

# 字符串匹配算法（中）

## 字符串匹配算法（中）

- 一、字符串的哈希匹配算法
- 二、shift\_and 算法
- 三、字典树结构
- 四、海贼 OJ-282-最大异或和

## 一、字符串的哈希匹配算法

---

1. 可以使用哈希操作判断两个字符串是否相等
2. 哈希值不同的话，两个字符串一定不相等，从而就不需要按位比较了
3.  $H = (\sum_{k=0}^{n-1} C_k \times base^k) \% P$
4. 在文本串上，每一位字符串哈希值的前缀和，方便一会求区间和
5.  $H(i, j) = (HS_j - HS_{i-1}) \times (base^i)^{-1} \% P$

快速求逆元的推导过程

$$\begin{aligned}x \times x^{-1} &\equiv 1 \pmod{P} \\ \text{令: } P \% x &= r \\ P &= kx + r \\ kx + r &\equiv 0 \pmod{P} \\ kr^{-1} + x^{-1} &\equiv 0 \pmod{P} \\ x^{-1} &\equiv -kr^{-1} \pmod{P}\end{aligned}$$

哈希匹配法

abdefc

1 2 4 5 6 5 3

$s_1$   $s_2$

4 3

哈希值

base

$C_0 C_1 C_2 C_3$

$(C_0 \text{base}^0 + C_1 * \text{base}^1 + C_2 * \text{base}^2 + C_3 * \text{base}^3) \% \text{mod}$

$H = (\sum_{i=1}^n C_i * \text{base}^i) \% \text{mod}$

PNA i j

$H_i = (\sum_{k=i}^j C_k * \text{base}^k) \% \text{mod}$

$H = (H_i * (\text{base}^k)) \% \text{mod}$

$5 \times 3 = 15$

$5 / 3 = 0.6$

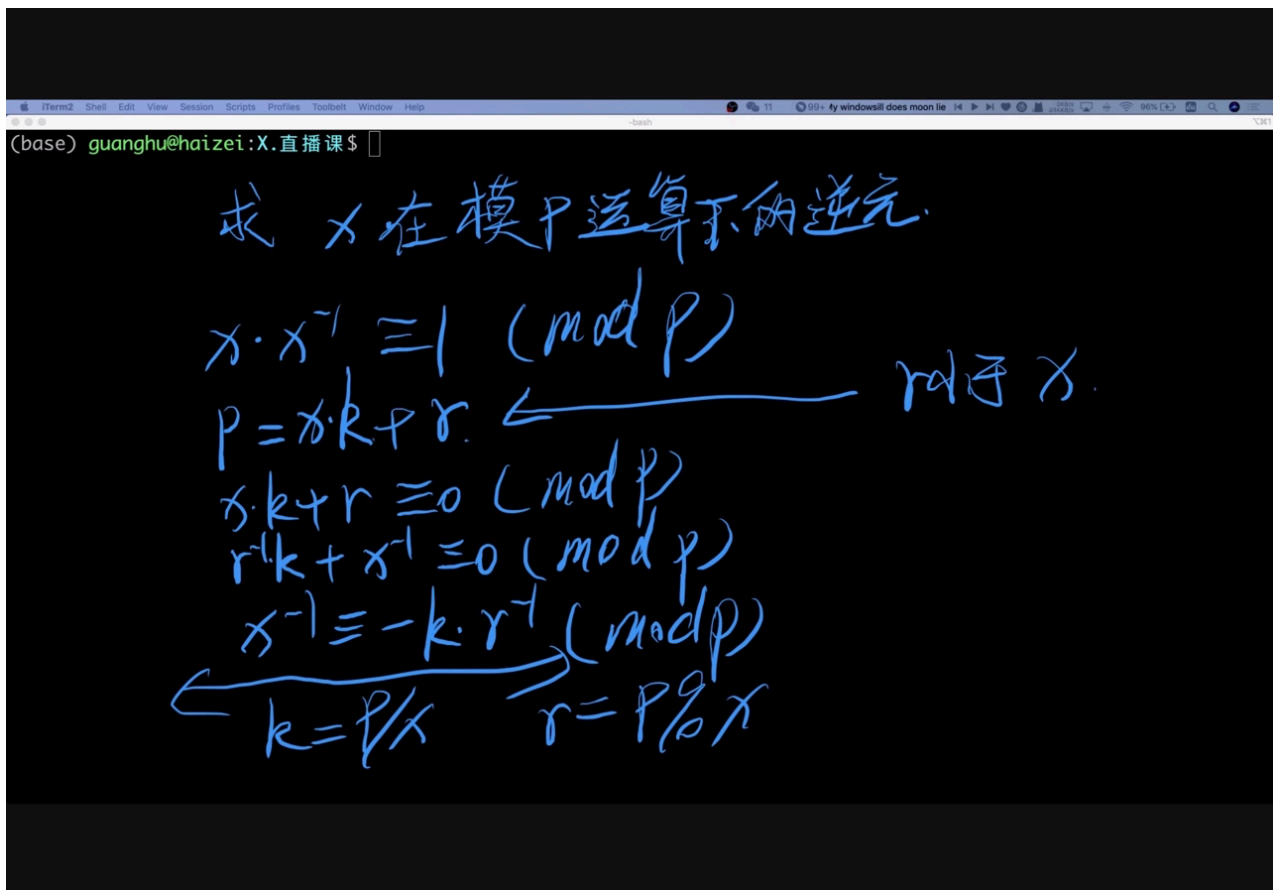
$5 * \frac{1}{3} = 0.6$

3 和  $\frac{1}{3}$  互为逆元

$x_1 \cdot x_2 \% p = 1 \rightarrow x_1 \text{ 与 } x_2 \text{ 互为逆元}$

$2 * 4 \% 7 = 1 \rightarrow 2 \text{ 与 } 4 \text{ 互为逆元}$

$(16 \ 14) \% 7 = 4 \Leftrightarrow (16 * 2) \% 7 = 4$



## 二、shift\_and 算法

1. 第一步对模式串做特殊处理，把每一种字符出现的位置，转换成相应的二进制编码
2. 后续匹配的过程中跟模式串一毛钱关系都没有
3.  $p_i = (p_{i-1} \ll 1) \& d[s_i]$
4.  $p_i$  第  $j$  位二进制为 1，代表当前位置为结尾，可以匹配成功模式串的第  $j$  位

## SHIFT-AND 算法

母串 **s**

a	e	c	a	e	a	e	c	a	e	d
---	---	---	---	---	---	---	---	---	---	---

	0	1	2	3	4	5
d[a]	1	0	0	1	0	0
d[c]	0	0	1	0	0	0
d[d]	0	0	0	0	0	1
d[e]	0	1	0	0	1	0
P	0	0	0	0	0	0

$$P_2 = (P_1 \ll 1 | 1) \& d[s[2]]$$

$$P = (P \ll 1 | 1) \& d[s[i]]$$

$$P \& (1 \ll (n - 1)) \neq 1$$

## 三、字典树结构

1. 也叫做：前缀索引树
2. 把每个字符串按照前缀的顺序插入到树形结构中
3. 字典树可以用于字符串的排序，时间复杂度  $O(n)$

## 四、海贼 OJ-282-最大异或和

1. 思考：如何使得异或结果尽可能大
2. 结论：参与异或运算的两个数字，参与异或运算的每一位尽可能不同
3. 问题转换为：确定一个数字的情况下，找到从高为到低位与当前数字尽量不同的另外一个数字
4. 把每个数字看成一个二进制字符串，插入到字符串中，采用贪心策略进行选择