4.3 串的模式匹配算法

- 定义 子串的定位操作通常称做串的模式匹配(其中P、T称为模式串，PaTtern），是各种串处理系统中最重要的操作之一。
- 名词 在串的模式匹配中，子串P称为模式，主串S称为目标。

➢ 【示例】 目标 S : “Beijing”

➢ 模式 P : “jin”

➢ 匹配结果 = 3（匹配位置从 0 开始）

□ 讨论两种匹配算法：BF 算法和 KMP 算法。

4.3.1 求子串位置的定位函数

Brute-Force简称为**BF算法**，亦称简单匹配算法。采用穷举的思路。

- 初始时让目标S的第 0 位与模式P的第 0 位对齐；
- 顺序比对目标S与模式P中的对应字符：
  - 若 P 与 S 比对发现对应位不匹配，则本趟失配。将 P 右移一位与 S 对齐，进行下一趟比对；
  - 若 P 与 S 对应位都相等，则匹配成功，返回 S当前比较指针停留位置减去 P 的长度，即目标S 中匹配成功的位置，算法结束。
  - 若 P 与 S 比对过程中，S 后面所剩字符个数少于 P 的长度，则模式匹配失败。

BF 算法匹配过程的示例

第1趟

i=2

S a b b a b a

P a b a

j=2

第2趟

i=1

S a b b a b a

P a b a

j=0

第3趟

i=2

S a b b a b a

P a b a

j=0

第4趟

i=6

S a b b a b a

P a b a

j=3

- 这是最简单的模式匹配算法。

```
int index(SqString S, SqString P, int pos)
{
    int i = pos - 1,  j = 0;
    while(i < S.length && j < P.length) {
        if(S.SString[i] == P.SString[j]) {
            i++;     //主串和子串依次匹配下一个字符
            j++;
        }
        else { //主串、子串指针回溯重新开始下一次匹配
            i = i - j + 1;   //主串从下一个位置开始匹配
            j = 0;           //子串从头开始匹配
        }
    }
    if(j >= P.length)
        return(i - P.length);//返回匹配的第一个字符的下标
    else
        return -1;          //模式匹配不成功
}
```

BF算法分析：

- 若设 n 为目标 S 的长度，m 为模式 P 的长度，匹配算法最多比较 $n-m+1$趟。若每趟比较都比较到模式 P 尾部才出现不等，要做 m 次比较，则在最坏情况下，总比较次数 $(n-m+1)*m$。在多数场合下 m 远小于 n，因此，算法的运行时间为$O(n*m)$。

- 低效的原因在于：算法在字符比较不相等，需要回溯（即$i=i-j+1$）：即退到s中的下一个字符开始进行继续匹配。
- 如果消除了每趟失配后为实施下一趟比较时目标指针的回退，可以提高模式匹配效率。

4.3.2 模式匹配的一种改进算法KMP

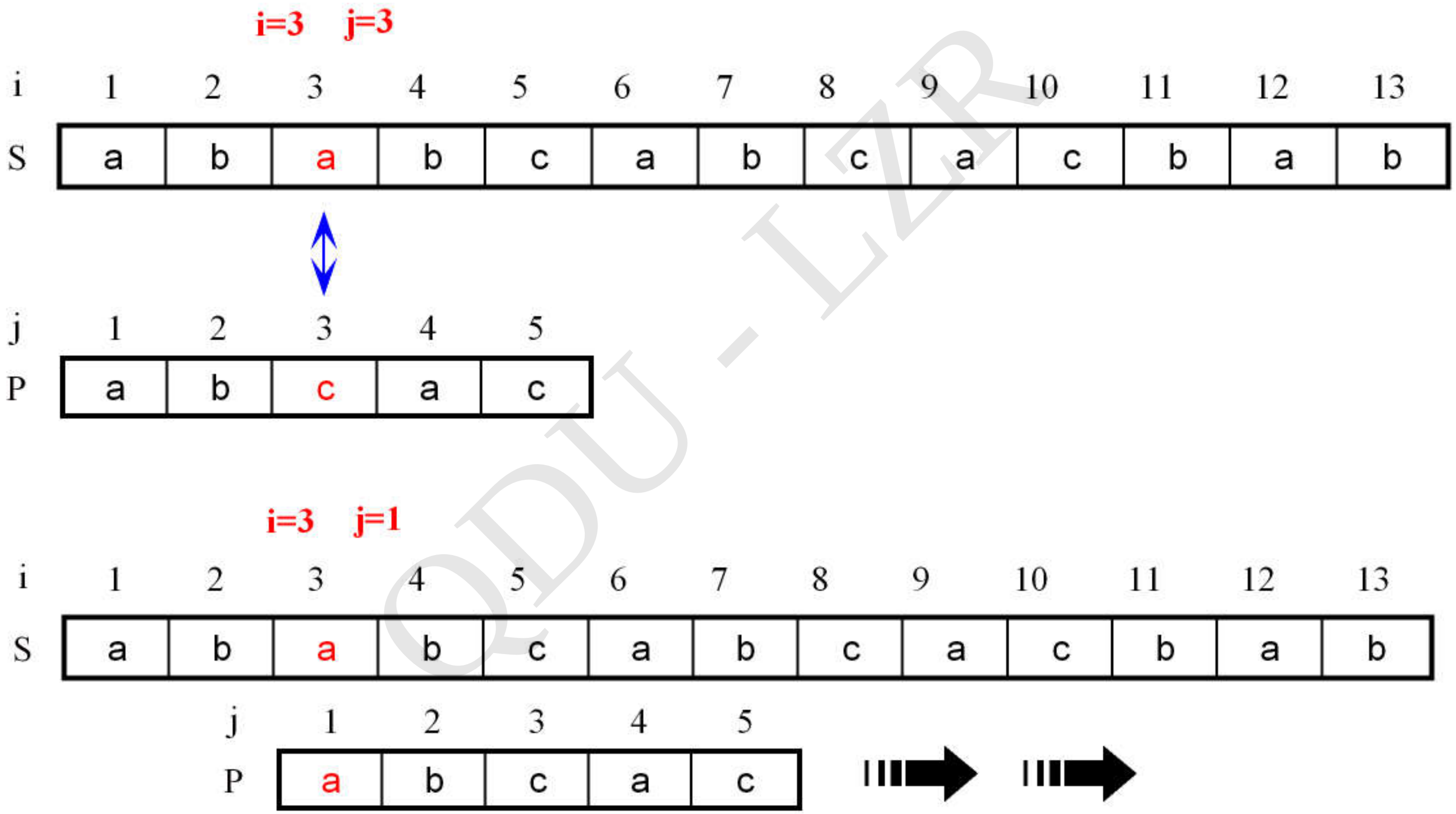
KMP算法是D.E.Knuth、J.H.Morris和V.R.Pratt共同提出的，简称KMP算法。

该算法较BF算法有较大改进：每当一趟匹配过程出现字符不相等时，主串指示器i不用回溯，而是利用已经得到的“部分匹配”结果，将模式串向右“滑动”尽可能远的一段距离后，继续进行比较。

S=“0000000000000000000000000000000001”

P=“000000000000000001”

改进算法的匹配过程示例



改进算法的匹配过程示例

i=7 j=5

i	1	2	3	4	5	6	7	8	9	10	11	12	13
S	a	b	a	b	c	a	b	c	a	c	b	a	b

j	1	2	3	4	5
P	a	b	c	a	c

改进算法的匹配过程示例

i=7　j=2

i	1	2	3	4	5	6	7	8	9	10	11	12	13
S	a	b	a	b	c	a	b	c	a	c	b	a	b

j	1	2	3	4	5
P	a	b	c	a	c

KMP算法就是一种基于分析模式串P蕴含信息的改进算法。

4.3.2 模式匹配的一种改进算法KMP

- **KMP算法思想**：若一趟匹配过程比对失配，在做下一趟匹配比对时，目标 S 的检测指针不回退，模式 P 右移，与 S 的检测指针对齐，再开始比对过程。
- 算法的时间代价：
  - 若每趟第一个不匹配，比较$n-m+1$趟，总比较次数最坏达$(n-m)+m = n$。
  - 若每趟第 m 个不匹配，总比较次数最坏亦达到 n。

KMP 算法匹配过程的示例

目标 S　$S_0 \quad S_1 \quad S_2 \quad \cdots\cdots \quad S_{m-1} \quad \cdots \quad S_{n-1}$

$\updownarrow \quad \updownarrow \quad \updownarrow \qquad\qquad \updownarrow$

模式 P　$p_0 \quad p_1 \quad p_2 \quad \cdots\cdots \quad p_{m-1}$　<u>若失配，右移</u>

目标 S　$S_0 \quad S_1 \quad S_2 \quad \cdots\cdots \quad S_{m-1} \quad S_m \quad \cdots \quad S_{n-1}$

$\updownarrow \quad \updownarrow \qquad\qquad \updownarrow \quad \updownarrow$

模式 P　$p_0 \quad p_1 \quad \cdots\cdots \quad p_{m-2} \quad p_{m-1}$　<u>若失配，右移</u>

目标 S　$S_0 \quad S_1 \quad \cdots\cdots \quad S_i \quad S_{i+1} \cdots\cdots \quad S_{i+m-2} \quad S_{i+m-1} \cdots \quad S_{n-1}$

$\| \quad \| \qquad\qquad \| \quad \|$

模式 P　$p_0 \quad p_1 \quad \cdots\cdots \quad p_{m-2} \quad p_{m-1}$

$$\begin{array}{lllllllllllll} \mathbf{S} & S_0\ S_1 \ldots S_{s-1} & S_s & S_{s+1} & S_{s+2} & \ldots & S_{s+j-1} & S_{s+j} & S_{s+j+1} \ldots S_{n-1} \\ & & \| & \| & \| & & \| & \times & \\ \mathbf{P} & & p_0 & p_1 & p_2 & \ldots & p_{j-1} & p_j & \end{array}$$

则有 $S_s\ S_{s+1}\ S_{s+2} \ldots S_{s+j-1} = p_0\ p_1\ p_2 \ldots p_{j-1}$ (1)

如果 $p_0\ p_1 \ldots p_{j-2} \neq p_1\ p_2 \ldots p_{j-1}$ (2)

则立刻可以断定

$$p_0\ p_1 \ldots p_{j-2} \neq S_{s+1}\ S_{s+2} \ldots S_{s+j-1}$$

下一趟必不匹配

改进算法的匹配过程示例

i=7 j=5

i	1	2	3	4	5	6	7	8	9	10	11	12	13
S	a	b	a	b	c	a	b	c	a	c	b	a	b

j	1	2	3	4	5
P	a	b	c	a	c

同样，若 $p_0\ p_1\ \cdots p_{j-3} \neq p_2\ p_3\ \cdots p_{j-1}$

则再下一趟也不匹配，因为有

$$p_0\ p_1\ \cdots p_{j-3} \neq S_{s+2}\ S_{s+3}\ \cdots\ S_{s+j-1}$$

直到对于某一个“k”值，使得

$$p_0\ p_1\ \cdots p_{k+1} \neq p_{j-k-2}\ p_{j-k-1}\ \cdots p_{j-1}$$

且 $p_0\ p_1\ \cdots p_k = p_{j-k-1}\ p_{j-k}\ \cdots p_{j-1}$

则

$$\begin{array}{lcccc} p_0\ p_1\ \cdots p_k = & S_{s+j-k-1} & S_{s+j-k} & \cdots & S_{s+j-1} \\ & \| & \| & & \| \\ & p_{j-k-1} & p_{j-k} & \cdots & p_{j-1} \end{array}$$

下一趟可以直接用 p_{k+1} 与 S_{s+j} 继续比较。

改进算法的匹配过程示例

i=7 j=2

i	1	2	3	4	5	6	7	8	9	10	11	12	13
S	a	b	a	b	c	a	b	c	a	c	b	a	b

j	1	2	3	4	5
P	a	b	c	a	c

KMP算法就是一种基于分析模式串P蕴含信息的改进算法。

END