T[1,...,k] = T[i-k+1,...,i],\ k = 0,...,i-1\}$$

## kmp 算法

我们来回顾一下 kmp 算法的流程。首先有一个 next 数组，我这里把它写成 $\pi$，定义如下：

$$\pi(i) = \max\{k \mid T[1, ..., k] = T[i-k+1, ..., i],\ k = 0, ..., i-1\}$$

我们把 $T$（长 $m$）和 $S$（长 $n$，$n > m$）放到一起：`A = T#S`

对于一个位置 $i$，如果 $\pi(i) = m$，说明 $T$ 从 $i-m+1$ 位置开始出现；反过来，如果 $T$ 从 $i-m+1$ 位置开始出现，那么一定 $\pi(i) = m$。

我们只要找到 $\pi(i) = m$ 的位置，然后就知道答案了。

# kmp 算法

现在考虑如何求 $\pi$ 数组。
最暴力的求法，$O(n^3)$：

```
for(int i = 1; i <= n; i++){
    int k;
    for(k = i-1; k >= 0; k--)
        if(子串(1,...,k) == 子串(i-k+1,...,i))
            break;
    pi[i] = k;
}
```

# kmp 算法

我们观察到：$\pi(i+1) \leq \pi(i)+1$，为什么？(举例说明)

# kmp 算法

我们观察到：$\pi(i+1) \leq \pi(i)+1$，为什么？(举例说明)

借助这个观察，我们可以优化代码：

```
for(int i = 1; i <= n; i++){
    int k;
    for(k = pi[i-1]+1; k >= 0; k--)
        if(子串(1,...,k) == 子串(i-k+1,...,i))
            break;
    pi[i] = k;
}
```

这个复杂度是 $O(n^2)$

## kmp 算法

现在我们看这个集合：

$$\mathcal{P}(i) = \{k \mid T[1, ..., k] = T[i-k+1, ..., i],\ k = 0, ..., i-1\}.$$

注意到 $\pi(i) = \max \mathcal{P}(i)$.

## kmp 算法

现在我们看这个集合：

$$\mathcal{P}(i)=\{k \mid T[1,...,k]=T[i-k+1,...,i],\ k=0,...,i-1\}.$$

注意到 $\pi(i)=\max\mathcal{P}(i)$.

又注意到 $\pi(i)-1\in\mathcal{P}(i-1)$（为什么？)。

# kmp 算法

现在我们看这个集合：

$$\mathcal{P}(i) = \{k \mid T[1, ..., k] = T[i-k+1, ..., i],\ k = 0, ..., i-1\}.$$

注意到 $\pi(i) = \max \mathcal{P}(i)$.

又注意到 $\pi(i) - 1 \in \mathcal{P}(i-1)$（为什么？）。那么 $k$ 循环可以只遍历 $\mathcal{P}(i-1)$ 里的数，像这样：

```
pi[1] = 0;
for(int i = 2; i <= n; i++){
    int k;
    for(k = max(P(i-1)); k >= 0; k=P(i-1)里面比k小的那个数)
        if(A[k+1] == A[i])
            break;
    pi[i] = k + 1;
}
```

# kmp 算法

现在我们看这个集合：

$$\mathcal{P}(i) = \{k \mid T[1, ..., k] = T[i-k+1, ..., i],\ k = 0, ..., i-1\}.$$

注意到 $\pi(i) = \max \mathcal{P}(i)$.

又注意到 $\pi(i) - 1 \in \mathcal{P}(i-1)$（为什么？）。那么 $k$ 循环可以只遍历 $\mathcal{P}(i-1)$ 里的数，像这样：

```
pi[1] = 0;
for(int i = 2; i <= n; i++){
    int k;
    for(k = max(P(i-1)); k >= 0; k=P(i-1)里面比k小的那个数)
        if(A[k+1] == A[i])
            break;
    pi[i] = k + 1;
}
```

注意到 $\mathcal{P}(i)$ 里面第二大的数是 $\pi(\pi(i))$，第三大的数是 $\pi(\pi(\pi(i)))$，⋯⋯（为什么？）所以上面的 $k$ 循环很好实现。总复杂度 $O(n)$.

## [HNOI2008] GT 考试

阿申准备报名参加 GT 考试，准考证号为 $N$ 位数 $X_1, X_2 ... X_n$ $(0 \le X_i \le 9)$，他不希望准考证号上出现不吉利的数字。

他的不吉利数字 $A_1, A_2, \cdots, A_m$ $(0 \le A_i \le 9)$ 有 $M$ 位，不出现是指 $X_1, X_2 \cdots X_n$ 中没有恰好一段等于 $A_1, A_2, \cdots, A_m$，$A_1$ 和 $X_1$ 可以为 $0$。

阿申想知道不出现不吉利数字的号码有多少种，输出模 $K$ 取余的结果。

$N \le 10^9$，$M \le 20$，$K \le 1000$。

# [HNOI2008] GT 考试

设 $f_{i,j}$ 表示当前考虑到第 $i$ 位，其中末尾和 $A_1 ... A_m$ 匹配了 $j$ 位。
设 $g_{j,k}$ 表示当前末尾匹配了 $j$ 位，如果添加一个数字后能够匹配 $k$ 位，有多少种添加数字的方案。这是一个可以预处理的数组。

## [HNOI2008] GT 考试

设 $f_{i,j}$ 表示当前考虑到第 $i$ 位，其中末尾和 $A_1 \ldots A_m$ 匹配了 $j$ 位。
设 $g_{j,k}$ 表示当前末尾匹配了 $j$ 位，如果添加一个数字后能够匹配 $k$ 位，有多少种添加数字的方案。这是一个可以预处理的数组。

我们得到转移方程：

$$f_{i,j} = \sum_{k=0}^{m-1} f_{i-1,k} g_{k,j}.$$

显然可以用矩阵快速幂优化（应该都会吧？)。

# [HNOI2008] GT 考试

设 $f_{i,j}$ 表示当前考虑到第 $i$ 位，其中末尾和 $A_1...A_m$ 匹配了 $j$ 位。
设 $g_{j,k}$ 表示当前末尾匹配了 $j$ 位，如果添加一个数字后能够匹配 $k$ 位，有多少种添加数字的方案。这是一个可以预处理的数组。

我们得到转移方程：

$$f_{i,j} = \sum_{k=0}^{m-1} f_{i-1,k} g_{k,j}.$$

显然可以用矩阵快速幂优化（应该都会吧 ?）。

$g$ 数组可以用 kmp 来算。枚举当前匹配长度 $j$，再枚举下一个数字 $c$，用 next 数组算一下添加 $c$ 之后末尾匹配长度是多少（记为 $k$），然后令 $g_{j,k}++$.

## 扩展 kmp （Z 函数）

对于一个长度为 $n$ 的字符串，定义 $z_i$ 表示 $s$ 与 $s[i...n]$ 的最长公共前缀长度。这就是 **Z 函数**。

在研究如何求 Z 函数之前，我们先来看看 Z 函数的应用。

# 字符串匹配

给定文本串 $S$ 和模式串 $T$，求 $T$ 在 $S$ 中完整出现的所有位置。

# 字符串匹配

给定文本串 $S$ 和模式串 $T$，求 $T$ 在 $S$ 中完整出现的所有位置。

【解】令 A=T#S，求出 $A$ 的 Z 函数，然后找到所有 $z_i = |T|$ 的位置即可。

# 本质不同的子串

给定文本串 $S$，现在往 $S$ 的开头添加一个字母 $c$，问增加了几个本质不同的子串。

# 本质不同的子串

给定文本串 $S$，现在往 $S$ 的开头添加一个字母 $c$，问增加了几个本质不同的子串。

【解】求 $cS$ 的 Z 函数，取最大值 $z_{\max}$，显然，长度超过 $z_{\max}$ 的前缀都是新增的本质不同子串（反之，长度不超过 $z_{\max}$ 的前缀都不是新增的本质不同子串）。

# 字符串的最小周期

给定一个长度为 $n$ 的字符串 $S$，找到其最短的整周期，即寻找一个最短的字符串 $T$，使得 $S$ 可以被若干个 $T$ 拼接而成的字符串表示。

# 字符串的最小周期

给定一个长度为 $n$ 的字符串 $S$，找到其最短的整周期，即寻找一个最短的字符串 $T$，使得 $S$ 可以被若干个 $T$ 拼接而成的字符串表示。

【解】求 $S$ 的 Z 函数，找到最小的 $n$ 的因数 $i$，满足 $i + z_i = n$.

*Thank You*