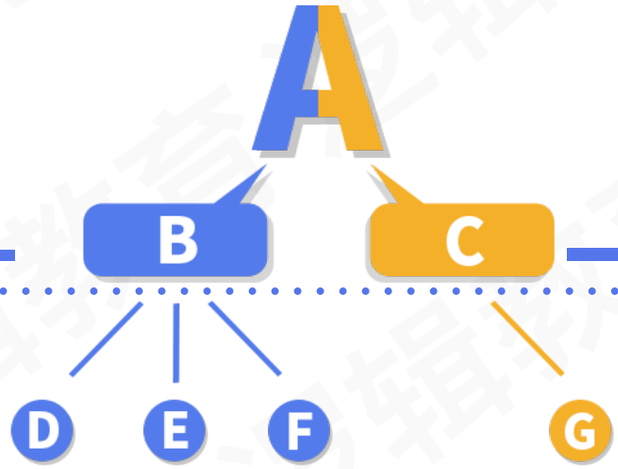
逻辑教育
Logic education

Hello 数据结构与算法



数据结构与算法—串的理解与应用

数据结构与算法主题

@CC老师

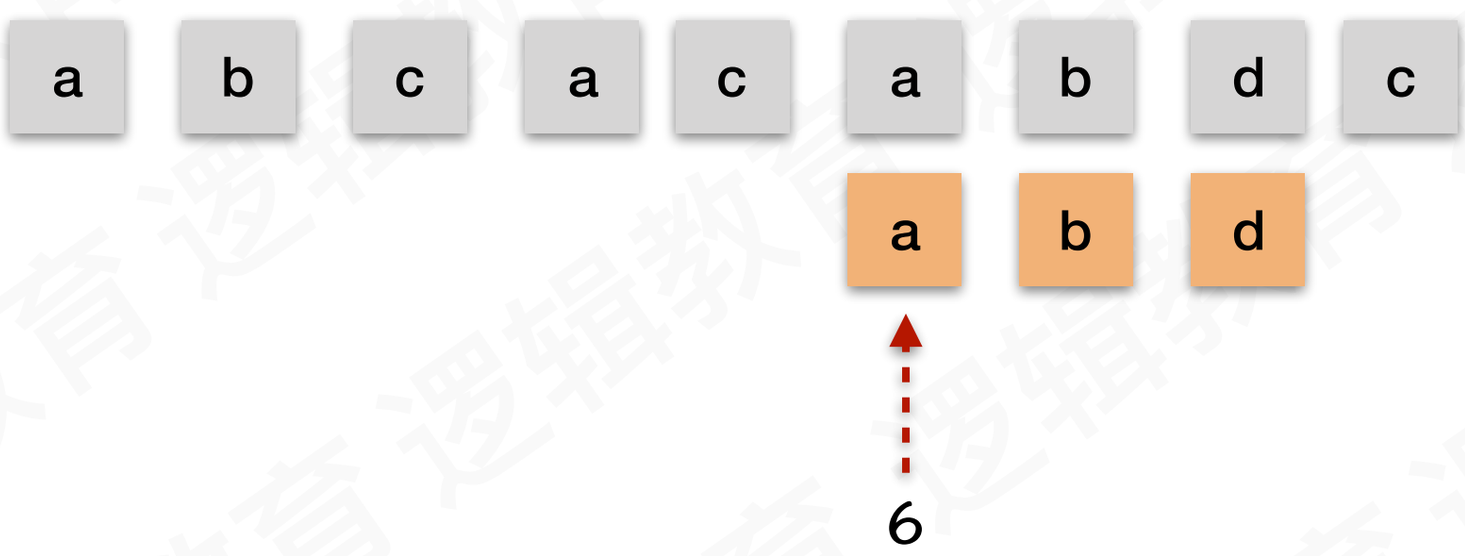
全力以赴．非同凡"想"

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

字符串匹配

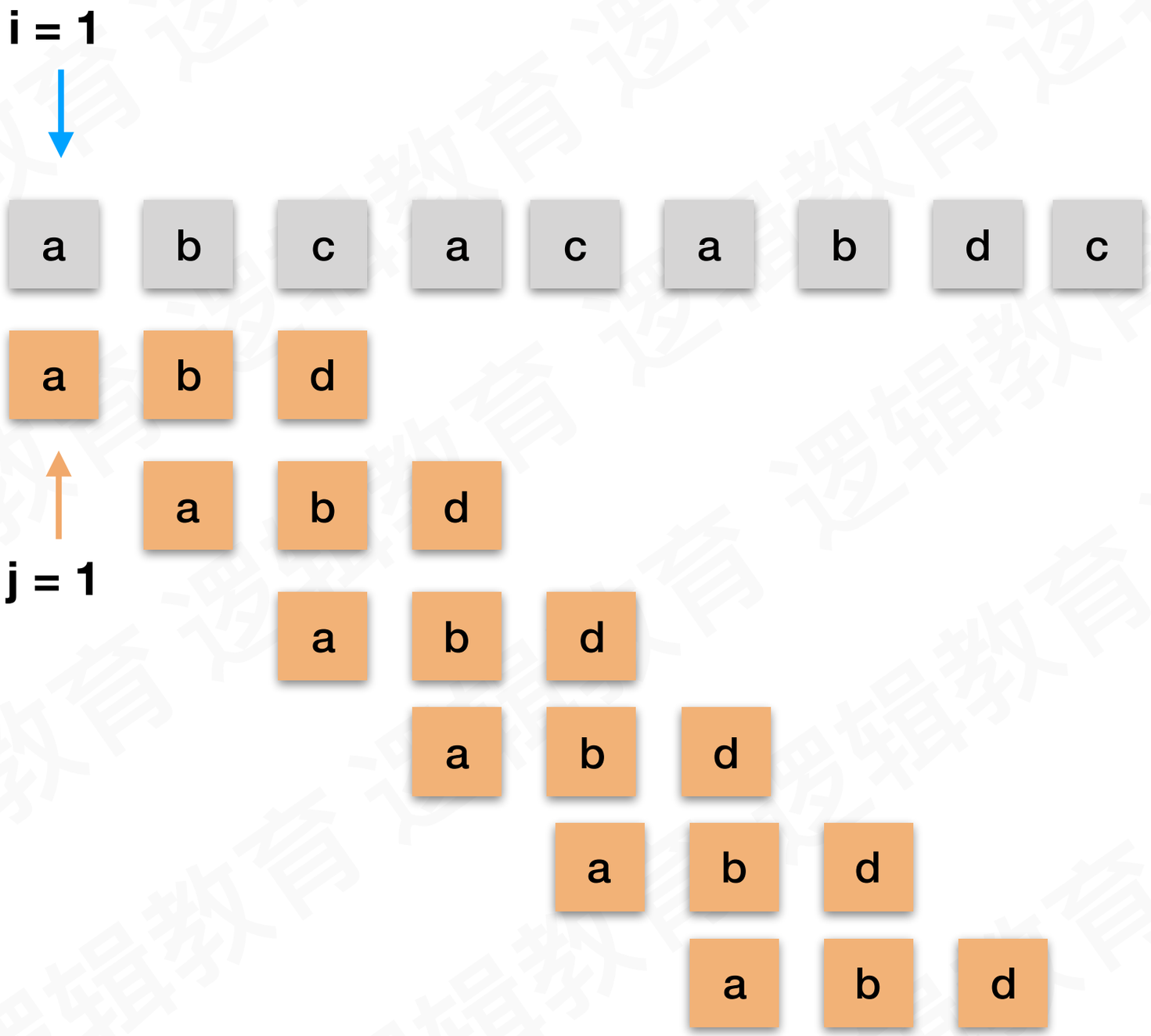
题目: 有一个主串S = {a, b, c, a, c, a, b, d, c}, 模式串T = { a, b, d }; 请找到模式串在主串中第一次出现的位置;
提示: 不需要考虑字符串大小写问题, 字符均为小写字母;



课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

BF 算法

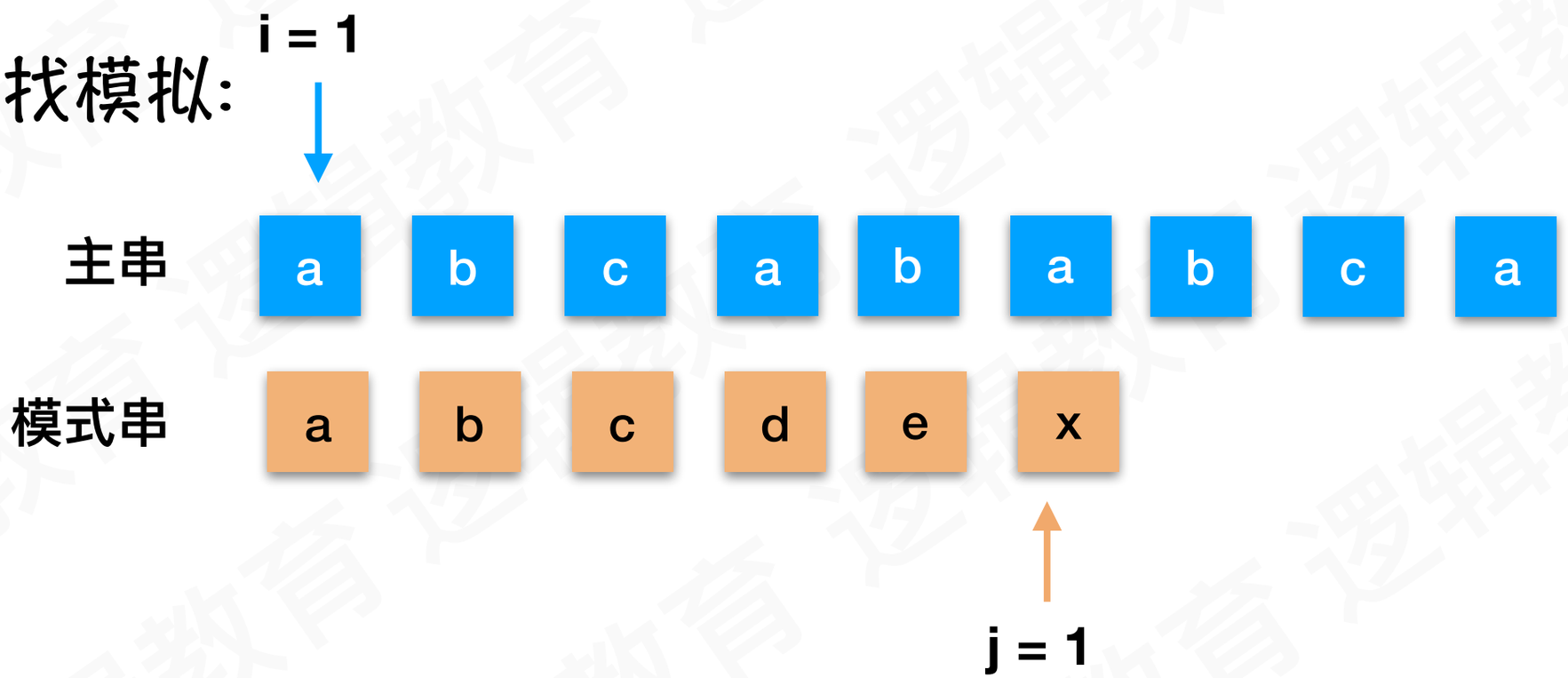


课程研发:CC老师
课程授课:CC老师

BF 算法

假设，主串S = “abcababca”；模式串 T = “abcdex”

BF算法查找模拟：

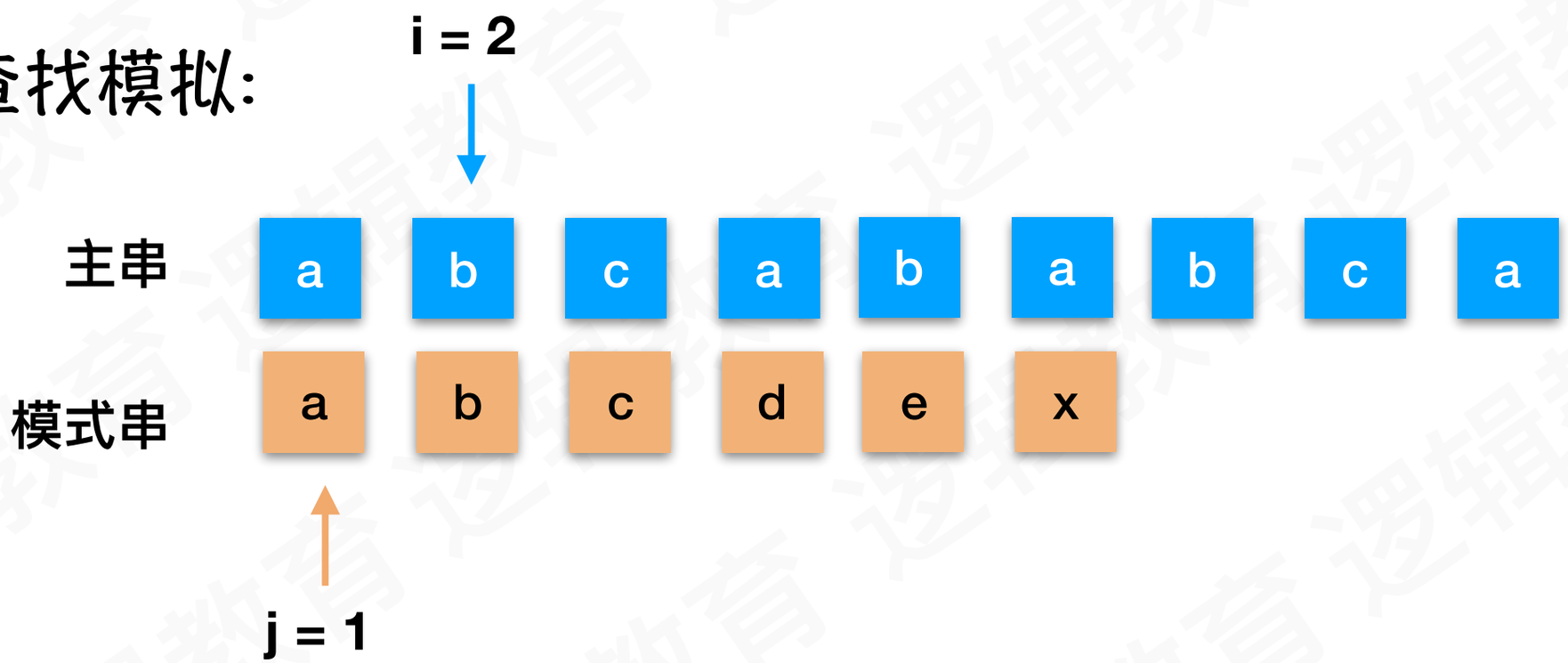


课程研发:CC老师
课程授课:CC老师

BF 算法

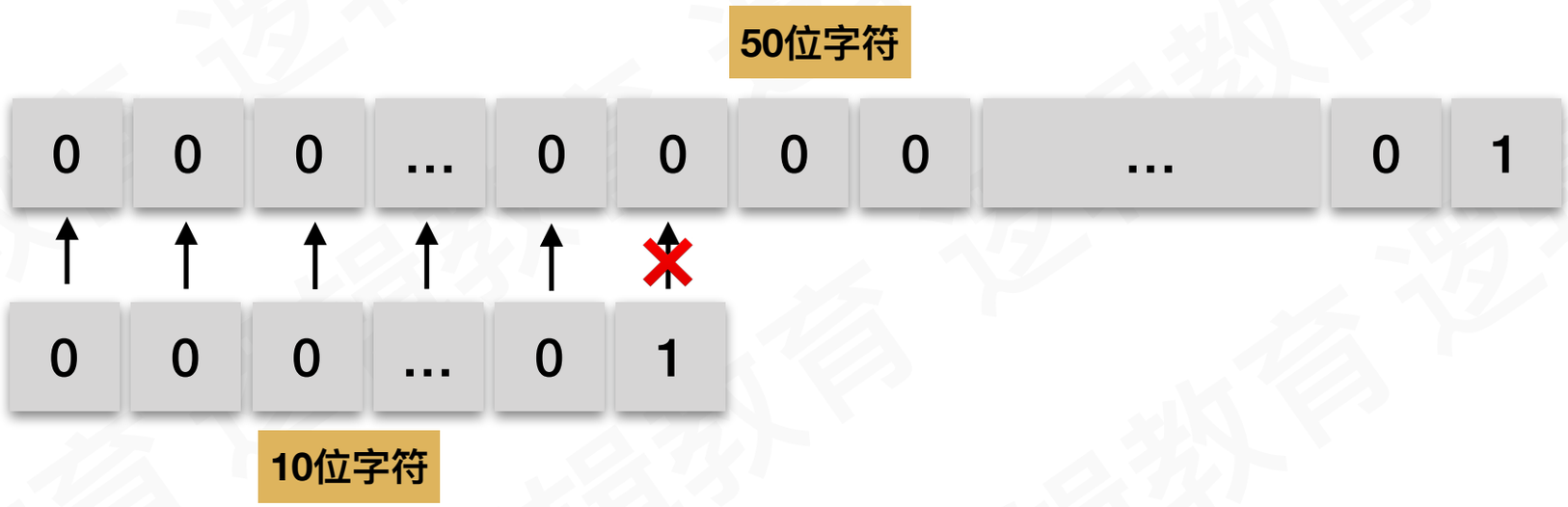
假设，主串S = “abcababca”；模式串 T = “abcdex”

BF算法查找模拟：



课程研发:CC老师
课程授课:CC老师

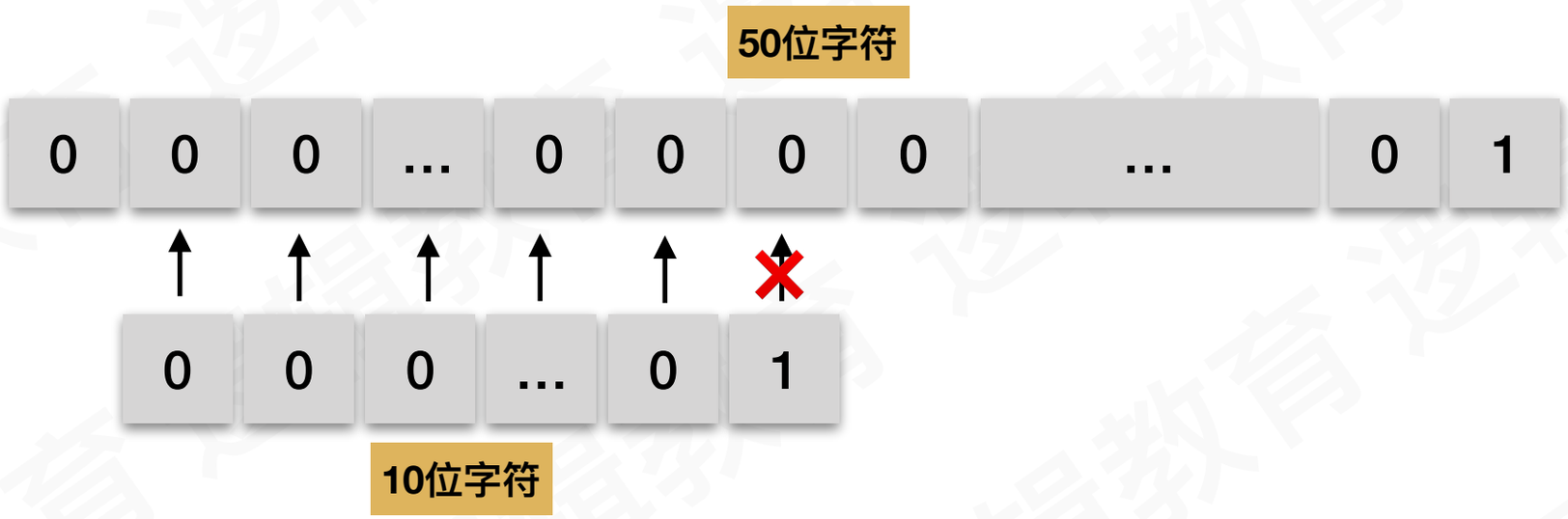
BF 算法效率分析



T在第一个位置判断了10次发现字符串不匹配

课程研发:CC老师
课程授课:CC老师

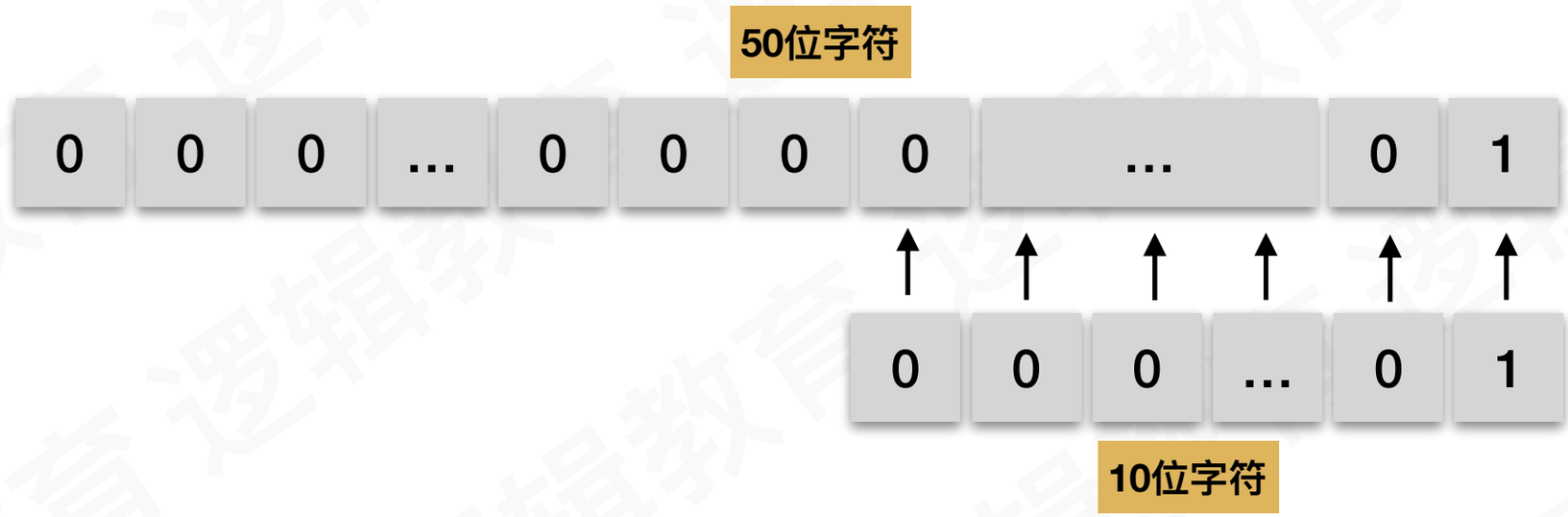
BF 算法效率分析



T在第二个位置判断了10次发现字符串不匹配

课程研发:CC老师
课程授课:CC老师

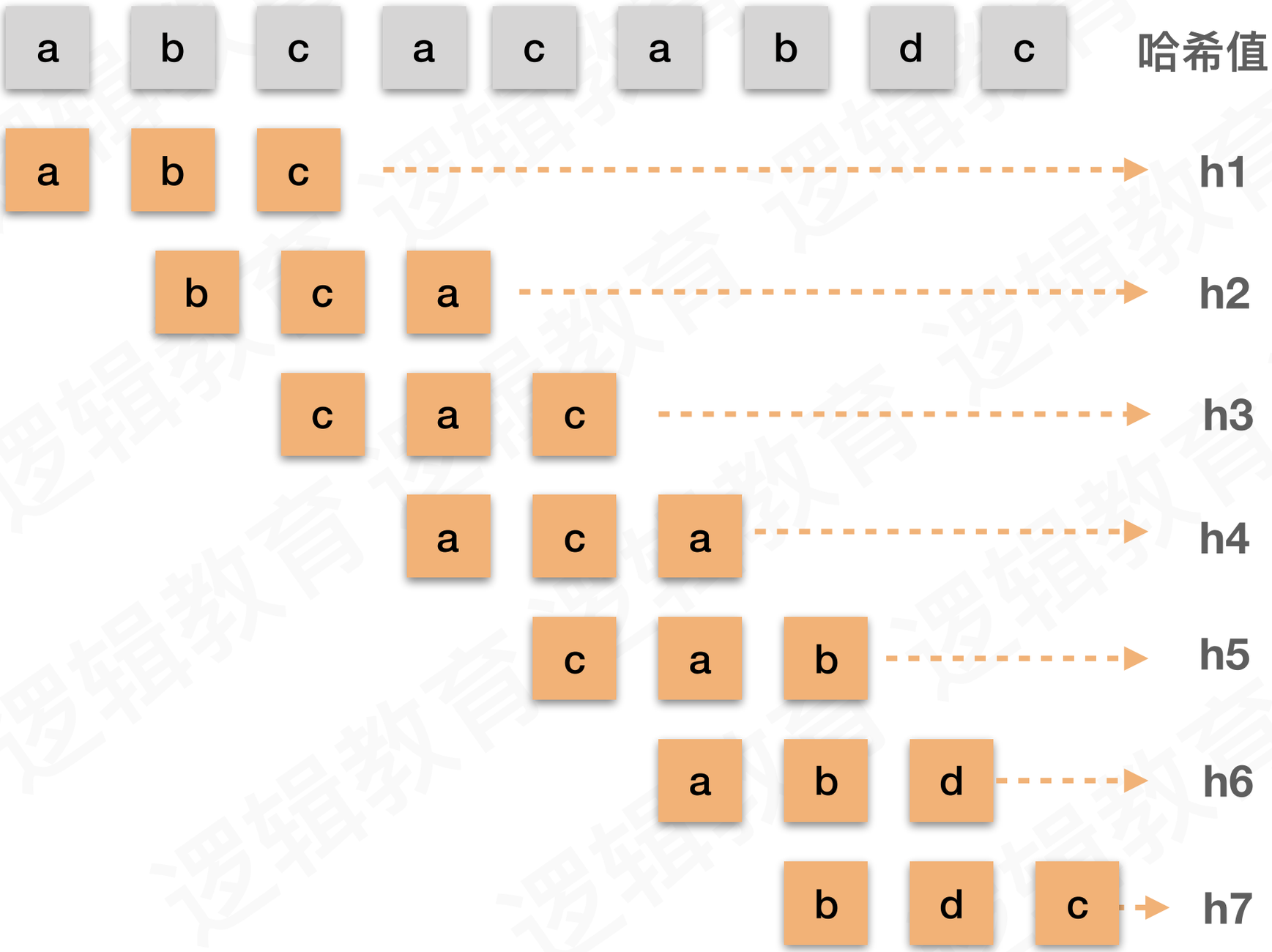
BF 算法效率分析



T在第41个位置判断了10次发现字符串终于匹配成功
期间进行了(50-10+1)*10次判断操作

课程研发:CC老师
课程授课:CC老师

RK 算法核心思想



课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

RK 算法核心思想

如何将模式串或者主串拆分后的子串换算成一个哈希值?

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

Hash (哈希算法)

Hash (哈希). 一般中文也翻译做”散列”; 也可以直接音译”哈希”;
散列在开发中是常见手段! 比如大家常用的MD5 算法就是哈希算法;
哈希算法在安全方面应用是非常多,一般体现在如下这几个方面:

1. 文件校验
2. 数字签名
3. 鉴权协议

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

Hash (哈希算法)

将不同的字符组合能够通过某种公式的计算映射成不同的数字!

例如

比较“abc”与“cde”；比较 123 与 456; 是一样的吗?

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

RK 算法核心思想

$$657 = 6 * 10 * 10 + 5 * 10 + 7 * 1$$

$$657 = 6 \times 10^2 + 5 \times 10^1 + 7 \times 10^0$$

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

RK 算法核心思想

字母换算成哈希值

课程研发:CC老师
课程授课:CC老师

RK 算法核心思想

将’当前字母’ - ‘a’ = 数字

例如

a - a = 0;

b - a = 1;

c - a = 2;

d - a = 3;

e - a = 4;

...

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

RK 算法核心思想

小写字母之间存在的进制

课程研发:CC老师
课程授课:CC老师

RK 算法核心思想

“ cba ” = ‘c’ * 26 *26 + ‘b’ * 26 + ‘a’ * 1

= 2 * 26 * 26 + 1* 26 + 0 * 1

= 1378

“ cba ” = $c \times 26^2 + b \times 26^1 + a \times 26^0$

= $2 \times 26^2 + 1 \times 26^1 + 0 \times 26^0$

= 1352 + 26 + 0

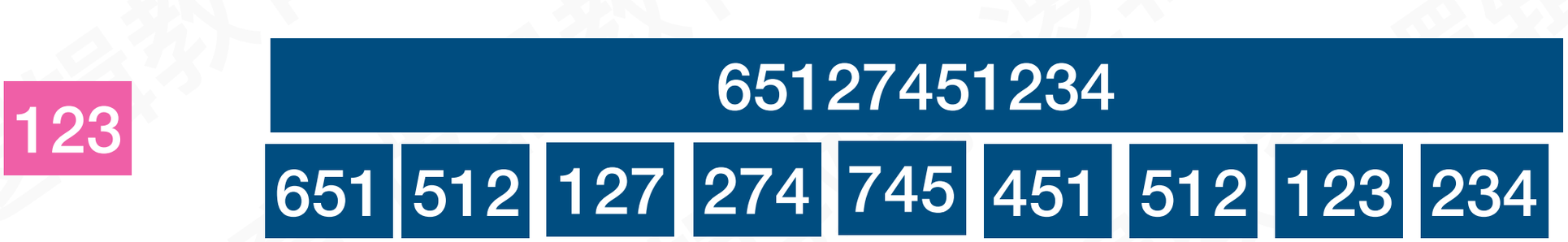
= 1378

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

RK 算法核心思想

为了让大家知道接下来推演过程，以数字为例,会更容易让大家理解，它的全集是 {0,1,2,3,4,5,6,7,8,9}. d = 10; 模式串p = 123, 主串s = 6512745123 4



求解 子串A的哈希值 $= 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$. 代码实现!

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

RK 算法核心思想

主串拆解的子串与模式串的哈希值比较?

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

RK 算法核心思想

主串: d b c e d b

主串: d b c e d b

模式串: c c c

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

RK 算法核心思想

主串: d b c e d b

$= 3 \times 26^2 + 1 \times 26^1 + 2 \times 26^0$

主串: d b c e d b

$= 1 \times 26^2 + 2 \times 26^1 + 4 \times 26^0$

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

RK 算法核心思想

子串哈希值求解规律:

相邻的2个子串 s[i] 与 s[i+1] (i表示子串从主串中的起始位置,子串的长度都为m). 对应的哈希值计算公式有交集. 也就说我们可以使用s[i-1]计算出s[i]的哈希值;

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

RK 算法核心思想

以数字为例，它的全集是 {0,1,2,3,4,5,6,7,8,9}. d = 10;
模式串p = 123, 主串s = 6512745123 4

123

6512745123 4									
651	512	127	274	745	451	512	123	234	

$$s[i] = 1 \times 10^2 + 2 \times 10^1 + 7 \times 10^0$$

274

$$s[i+1] = 2 \times 10^2 + 7 \times 10^1 + 4 \times 10^0$$

课程研发:CC老师
课程授课:CC老师

RK 算法核心思想

$$s[i] = 1 \times 10^2 + 2 \times 10^1 + 7 \times 10^0$$

$$s[i+1] = 2 \times 10^2 + 7 \times 10^1 + 4 \times 10^0$$

$$s[i+1] = 10 \times (127 - 1 \times 10^2) + 4$$

$$s[i+1] = 10 \times (s[i] - 1 \times 10^2) + 4$$

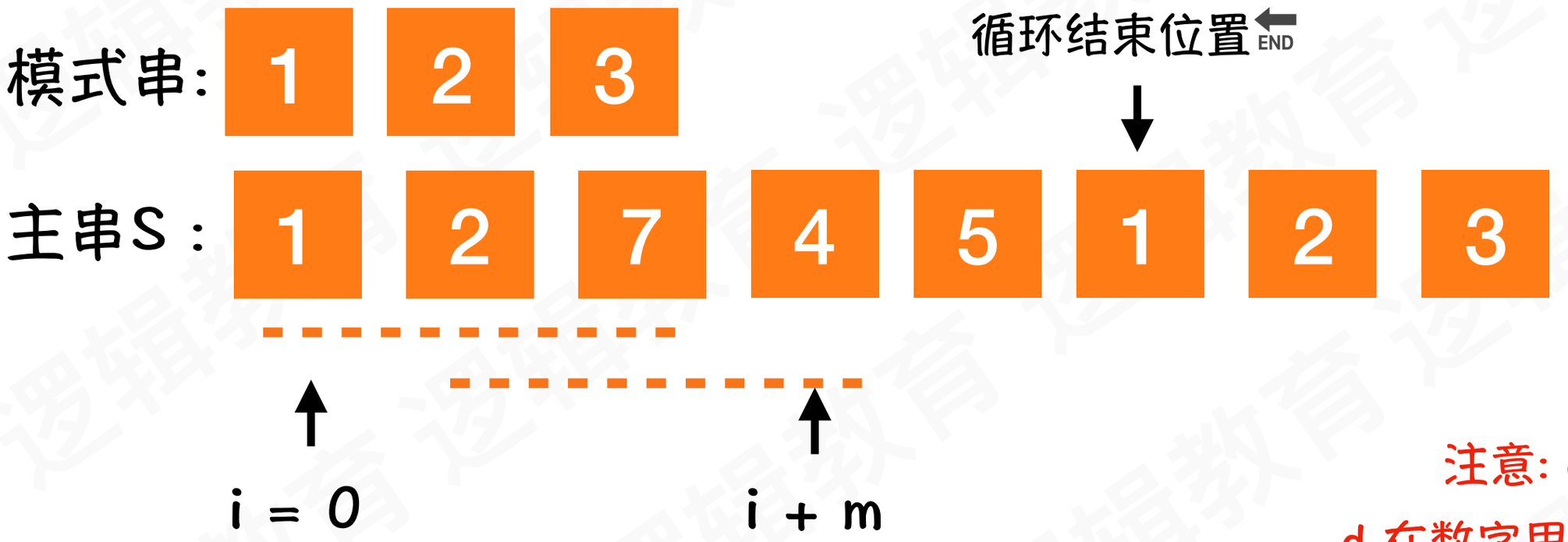
s[i+1] 实现上是上一个s[i]去掉最高位数据,其余的m-1为字符乘以d进制. 再加上最后一个为字符得到;

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

RK 算法核心思想

m = 3; n = 8



注意: d 指的是进制!

d 在数字里指的是十进制

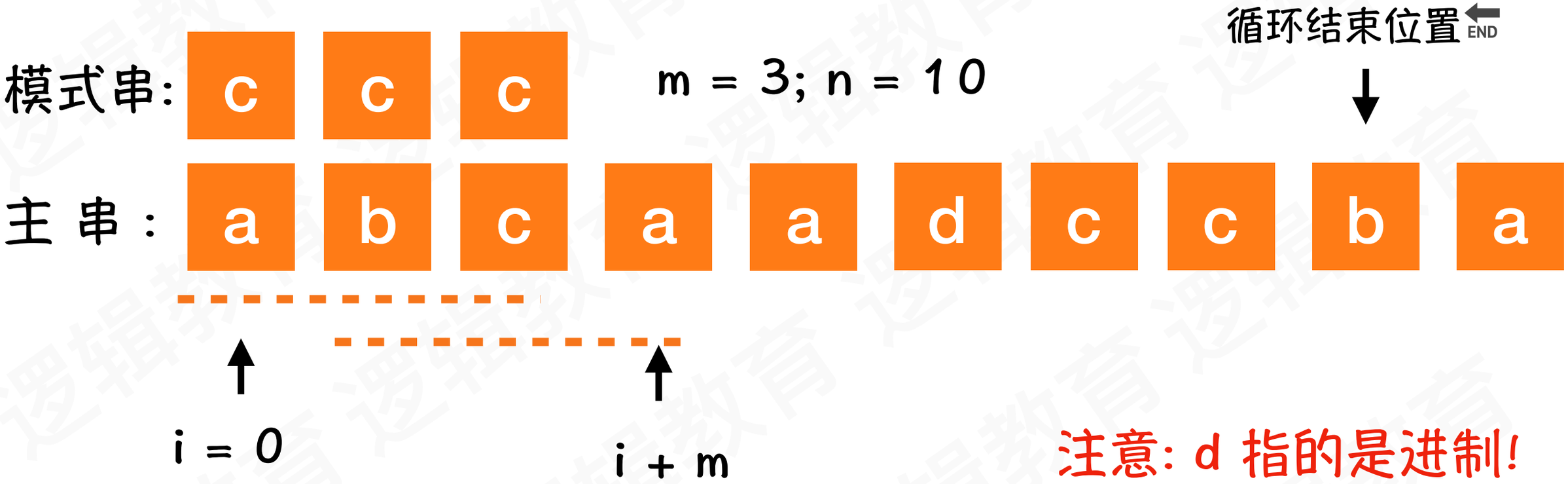
$$274 = (127 - 1 \times d^2) \times d + 4$$

主串分割后的哈希值:$St[i+1] = (st[i] - d^2 \times s[i]) \times d + s[i+m]$

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

RK 算法核心思想



注意: d 指的是进制!

$bca = (abc - a \times d^2) \times d + a$

$St[i] = (st[i-1] - d^2 \times (s[i] - 'a') \times d + (s[i+m] - 'a'))$

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

RK 算法核心思想

请用代码实现 刚刚关于主串与子串哈希值计算与比较的代码!

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

RK 算法核心思想

哈希冲突解决方案!

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

RK 算法核心思想

哈希冲突解决方案!

1. 设计更复杂的哈希公式; (后面课程会讲解关于哈希值的相关课程)

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

RK 算法核心思想

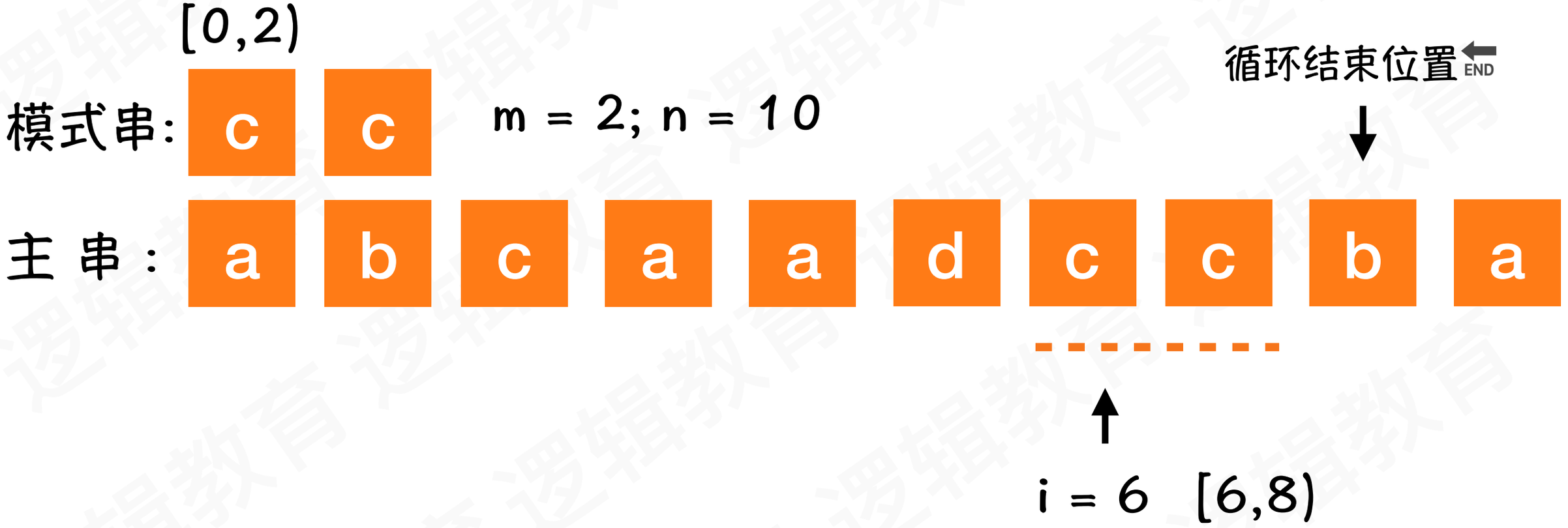
哈希冲突解决方案!

2. 如果相等时,不要直接返回结果. 而是重新核实!

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

RK 算法核心思想



请用代码实现 刚刚关于主串[i,i+m) 的哈希值与子串(0,m)哈希值相等时,二次核实函数!

课程研发:CC老师
课程授课:CC老师

RK 算法核心思想

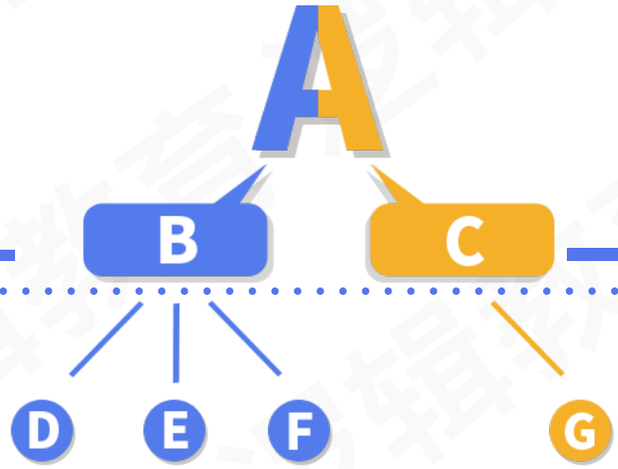
1. 如果不做哈希冲突二次核查 比较次数是n-m+1次; 那么时间复杂度为O(n);
2. 但是要想解决冲突存在可能性.就需要添加二次核查! 那么就需要m次比对; 那么时间复杂度为O(n*m);

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

Hello 数据结构与算法

数据结构与算法—KMP模式匹配算法



数据结构与算法主题

@CC老师

全力以赴．非同凡"想"

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

KMP 模式匹配算法

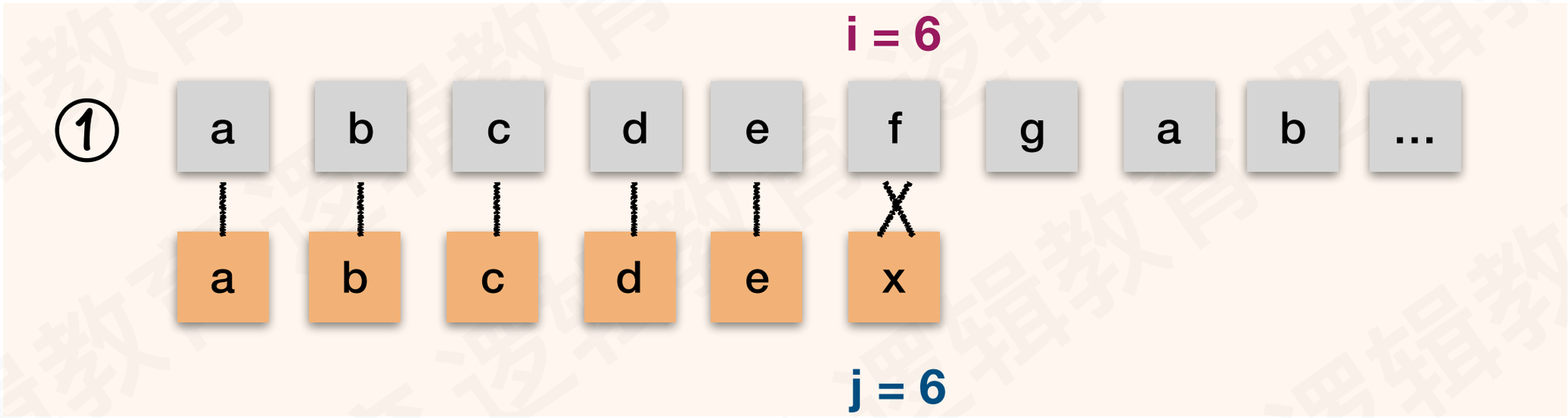
D.E.Knuth, J.H.Morris 和 V.R.Pratt 共同发表模式匹配算法，称之克鲁特-莫里斯-普拉特算法. 简称 KMP 算法.

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

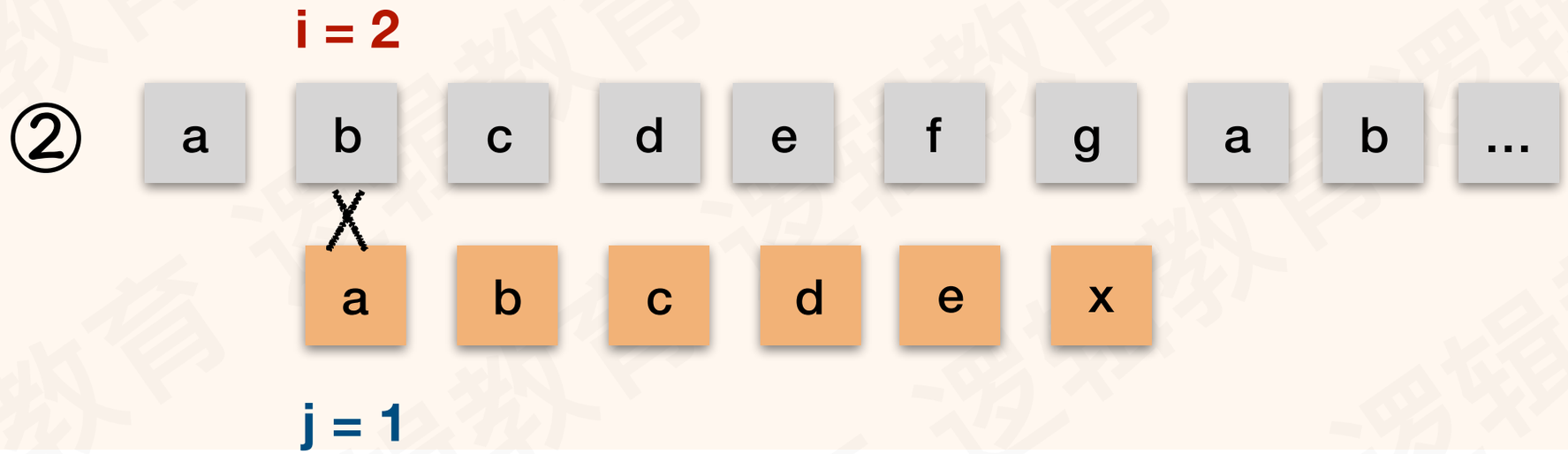
KMP 模式匹配算法原理探索

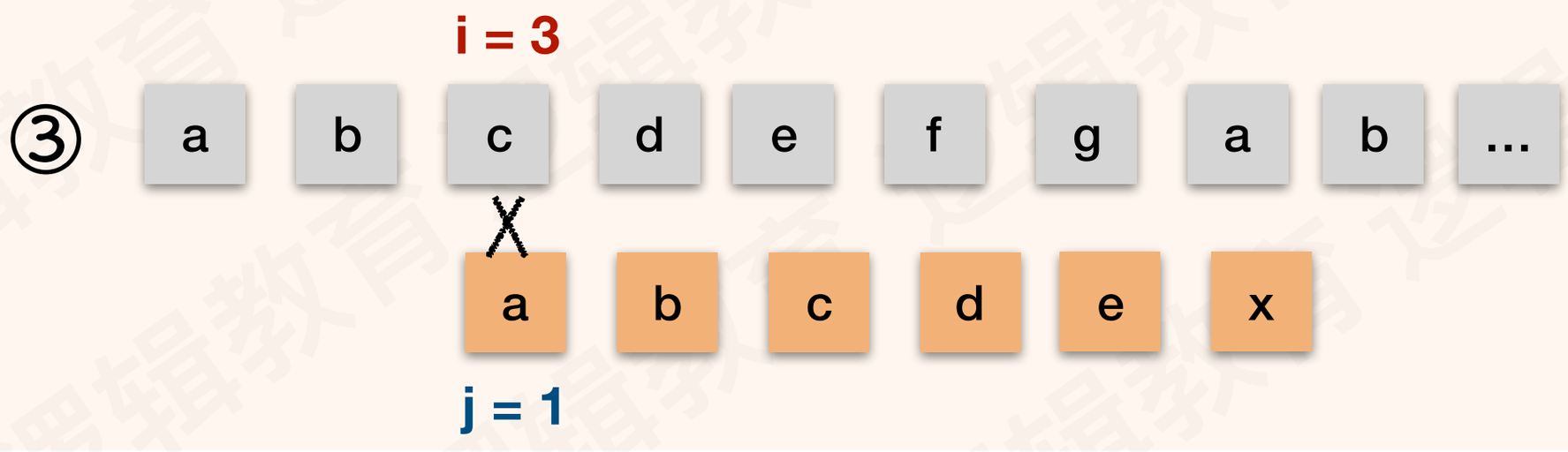
假设，主串S = “abcdefgab”；模式串 T = “abcdex”



课程研发:CC老师
课程授课:CC老师

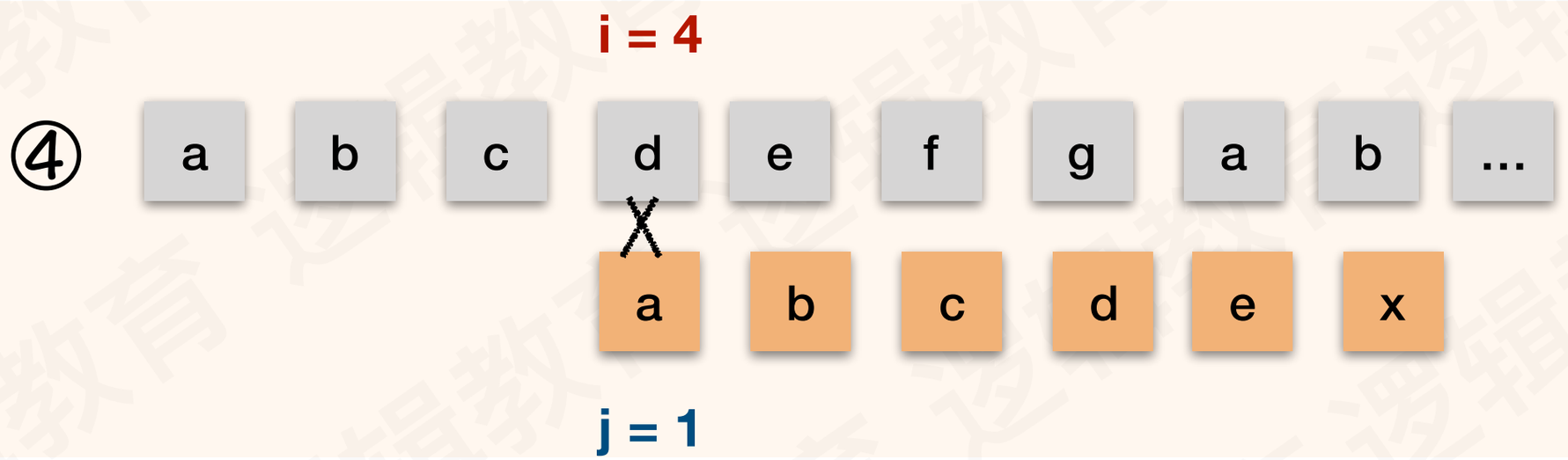
KMP 模式匹配算法原理探索

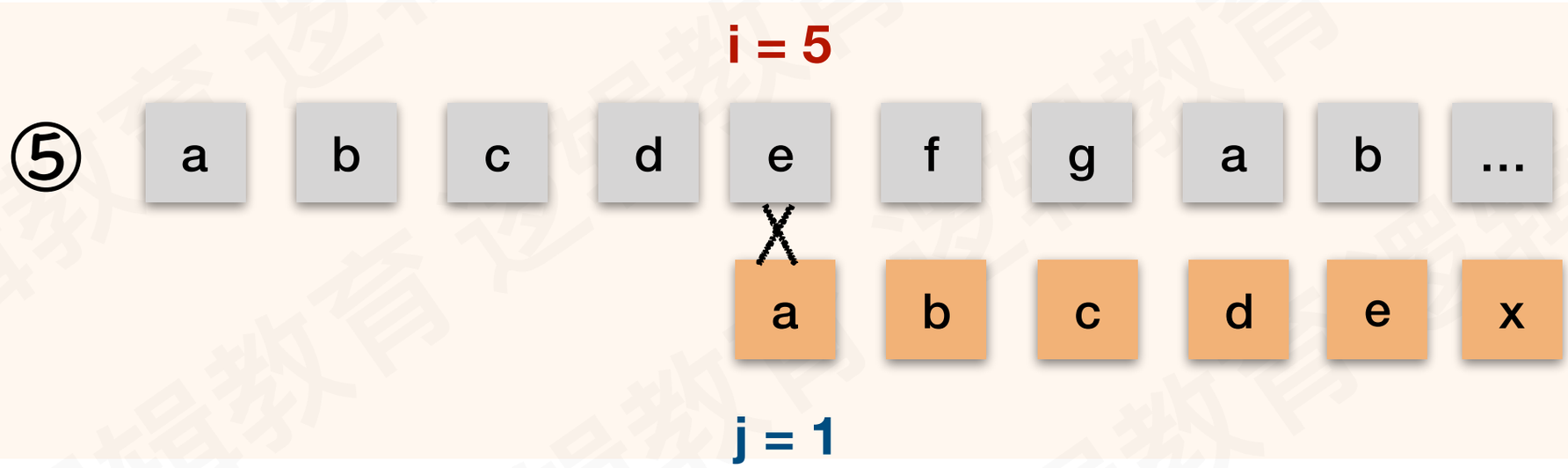




课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法原理探索

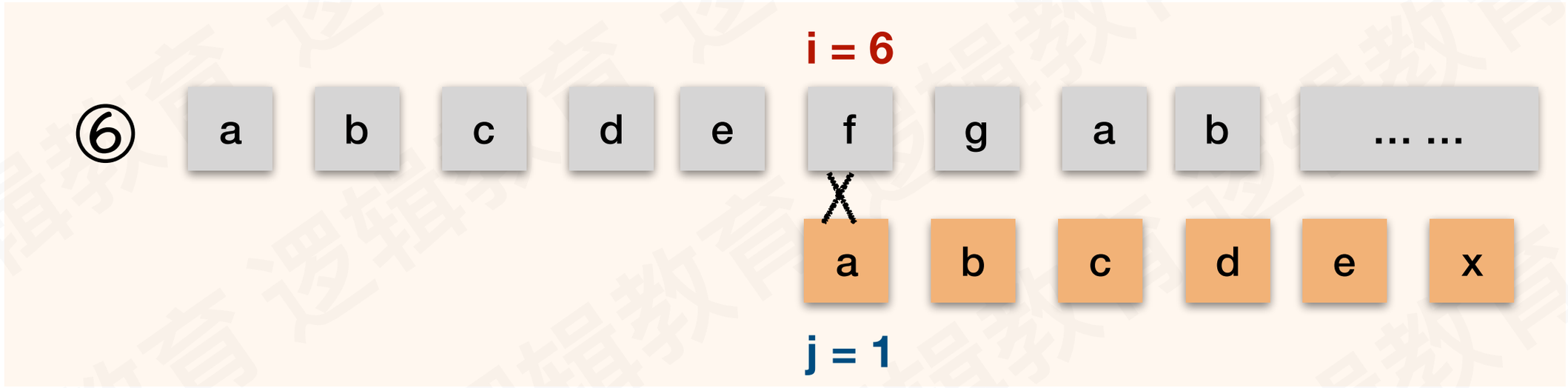




课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

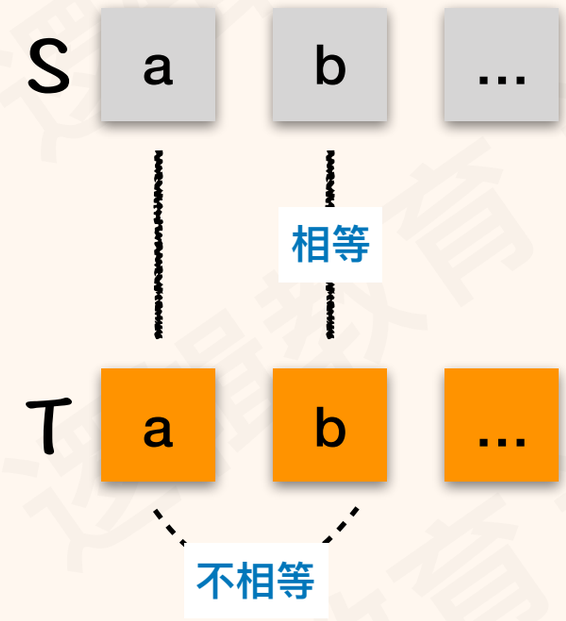
KMP 模式匹配算法原理探索



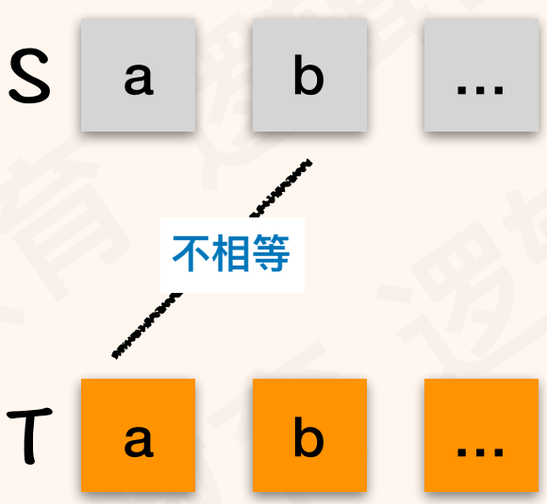
课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

KMP 模式匹配算法原理探索



T[1] = a; T[2] = b; S[2] = b;
显然 T[1] != T[2], T[2] = S[2]



因此 T[1] != S[2]

多余比较!

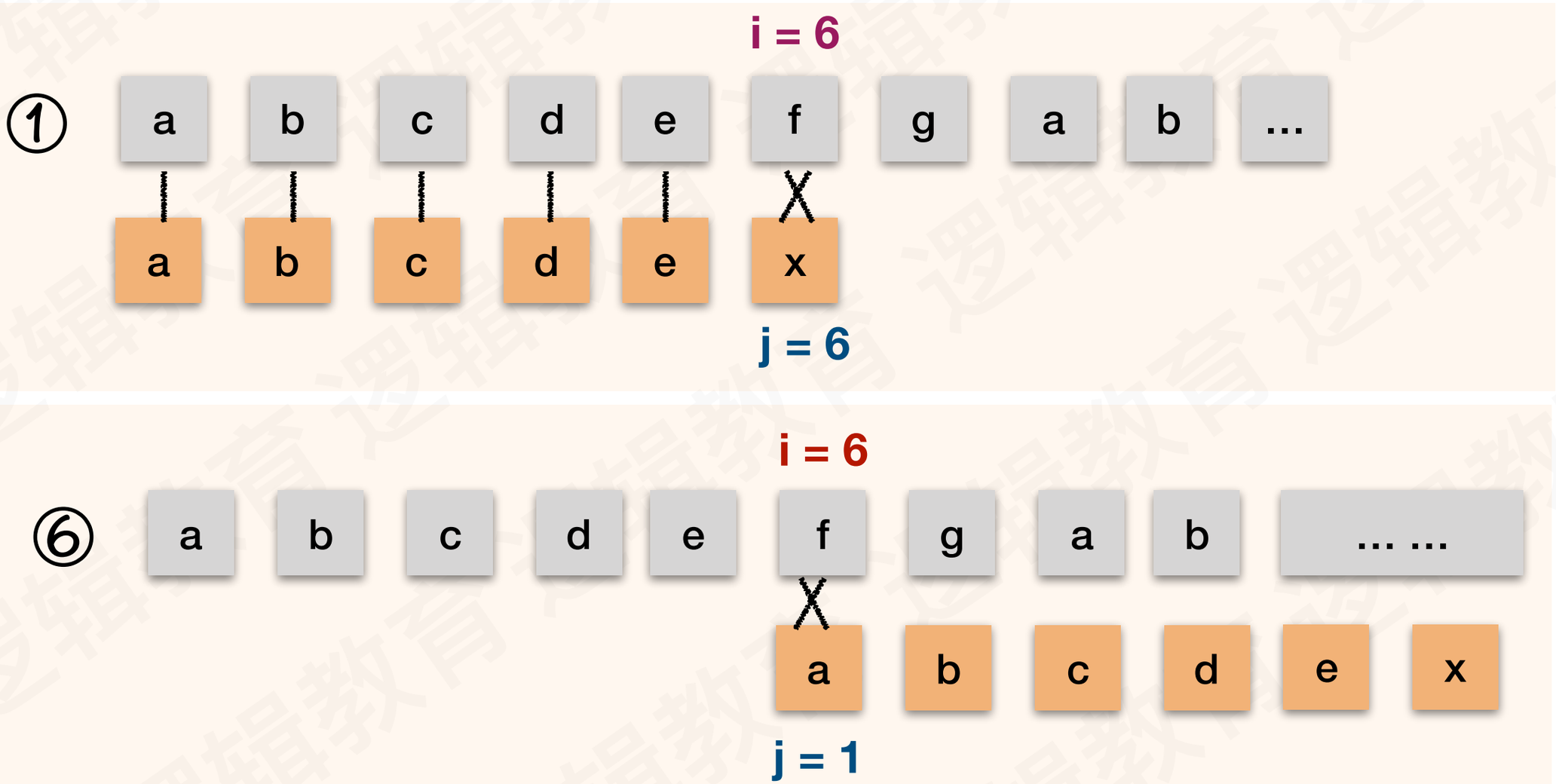
因此判断T 串第二位置的
判断就根本不需要进行

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

KMP 模式匹配算法原理探索

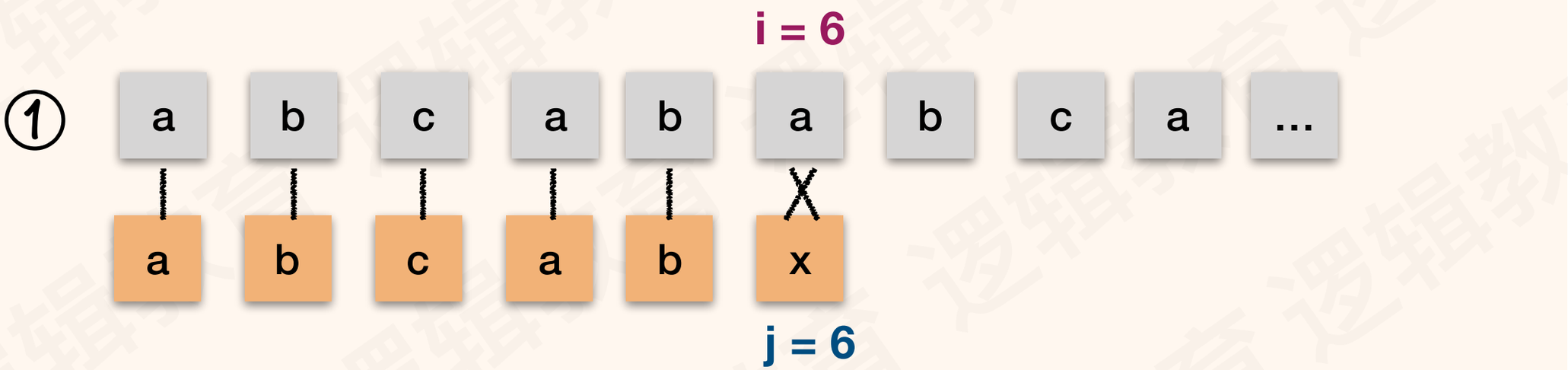
假设，主串S = “abcdefgab”；模式串 T = “abcdex”



课程研发:CC老师
课程授课:CC老师

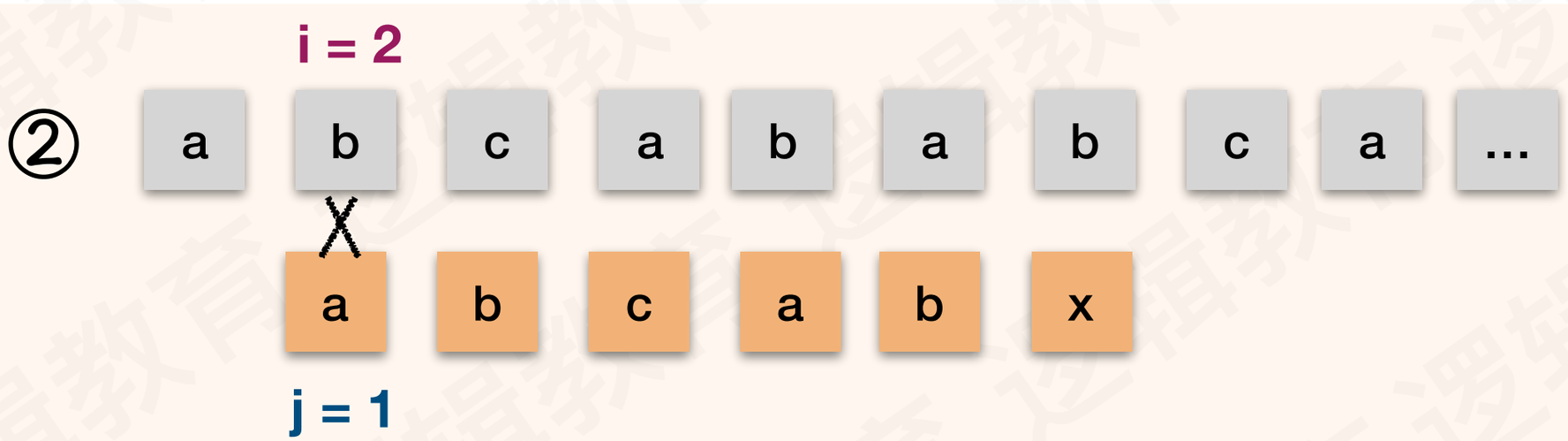
KMP 模式匹配算法原理探索

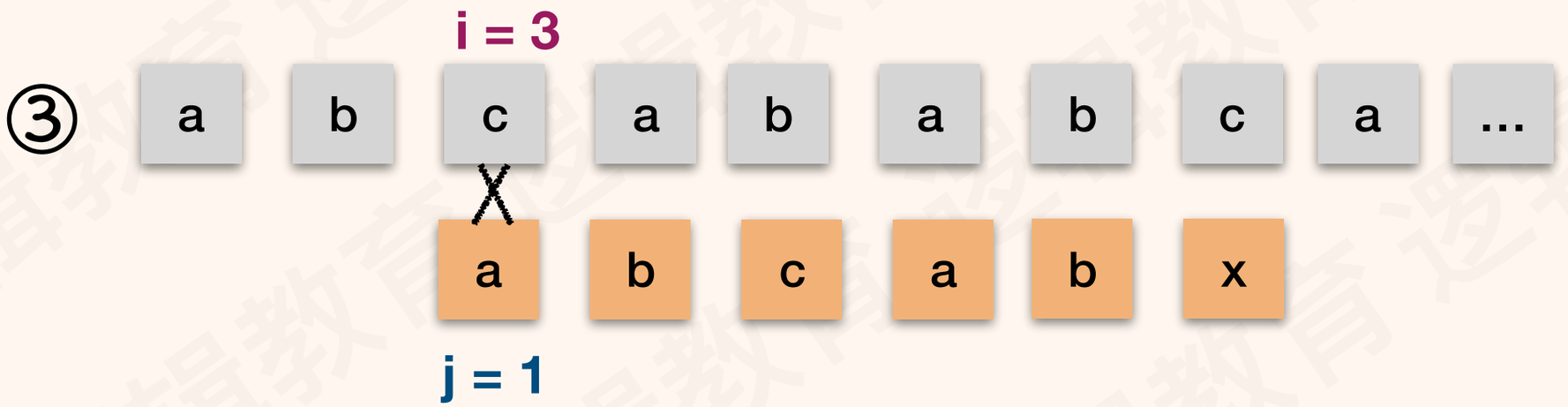
假设，主串S = “abcababca”；模式串 T = “abcabx”



课程研发:CC老师
课程授课:CC老师

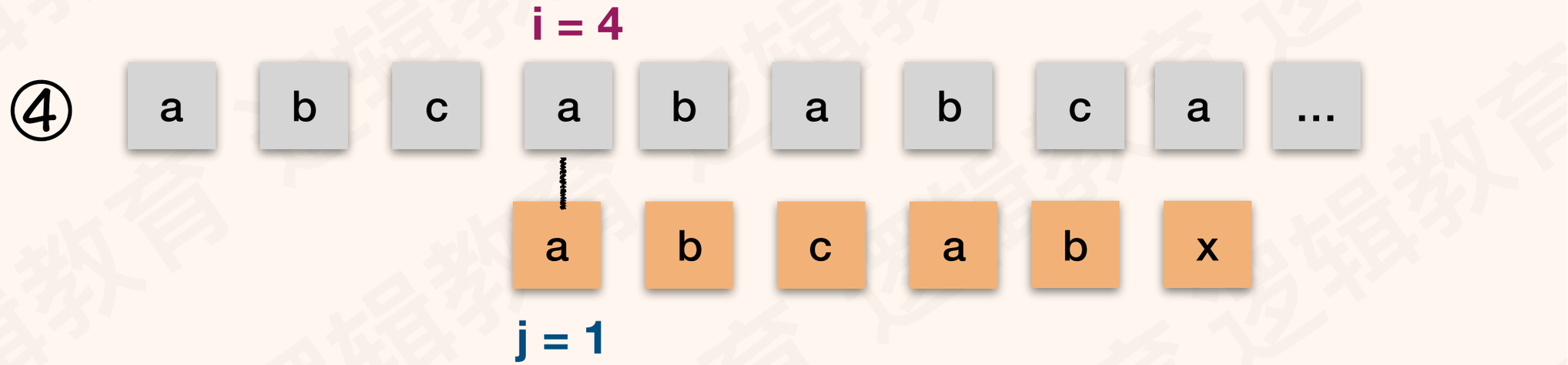
KMP 模式匹配算法原理探索

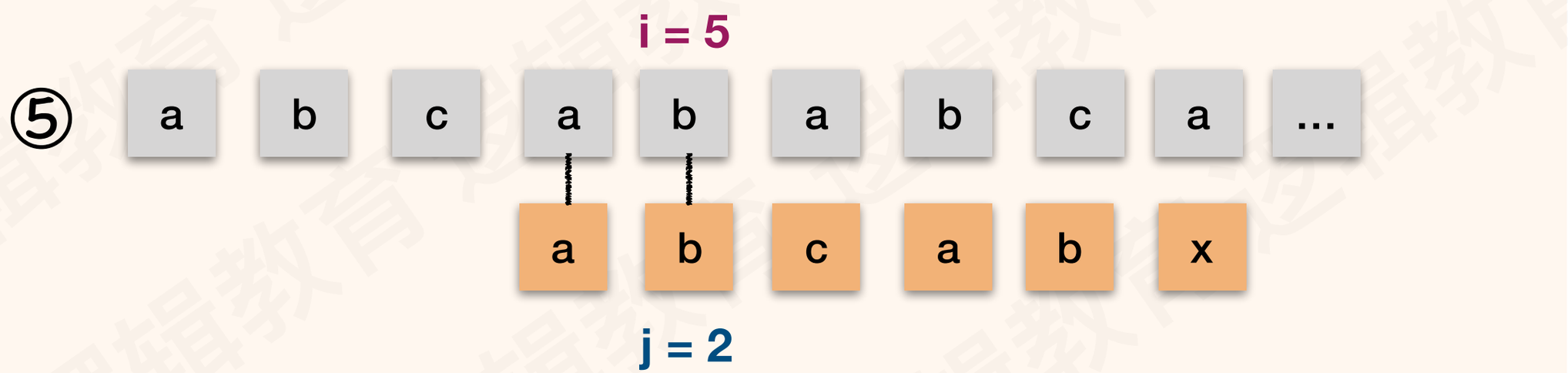




课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法原理探索

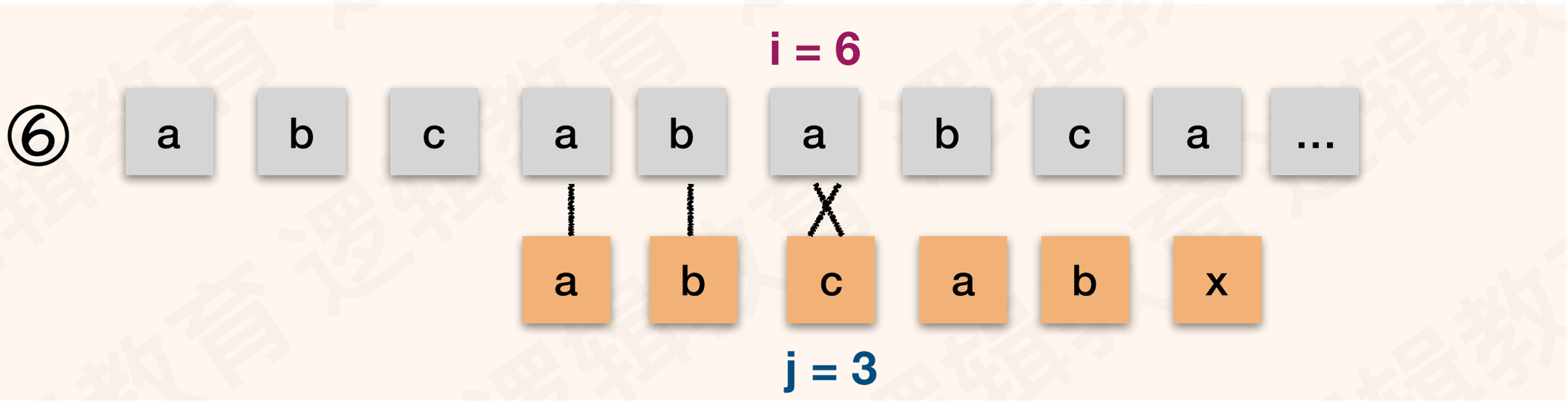




课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

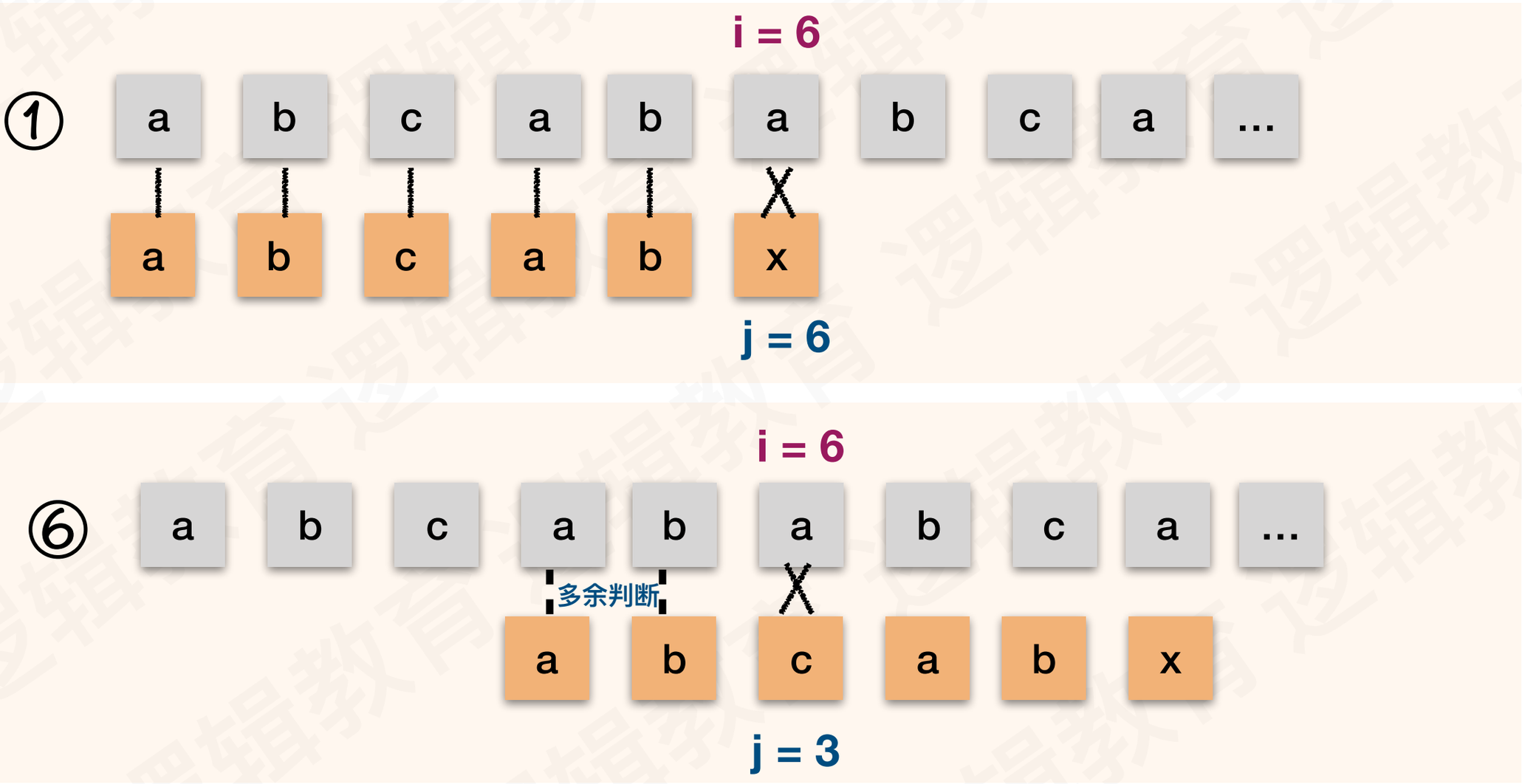
KMP 模式匹配算法原理探索



课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法原理探索

假设，主串S = “abcababca”；模式串 T = “abcabx”



课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

KMP 模式匹配算法_next 数组值推导

我们把 T 串各个位置 j 值变化定义为一个 next 数组; 那么 next 的长度就是 T串的长度; 于是我们可以得出下面函数的定义:

$$next[j] = \begin{cases} 0, 当 j = 1 时 \\ Max\{k \mid 1<k<j, 且'p_1 \cdots p_{k-1}' = 'p_{j-k+1} \cdots p_{j-1}'\} 当此集合不为空时 \\ 1, 其他情况 \end{cases}$$

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

KMP 模式匹配算法_next 数组值推导

T = “abcdex”

j	123456
模式串T	abcdex
next[j]	011111

- 当 j=1时, next[1] = 0
- 当 j=2时, j 由 1到 j-1 范围内只有字符 “a”, 属于其他情况 next[2] = 1;
- 当 j=3时, j 由 1到 j-1 范围内有字符 “ab”,显然 a 不等于 b, 属于其他情况 next[3] = 1;
- 当 j=4时, j 由 1到 j-1 范围内有字符”abc”,显然abc 不存在相等情况,则属于其他情况 next[4] = 1;
- 当 j=5时, j 由 1到 j-1 范围内有字符”abcd”,显然abcd 不存在相等情况,则属于其他情况 next[5] = 1;
- 当 j=6时, j 由 1到 j-1 范围内有字符”abcde”,显然abcde 不存在相等情况,则属于其他情况next[6] = 1;

逻辑教育
Logic education

KMP 模式匹配算法_next 数组值推导

T = “abcabx”

j	123456
模式串T	abcabx
next[j]	011123

解读:

- 当 j=1时, next[1] = 0
- 当 j=2时, j 由 1到 j-1 范围内只有字符 “a”, 属于其他情况 next[2] = 1;
- 当 j=3时, j 由 1到 j-1 范围内有字符 “ab”,显然 a 不等于 b, 属于其他情况 next[3] = 1;
- 当 j=4时, j 由 1到 j-1 范围内有字符”abc”,显然abc 不存在相等情况,则属于其他情况 next[4] = 1;

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

KMP 模式匹配算法_next 数组值推导

T = “abcabx”

j	123456
模式串T	abcabx
next[j]	011123

j = 5的情况
主串S: abcad
模式串: abcab

j = 6的情况
主串S: abcabc
模式串: abcabx

解读:

- 当 j=5时, j 由 1到 j-1 范围内有字符”abca”,显然 abca 前缀字符 “a” 与 后缀字符 “a” 相等 ; (由于 ’$p_1 \cdots p_{k-1}$’ = ‘ $p_{j-k+1} \cdots p_{j-1}$’,得到$p_1 = p_4$) 因此可以推算出 k 值为2; 因此 next[5] = 2;

- 当 j=6时, j 由 1到 j-1 范围内有字符”abcab”,显然 abcab 前缀字符 “ab” 与 后缀字符 “ab” 相等; (由于 ’$p_1 \cdots p_{k-1}$’ = ‘ $p_{j-k+1} \cdots p_{j-1}$’,得到 $[p_1, p_{3-1}] = [p_{6-3+1}, p_5]$) 推导 k 值为 3, 因此next[6] = 3;

经验: 如果前后缀一个字符相等,K值是2; 两个字符相等是3; n个相等k值就是n+1;

逻辑教育
Logic education

KMP 模式匹配算法_next 数组值推导

T = “abababaaaba”

j	123456789
模式串T	abababaaaba
next[j]	011234223

j = 4的情况
主串S: abadd
模式串: ababa

j = 5的情况
主串S: ababe
模式串: ababa

解读:

- 当 j=1时, next[1] = 0
- 当 j=2时, j 由 1到 j-1 范围内只有字符 “a”, 属于其他情况 next[2] = 1;
- 当 j=3时, j 由 1到 j-1 范围内有字符 “ab”,显然 a 不等于 b, 属于其他情况 next[3] = 1;
- 当 j=4时, j 由 1到 j-1 范围内有字符 ”aba”, 显然”aba”, 前缀字符 “a” 与 后缀字符 ”a” 相等,所以 k = 2; next[4] = 2;
- 当 j=5时, j 由 1到 j-1 范围内有字符 ”abab”, 显然”abab”, 前缀字符 “ab” 与 后缀字符 ”ab” 相等,所以 k = 3; next[5] = 3;

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

KMP 模式匹配算法_next 数组值推导

T = “abababaaaba”

j	123456789
模式串T	abababaaaba
next[j]	011234223

j = 6的情况
主串S: ababag
模式串: ababaa

j = 7的情况
主串S: ababaax
模式串: ababaaa

j = 8的情况
主串S: ababaaak
模式串: ababaaab

j = 9的情况
主串S: ababaaay
模式串: ababaaab

解读:

- 当 j=6时, j 由 1到 j-1 范围内有字符 “ababa”,前缀 “aba” 与 后缀 “aba” 相等,那么此时 k = 4; next[6] = 4;
- 当 j=7时, j 由 1到 j-1 范围内有字符 “ababaa”,前缀 “a” 与 后缀 “a” 相等,那么此时 k = 2; next[7] = 2;
- 当 j=8时, j 由 1到 j-1 范围内有字符 “abababaa”,前缀 “a” 与 后缀 “a” 相等,那么此时 k = 2; next[8] = 2;
- 当 j=9时, j 由 1到 j-1 范围内有字符 “abababaaab”,前缀 “ab” 与 后缀 “ab” 相等,那么此时 k = 3; next[9] = 3;

KMP 模式匹配算法_next 数组值推导

T = “aaaaaaaaab”

j	123456789
模式串T	aaaaaaaab
next[j]	012345678

解读:

- 当 j=1时, next[1] = 0
- 当 j=2时, j 由 1到 j-1 范围内只有字符 “a”, 属于其他情况 next[2] = 1;
- 当 j=3时, j 由 1到 j-1 范围内有字符 “aa”, 前缀 ”a” 与 后缀“a” 相等,所以k = 2; next[3] = 2;
- 当 j=4时, j 由 1到 j-1 范围内有字符 “aaa”, 前缀 ”aa” 与 后缀“aa” 相等,所以k = 3; next[4] = 3;
- 当 j=5时, j 由 1到 j-1 范围内有字符 “aaaa”, 前缀 ”aaa” 与 后缀“aaa” 相等,所以k = 4; next[5] = 4;

逻辑教育
Logic education

KMP 模式匹配算法_next 数组值推导

T = “aaaaaaaaab”

j	123456789
模式串T	aaaaaaaab
next[j]	012345678

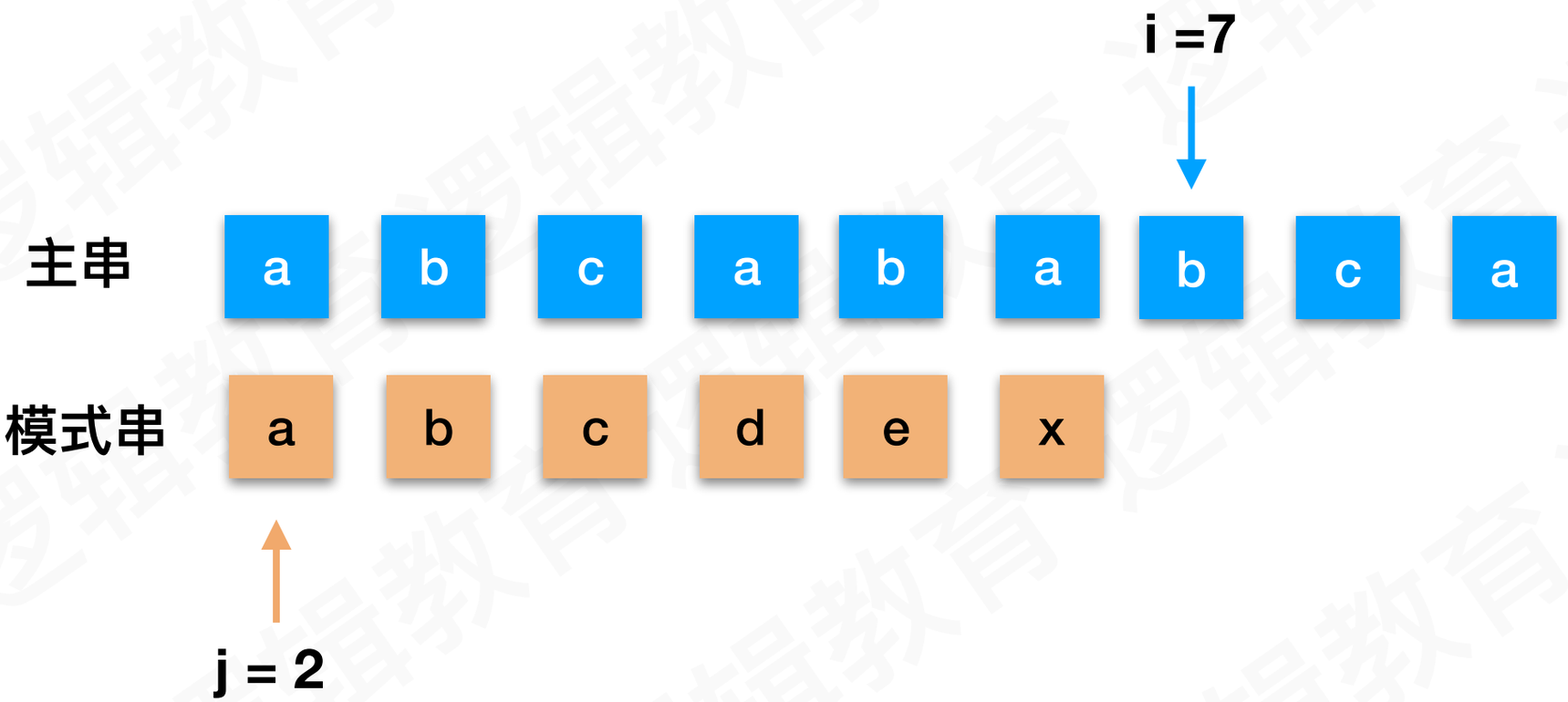
解读:

- 当 j=6时, j 由 1到 j-1 范围内有字符 “aaaaa”, 前缀 ”aaaa” 与 后缀“aaaa” 相等,所以k = 5; next[6] = 5;
- 当 j=7时, j 由 1到 j-1 范围内有字符 “aaaaaa”, 前缀 ”aaaaa” 与 后缀“aaaaa” 相等,所以k = 6; next[7] = 6;
- 当 j=8时, j 由 1到 j-1 范围内有字符 “aaaaaaa”, 前缀 ”aaaaaa” 与 后缀“aaaaaa” 相等,所以k = 7; next[8] = 7;
- 当 j=9时, j 由 1到 j-1 范围内有字符 “aaaaaaaa”, 前缀 ”aaaaaaa” 与 后缀“aaaaaaa” 相等,所以k = 8; next[9] = 8;

KMP 模式匹配算法_next 数组值推导 — 理解回溯 abcdex

假设，主串S = “abcababca”；模式串 T = “abcdex”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	1	1



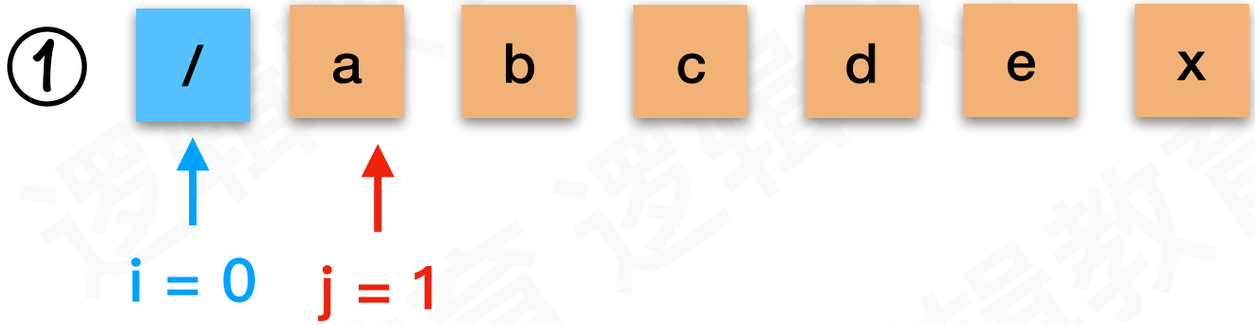
课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

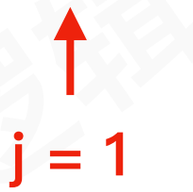
KMP 模式匹配算法_next 数组值推导 — 理解回溯 abcdex

假设，主串S = “abcababca”；模式串 T = “abcdex”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	1	1



下标	0	1	2	3	4	5	6
next	/	0	/	/	/	/	/



① 默认next[1] = 0
② i = 0, j = 1 开始 遍历
③ 当 j < T.length j 从1~length 遍历字符串;
④ 如果当 i = 0 表示[i , j] 这个范围内没有找到相同的字符,所以i 要回溯到1的位置; 表示next[j] = i;
⑤ 如果当 T[i] = T[j] 相等,表示找到与其相同字符的位置,所以next[j] = i;
⑥ 当以上2个条件都不满足,则将i 回溯到前面记录的next[i] 的位置;

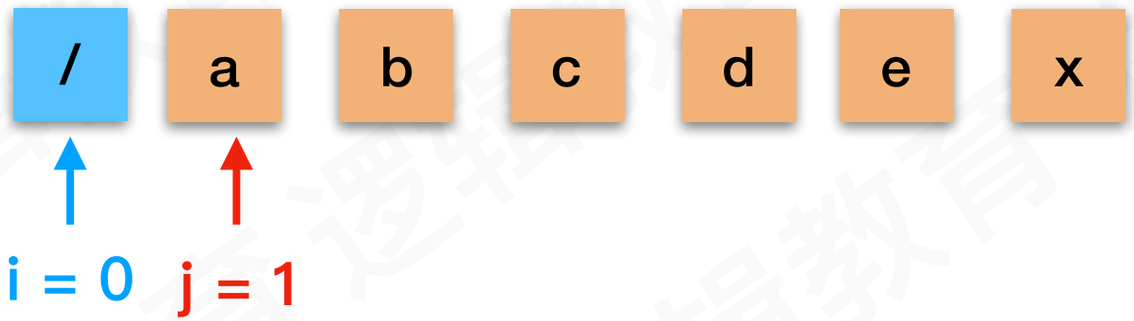
课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯

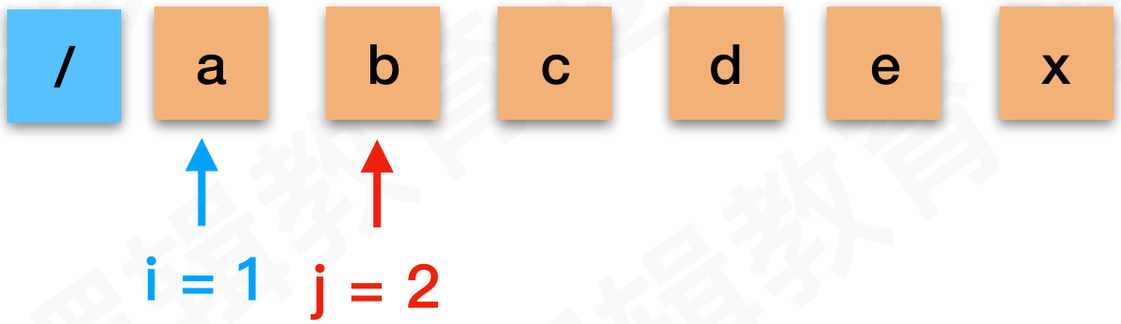
假设，主串S = “abcababca”；模式串 T = “abcdex”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	1	1

判断前



判断后



- 比较 T[i] != T[j] 但是 i = 0 ；则表示[0,1]这个范围[a] 只能从1的位置开始;
- j++, i++,所以 i = 1, j = 2;
- 并且更新next[j] = i; 则next[2] = 1;

下标	0	1	2	3	4	5	6
next	/	0	1	/	/	/	/

j = 2

课程研发:CC老师
课程授课:CC老师

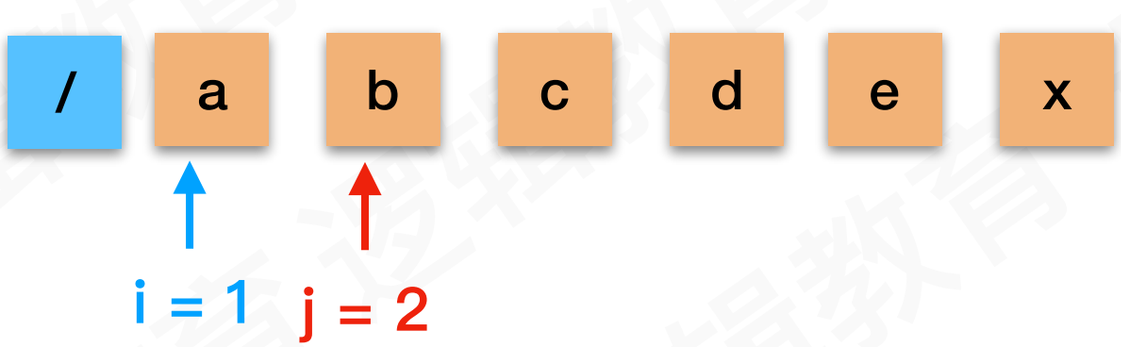
逻辑教育
Logic education

KMP 模式匹配算法_next 数组值推导 — 理解回溯

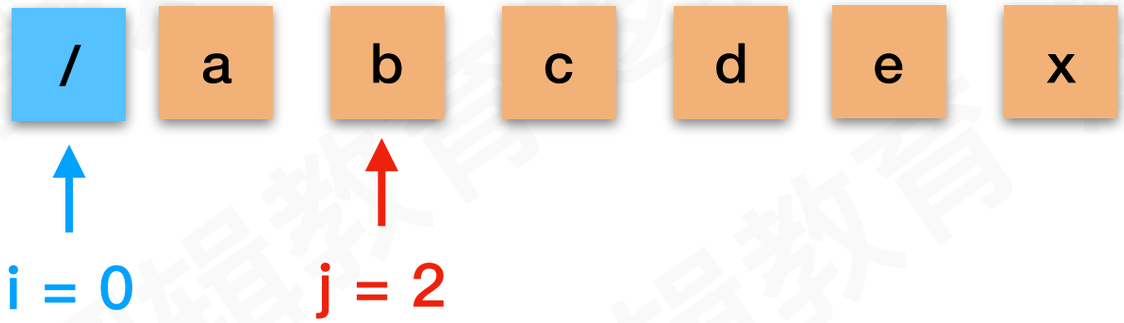
假设，主串S = “abcababca”；模式串 T = “abcdex”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	1	1

判断前



判断后



- 比较 T[i] != T[j] 所以将 i 的位置回退到a之后,那么 i = next[i] ; 即可 i = next[1] = 0;

下标	0	1	2	3	4	5	6
next	/	0	1	/	/	/	/

j = 2

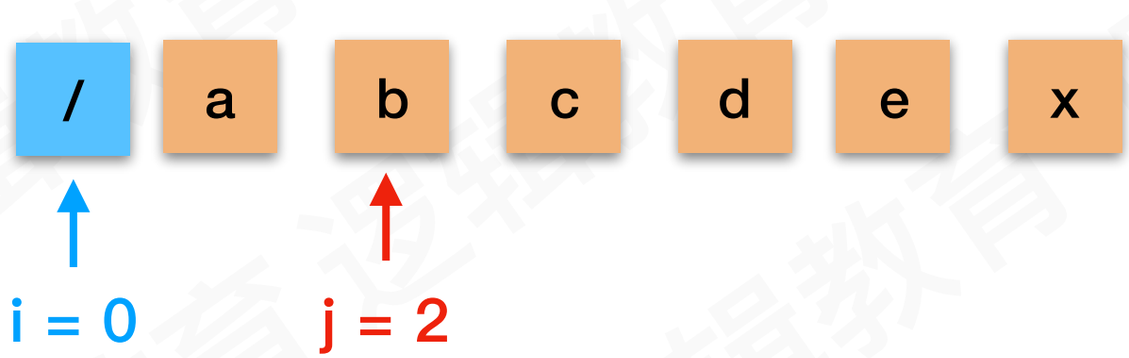
课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯 abcdex

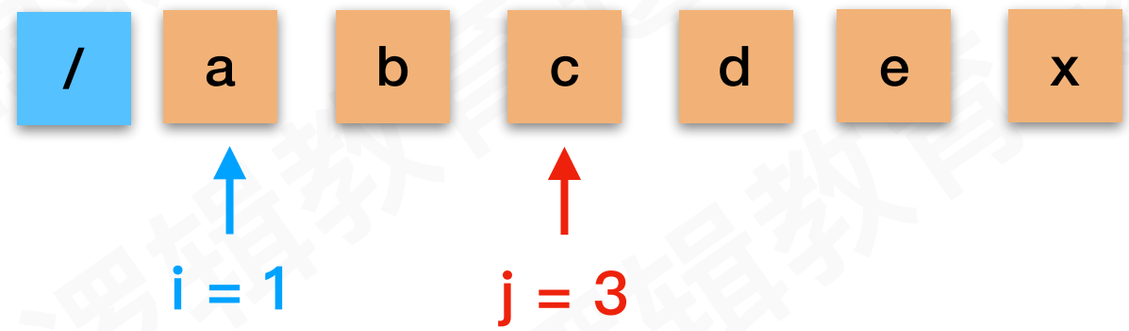
假设，主串S = “abcababca”；模式串 T = “abcdex”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	1	1

判断前



判断后



- 此时比较 [0,2] 这个范围是否存在相等字符出现;
- 那么由于 i = 0, 所以此时的next[j] 还是从第1个位置上的字符开始比较;
- i++,j++;　next[j] = next[3] = 1;

下标	0	1	2	3	4	5	6
next	/	0	1	1	/	/	/

j = 3

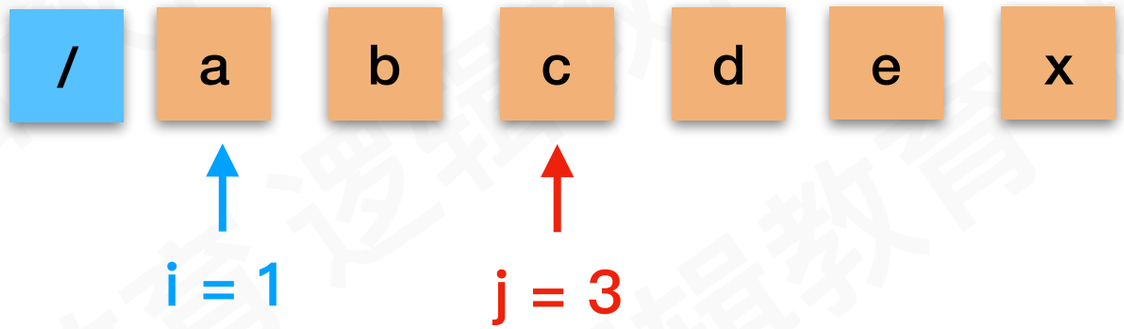
课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯

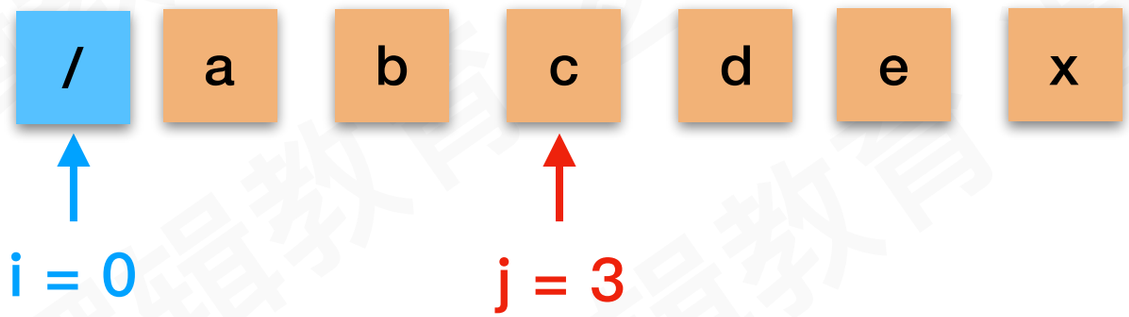
假设，主串S = “abcababca”；模式串 T = “abcdex”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	1	1

判断前



判断后



- 此时比较 [1,3] 这个范围是否存在相等字符出现;
- T[i] != T[j] 所以i 的位置又要回退.
- i = next[1] = 0; 此时 i = 0;

下标	0	1	2	3	4	5	6
next	/	0	1	1	/	/	/

j = 3

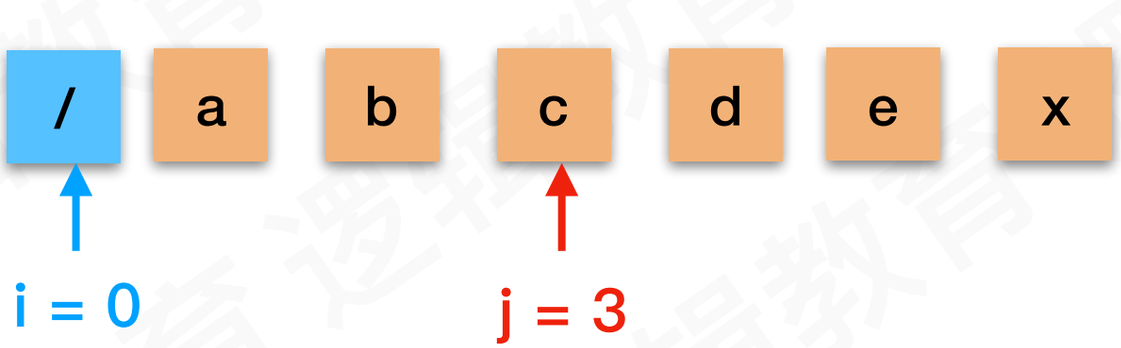
课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯

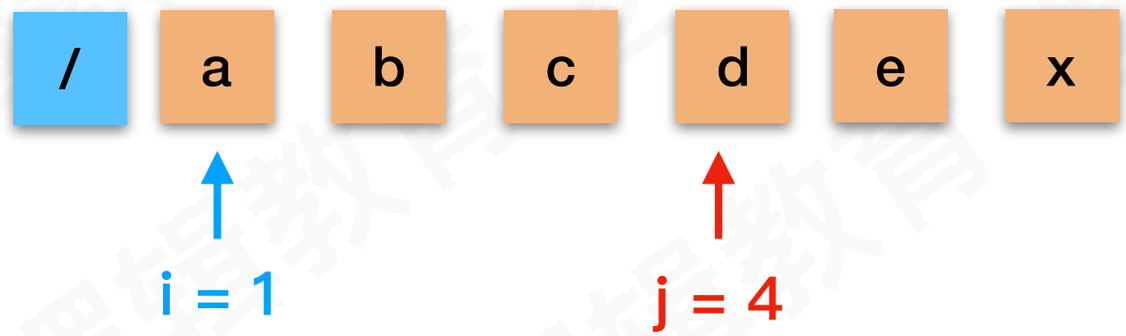
假设，主串S = “abcababca”；模式串 T = “abcdex”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	1	1

判断前



判断后



- 此时比较 [0,3] 这个范围是否存在相等字符出现;
- 因为 出现 i=0, 也就是字符串比较又要重头开始,则 i++,j++; i=1, j = 4;
- next[j] = i; next[4] = 1;

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	/	/

j = 4

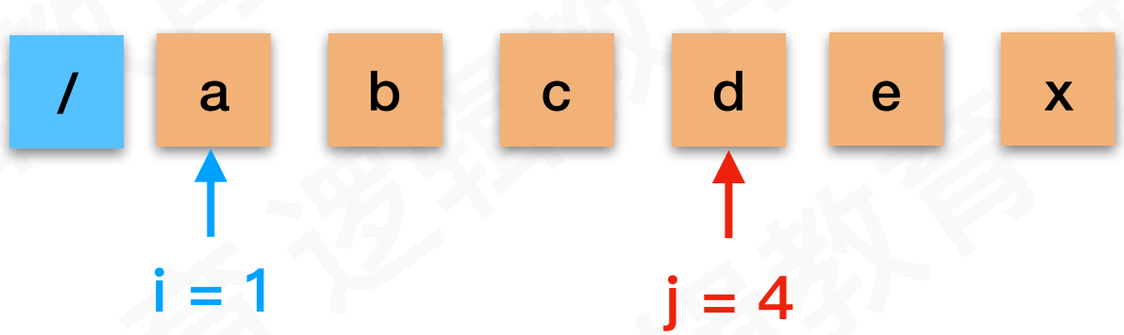
课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯

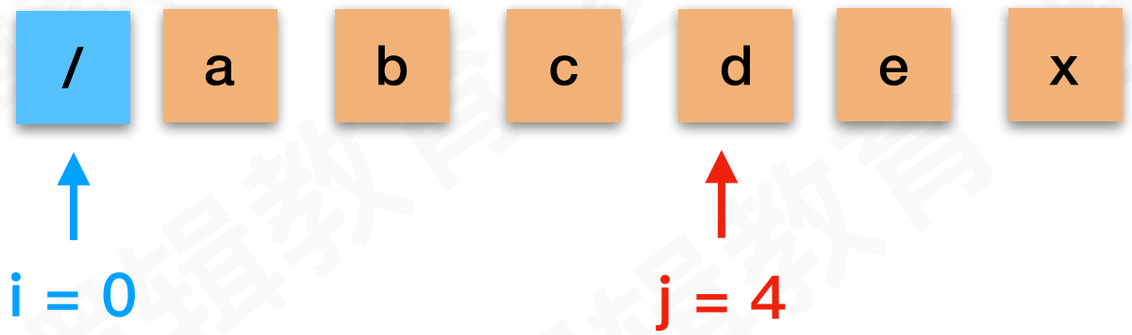
假设，主串S = “abcababca”；模式串 T = “abcdex”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	1	1

判断前



判断后



- 此时比较 [1,4] 这个范围是否存在相等字符出现;
- 那么T[i] != T[j] 所以 i 需要回溯. i = next[i] = next[1] = 0;

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	/	/

j = 4

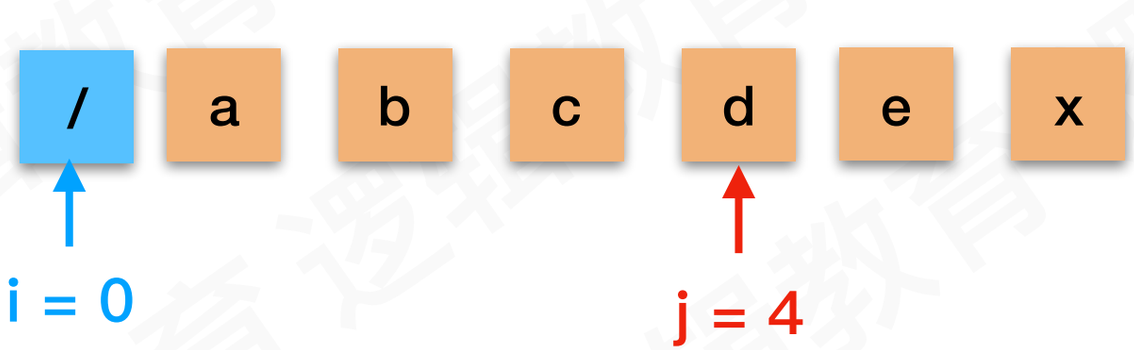
课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯

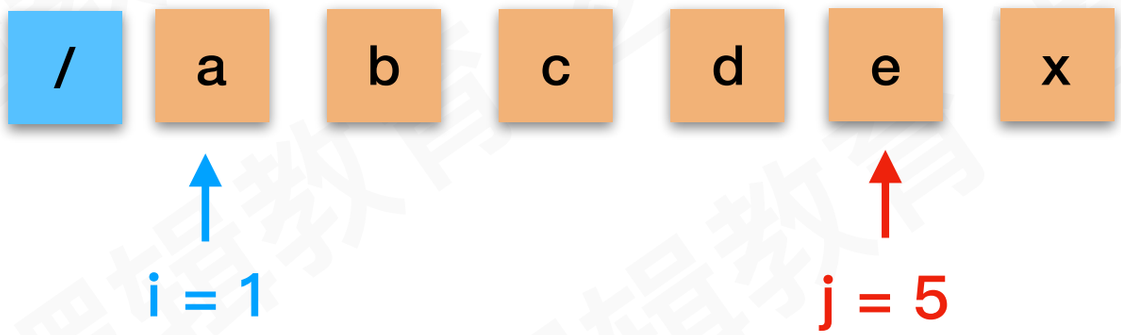
假设，主串S = “abcababca”；模式串 T = “abcdex”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	1	1

判断前



判断后



- 此时比较 [0,4] 这个范围是否存在相等字符出现;
- 因为 出现 i=0, 也就是字符串比较又要重头开始,则 i++,j++; i=1, j = 5;
- next[j] = i; next[5] = 1;

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	1	/

j = 5

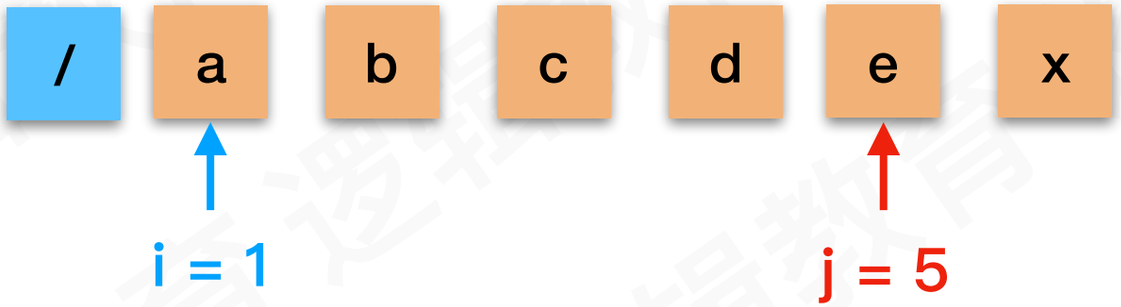
课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯

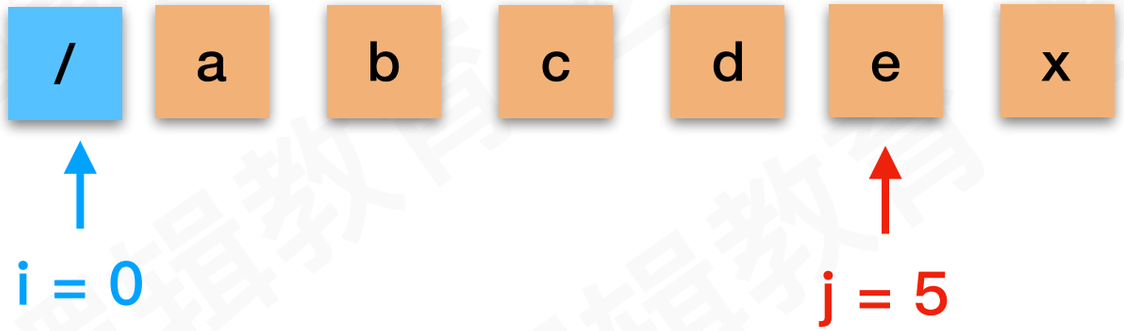
假设，主串S = “abcababca”；模式串 T = “abcdex”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	1	1

判断前



判断后



- 此时比较 [1,5] 这个范围是否存在相等字符出现;
- 那么T[i] != T[j] 所以 i 需要回溯. i = next[i] = next[1] = 0;

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	1	/

j = 5

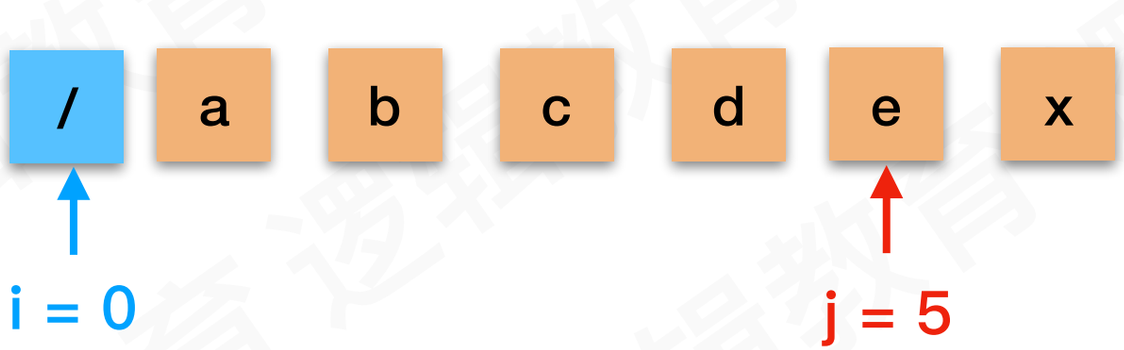
课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯

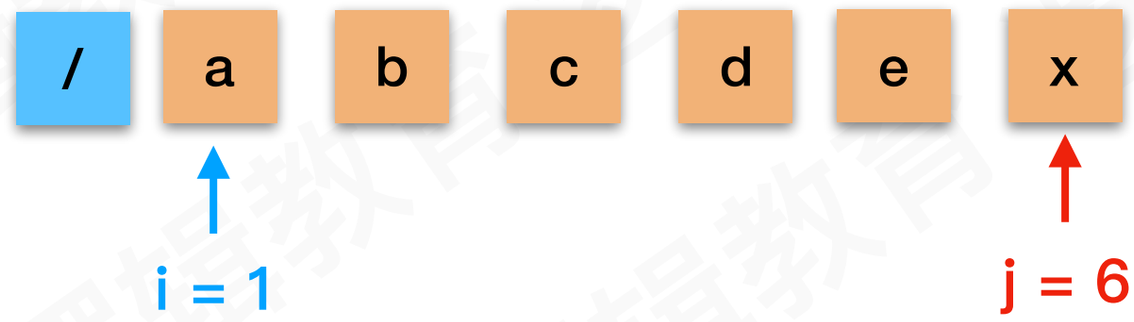
假设，主串S = “abcababca”；模式串 T = “abcdex”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	1	1

判断前



判断后



- 此时比较 [0,5] 这个范围是否存在相等字符出现;
- 但是由于i = 0 也就是字符串比较又要重头开始,则 i++,j++; i=1, j = 6;
- next[j] = i; next[6] = 1;

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	1	1

j = 6

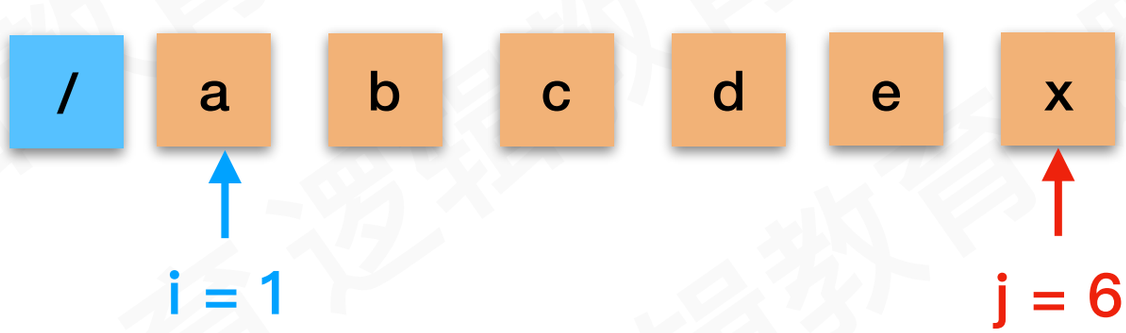
课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯

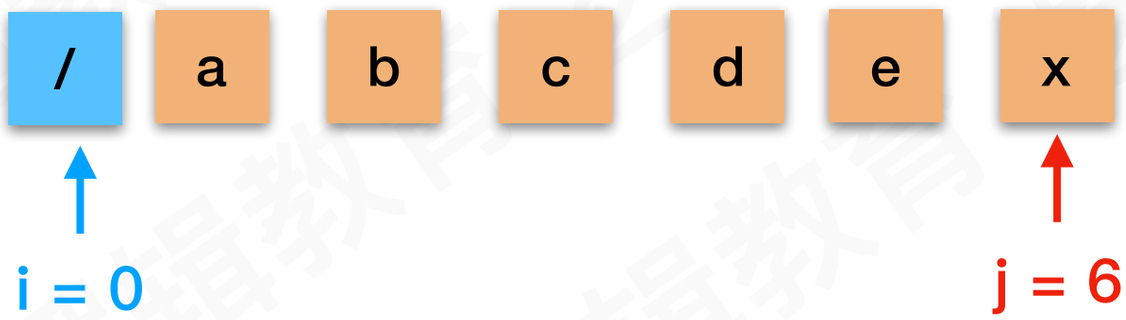
假设，主串S = “abcababca”；模式串 T = “abcdex”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	1	1

判断前



判断后



- 此时 j = 6 模式串已经处理完毕.则退出循环; j < S.length;

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	1	1

j = 6

课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯 总结

i = 0 ,表示比较过程中发现后面没有重复的字符,所以主串和模式串i=1的位置开始一轮比较;

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	1	1

在求解next数组的4种情况:

1. 默认next[1] = 0;
2. 当 i=0时,表示当前的比应该从头开始.则i++,j++,next[j] = i;
3. 当 T[i] == T[j] 表示2个字符相等,则i++,j++.同时`next[j] = i;
4. 当 T[i] != T[j]` 表示不相等,则需要将i 退回到合理的位置. 则 i = next[i];

T[i] != T[j] 进行回退. [i,j]范围有没有前缀和后缀;
[next[i],j] 范围有没有前缀和后缀;
最终直到[0,j]范围有没有前缀和后缀;

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

KMP 模式匹配算法_next 数组值推导 — 理解回溯 总结

i = 0 ,表示比较过程中发现后面没有重复的字符,所以主串和模式串i=1的位置开始一轮比较;

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	1	1

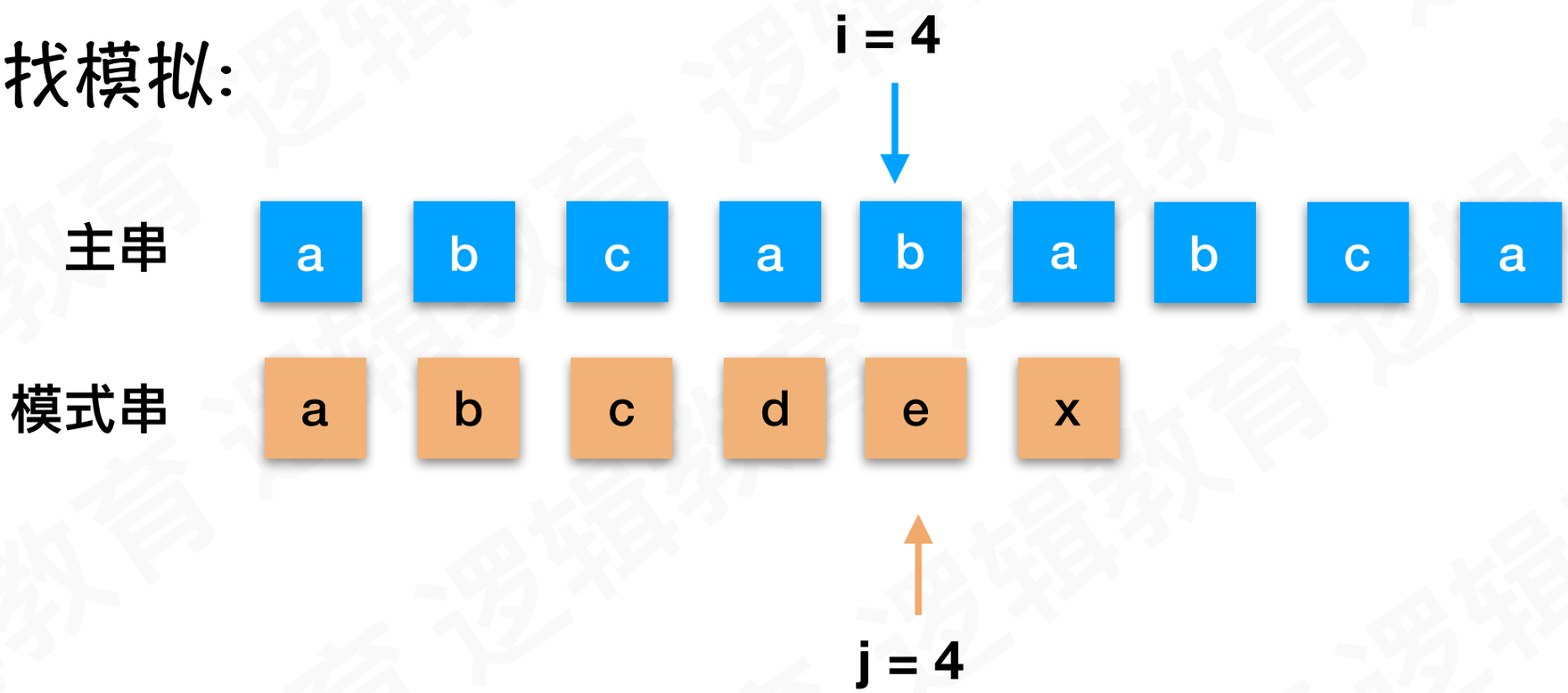
表示"abcababca" 与 "abcdex"
第一次 abcababca 与 abcdex 比较失败后; j = next[4] = 1;
第二次 abcababca 与 abcdex 还是需要从1这个位置开始重新比较.;
第三次 abcababca 与 abcdex 还是需要从1这个位置开始重新比较. j = next[3] = 1;
...
那么这个i是如何获取的就是从next数组中获取的;
遍历结束条件① 主串的索引 i 大于主串的长度;
② 模式串的 j 大于模式串的长度; 注意模式串的 j 如果不匹配是回退.是不会一直递增的;

课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯 abcdex

假设，主串S = “abcababca”；模式串 T = “abcdex”

BF算法查找模拟:

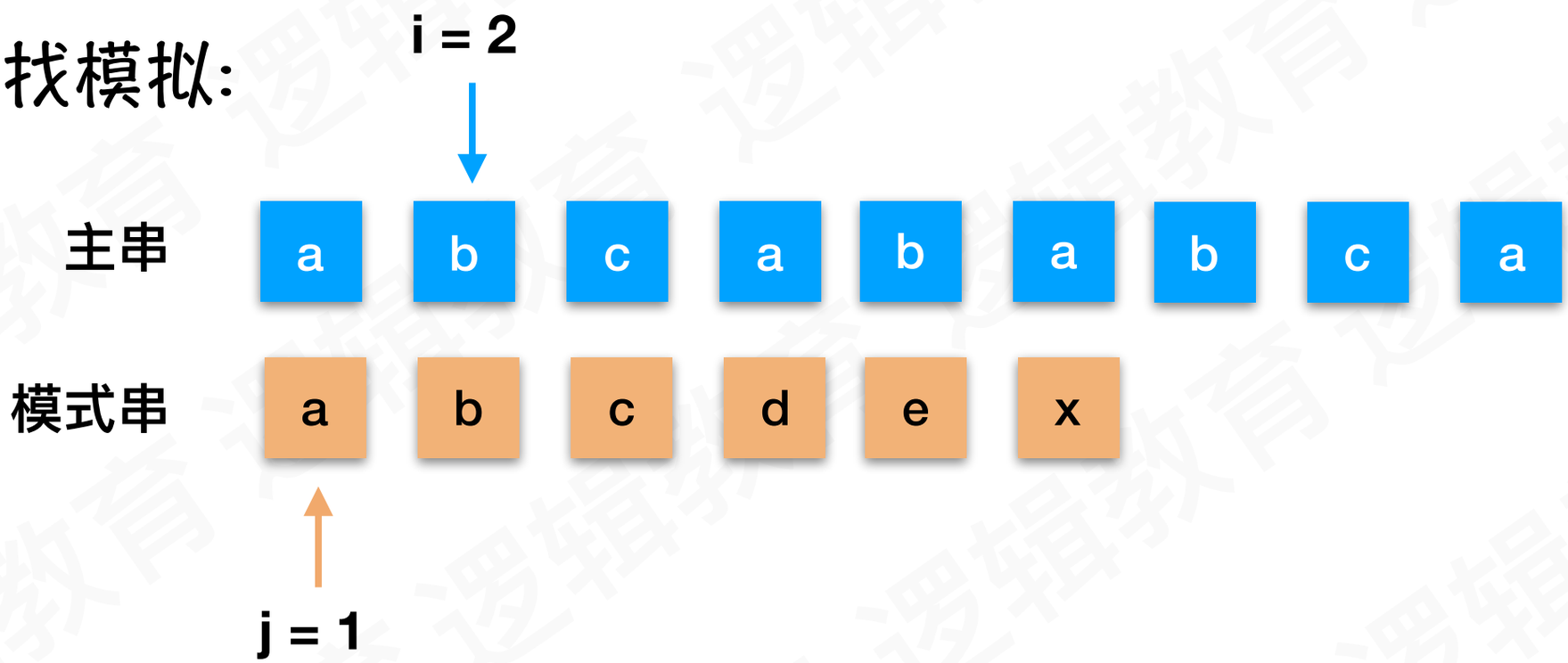


课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯 abcdex

假设，主串S = “abcababca”；模式串 T = “abcdex”

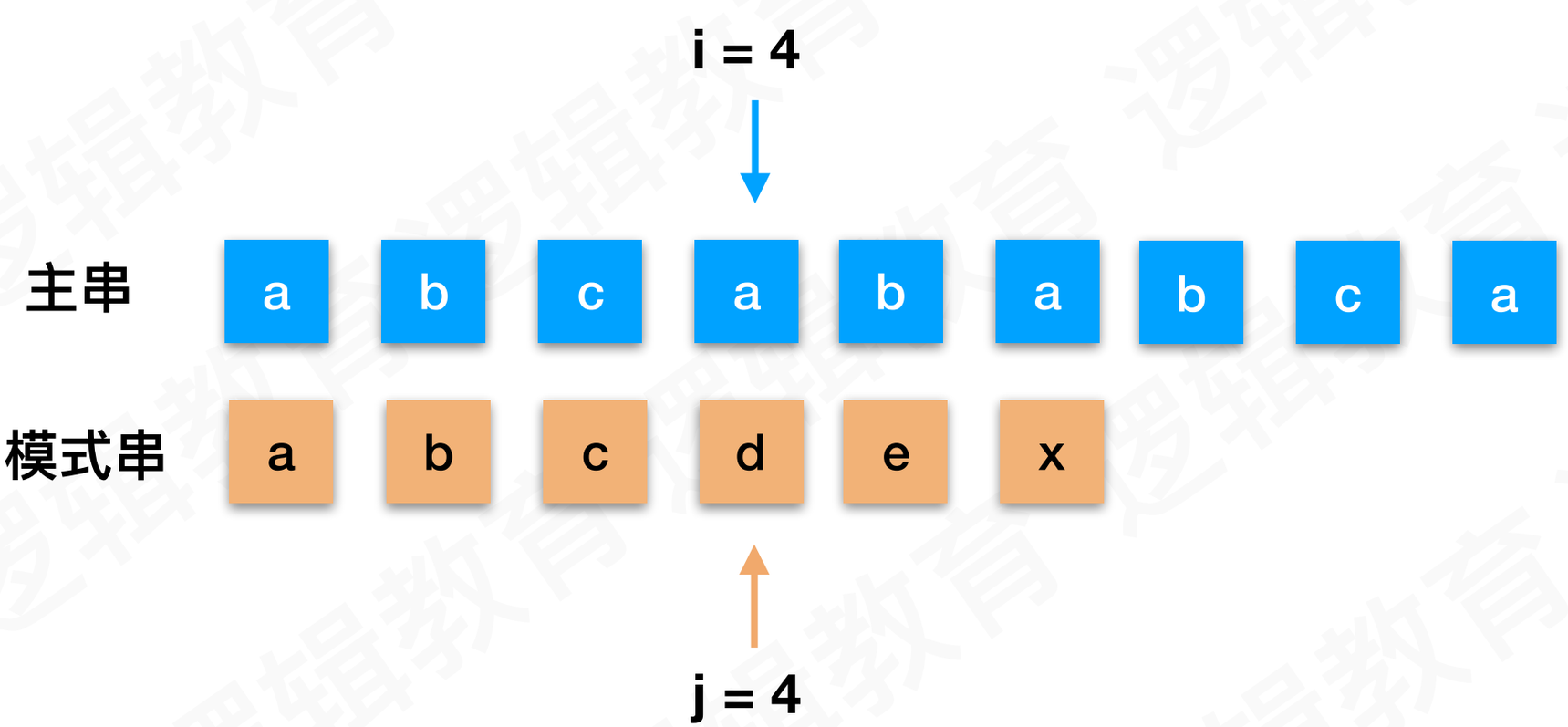
BF算法查找模拟:



KMP 模式匹配算法_next 数组值推导 — 理解回溯 abcdex

假设，主串S = “abcababca”；模式串 T = “abcdex”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	1	1



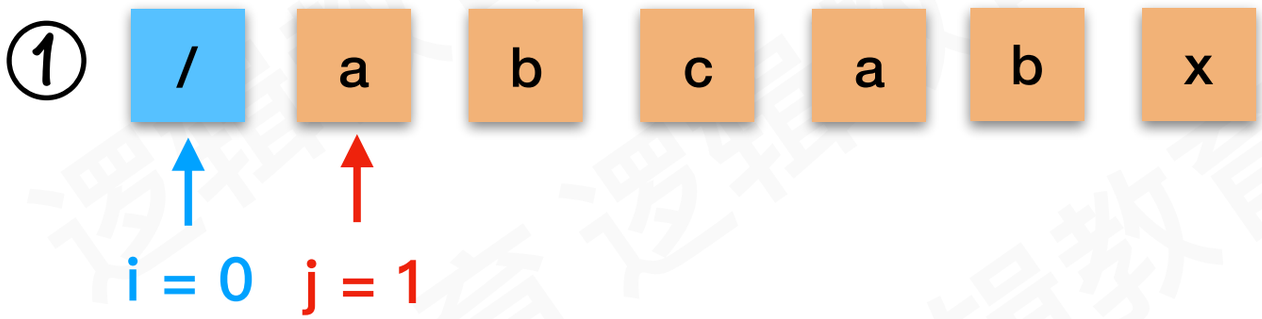
逻辑教育
Logic education

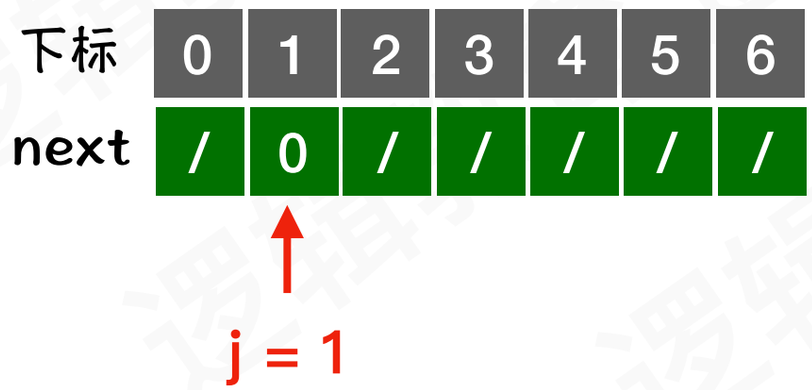
KMP 模式匹配算法_next 数组值推导 — 理解回溯 abcabx

假设，主串S = “abcababca”；模式串 T = “abcabx”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	2	3

① 默认next[1] = 0
② i = 0, j = 1 开始 遍历
③ 当 j < S.length j 从1~length 遍历字符串;
④ 如果当 i = 0 表示[i , j] 这个范围内没有找到相同的字符,所以i 要回溯到1的位置; 表示next[j] = i;
⑤ 如果当 T[i] = T[j] 相等,表示找到与其相同字符的位置,所以next[j] = i;
⑥ 当以上2个条件都不满足,则将i 回溯到前面记录的next[i] 的位置;





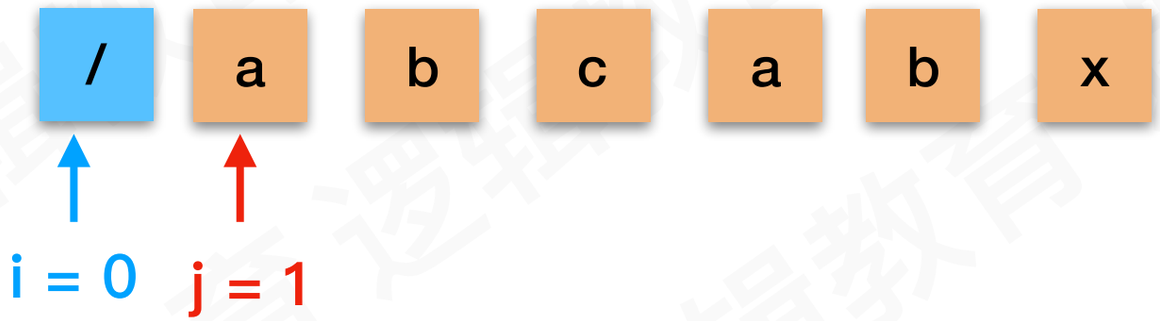
课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯

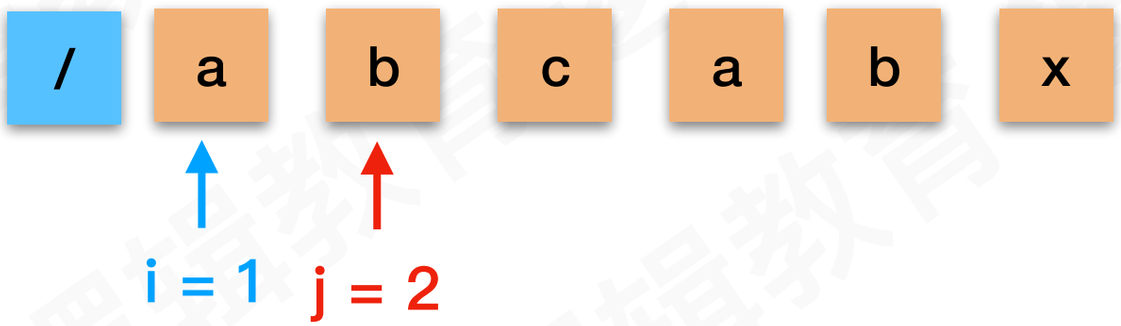
假设，主串S = “abcababca”；模式串 T = “abcabx”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	2	3

判断前



判断后



- 比较 T[i] != T[j] 但是 i = 0 ; 则表示[0,1]这个范围[a] 只能从1的位置开始;
- j++, i++,所以 i = 1, j = 2;
- 并且更新next[j] = i; 则next[2] = 1;

下标	0	1	2	3	4	5	6
next	/	0	1	/	/	/	/

j = 2

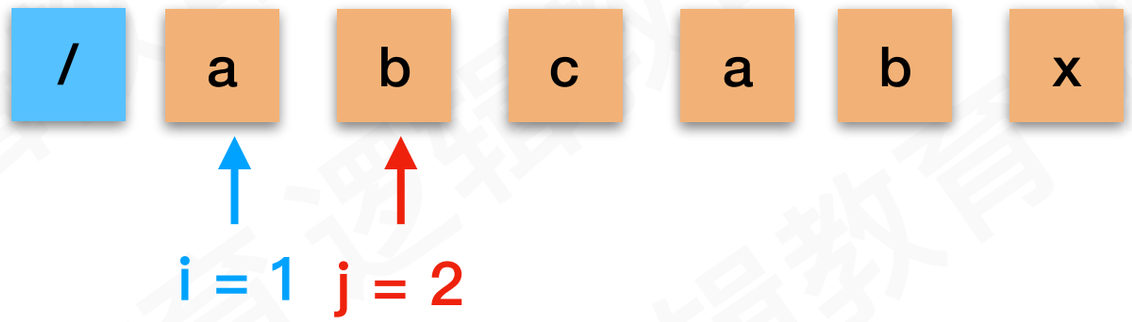
课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯

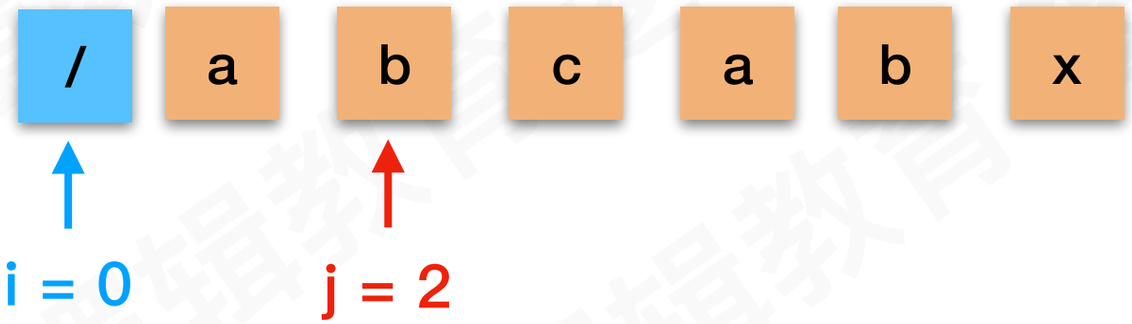
假设，主串S = “abcababca”；模式串 T = “abcabx”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	2	3

判断前



判断后



- 此时比较 [1,2] 这个范围是否存在相等字符出现;
- 那么T[i] != T[j] 所以 i 需要回溯. i = next[i] = next[1] = 0;

下标	0	1	2	3	4	5	6
next	/	0	1	/	/	/	/

j = 2

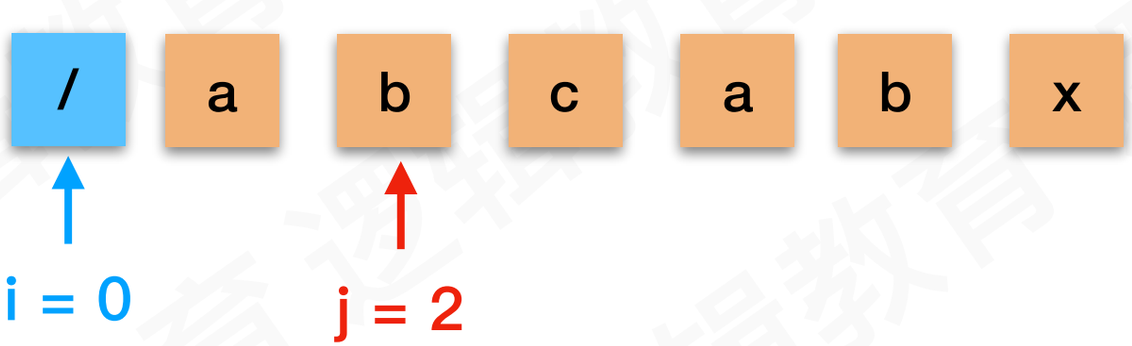
课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯

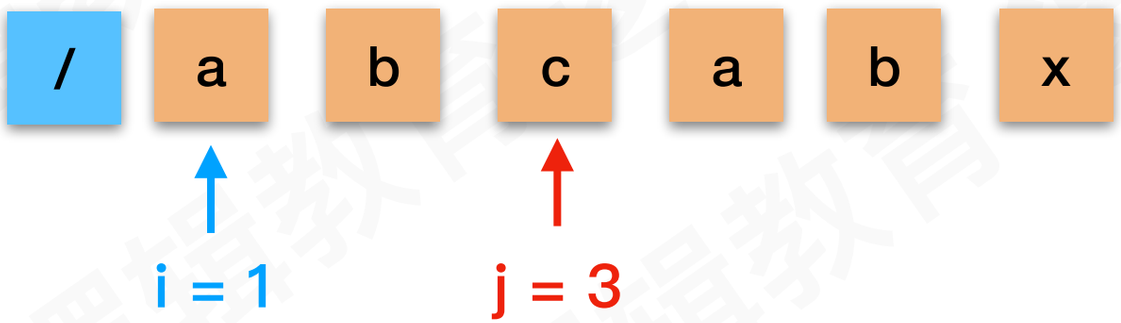
假设，主串S = “abcababca”；模式串 T = “abcabx”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	2	3

判断前



判断后



- 此时比较 [0,2] 这个范围是否存在相等字符出现;
- 但是由于i = 0 也就是字符串比较又要重头开始,则 i++,j++; i=1, j = 3;
- next[j] = i; next[3] = 1;

下标	0	1	2	3	4	5	6
next	/	0	1	1	/	/	/

j = 3

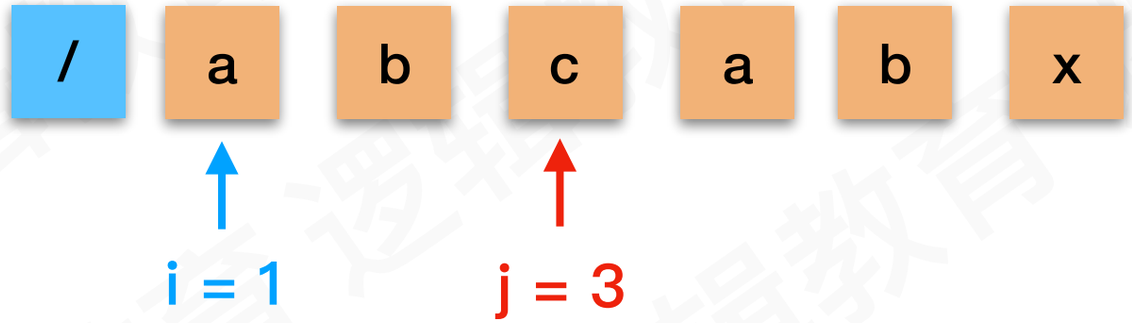
课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯

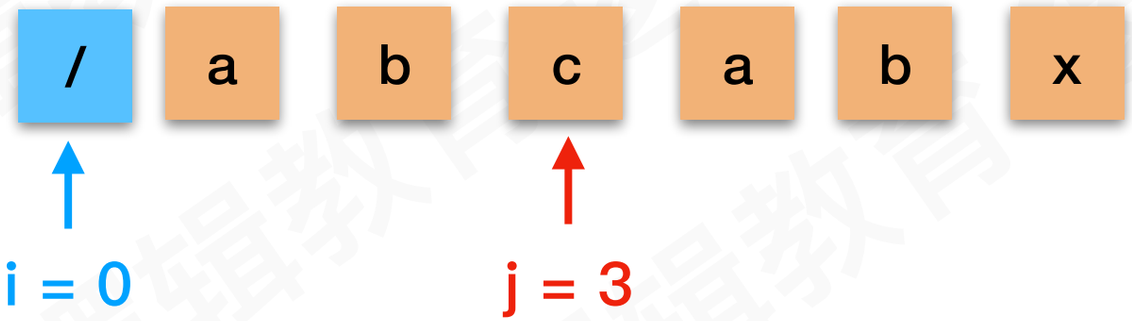
假设，主串S = “abcababca”；模式串 T = “abcabx”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	2	3

判断前



判断后



- 此时比较 [1,3] 这个范围是否存在相等字符出现;
- 那么T[i] != T[j] 所以 i 需要回溯. i = next[i] = next[1] = 0;

下标	0	1	2	3	4	5	6
next	/	0	1	1	/	/	/

j = 3

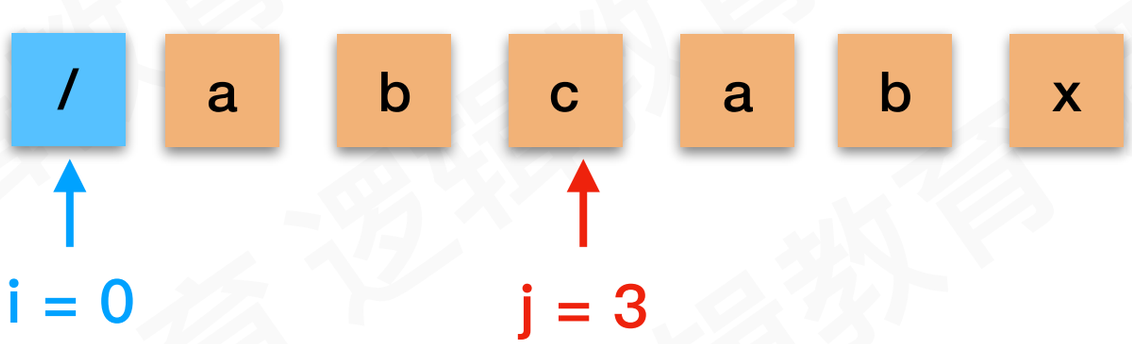
课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯

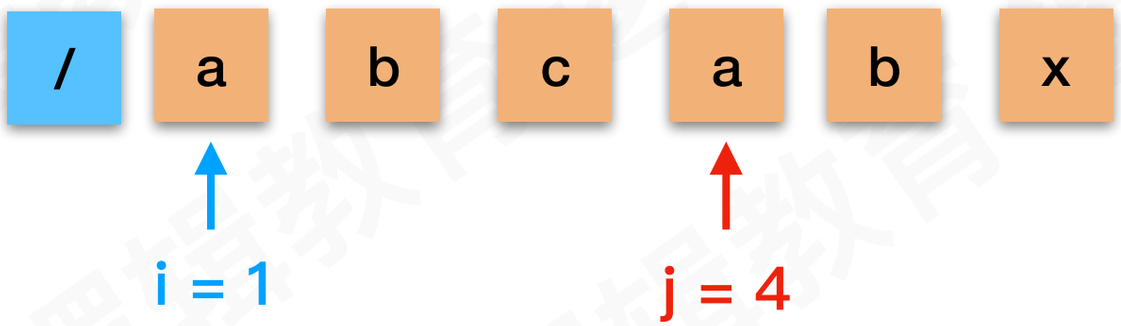
假设，主串S = “abcababca”；模式串 T = “abcabx”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	2	3

判断前



判断后



- 此时比较 [0,3] 这个范围是否存在相等字符出现;
- 但是由于i = 0 也就是字符串比较又要重头开始,则 i++,j++; i=1, j = 4;
- next[j] = i; next[4] = 1;

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	/	/

j = 4

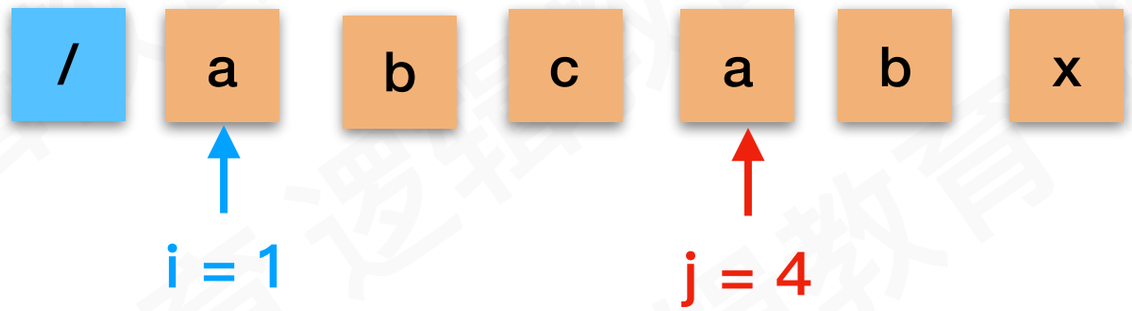
课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯

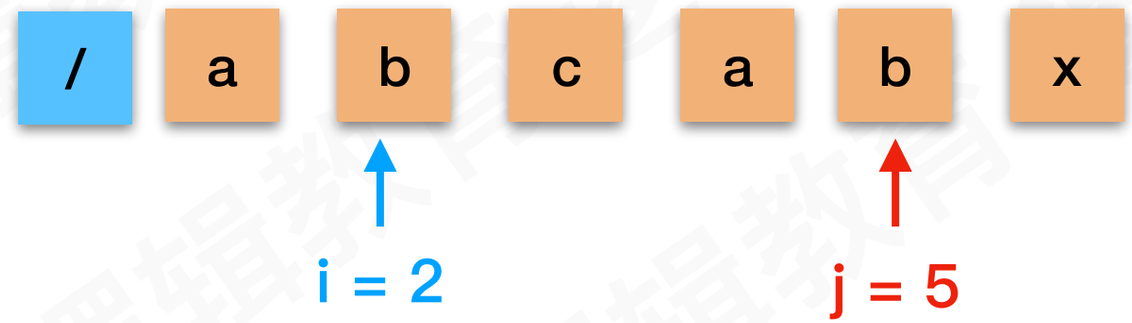
假设，主串S = “abcababca”；模式串 T = “abcabx”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	2	3

判断前



判断后



- 此时比较 [1,4] 这个范围是否存在相等字符出现;
- 那么T[i] == T[j] 所以 i++,j++; i=2,j = 5;
- next[5] = 2;

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	2	/

j = 5

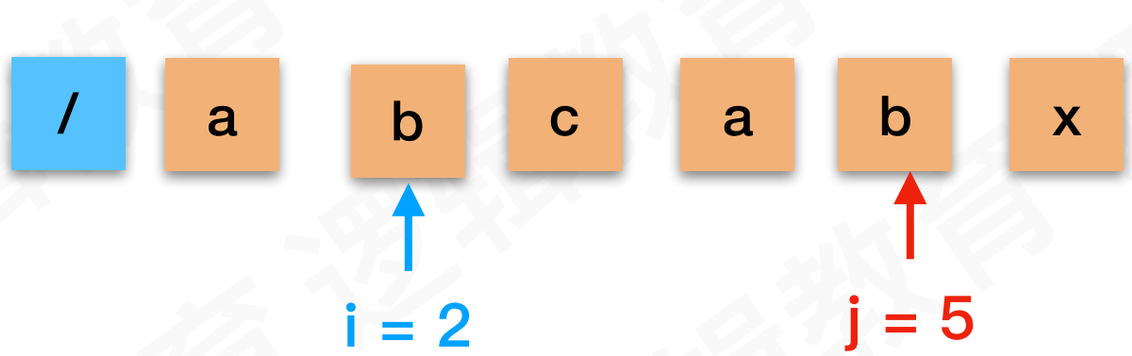
课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯

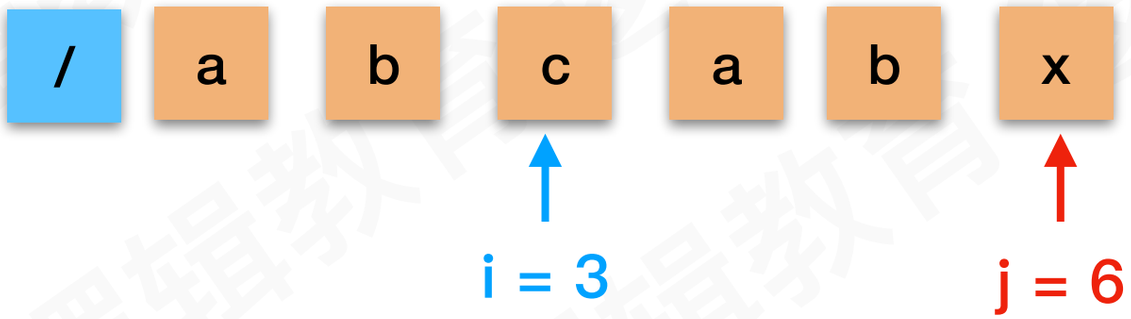
假设，主串S = “abcababca”；模式串 T = “abcabx”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	2	3

判断前



判断后



- 此时比较 [2, 5] 这个范围是否存在相等字符出现;
- 那么T[i] == T[j] 所以 i++,j++; i=3,j = 6;
- next[6] = 3;

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	2	3

j = 6

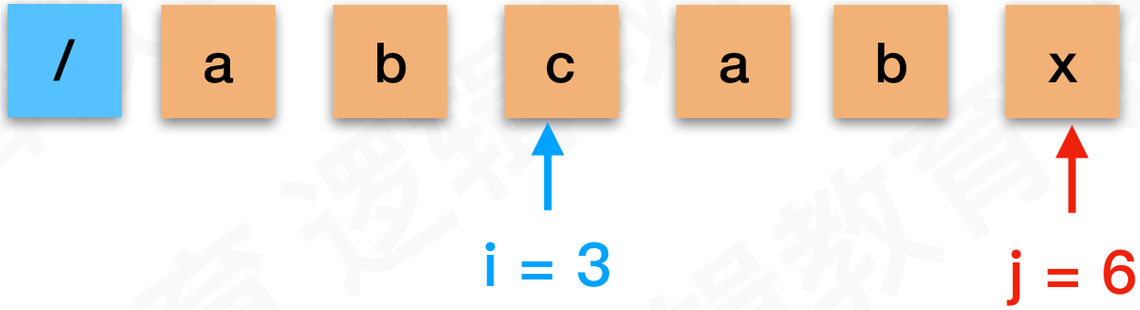
课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯

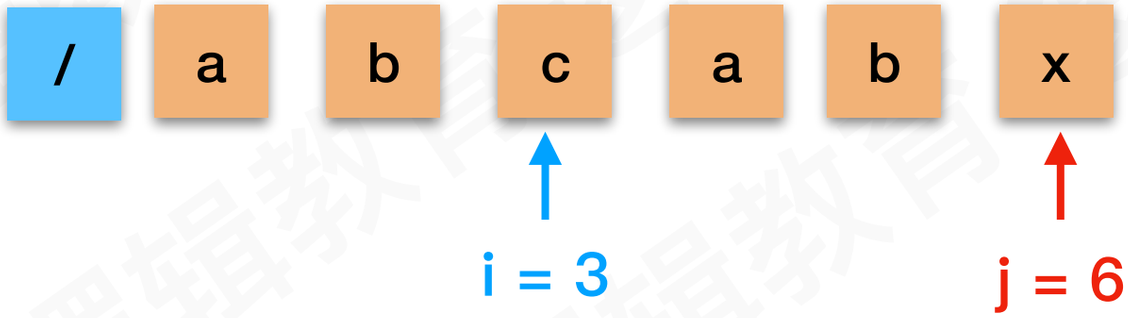
假设，主串S = “abcababca”；模式串 T = “abcabx”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	2	3

判断前



判断后



- 此时 j = 6 模式串已经处理完毕.则退出循环; j < S.length;

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	2	3

j = 6

课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法_next 数组值推导 — 理解回溯

假设，主串S = “abcababcabx”；模式串 T = “abcabx”

下标	0	1	2	3	4	5	6
next	/	0	1	1	1	2	3

表示”abcababcabx” 与 “abcabx”

第一次 abcababcax 与 abcabx　比较失败后; j = next[6] = 3;

第二次 abcababcax 与 abcabx　比较失败后, j = next[3] = 1;

第三次 abcababcax 与 abcabx　比较成功

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

KMP 模式匹配算法_next 数组值推导 — 理解回溯

根据刚刚的过程,完成next数组的求解代码实现

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

KMP 模式匹配算法 匹配函数的实现

假设，主串S = “abcababcabx”；模式串 T = “abcabx”

KMP思路:

1. 遍历模式串S,i 是用来标记主串的索引; 遍历模式串，j 是用来标记模式串的索引;
2. 结束条件是当i > S.length 和 j > T.length;
   - 如果 i > S.length 但是j 却小于T.length 表示遍历了整个主串,都没有找到与模式串匹配的情况
   - 只有1种可能,就是j > T.length 表示,已经在主串中找到模式串了. 因为你已经顺利的把T模式串中的每个字符串正常的依次比较下去了,直到它结束;

3. 当 j = 0 时,表示此时你需要将模式串从1这个位置与主串i+1这个位置开始比较;
4. 当 T[i] == T[j], 表示此时当前模式串j 与 主串i 这个2个字符是相等,则j++,i++;
5. 当 j != 0 并且T[i] != T[j] 时,表示此时需要移动模式串的 j ,那么我们让 j = next[j]; 来节省重复的比较次数;

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

KMP 模式匹配算法 完成KMP模式匹配查找

根据刚刚的过程,KMP 模式匹配算法(利用next数组)

课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法 匹配函数的实现

假设，主串S = “aaaabcde”；模式串 T = “aaaaax”

问题1：此时模式串 T 的next 数组为？

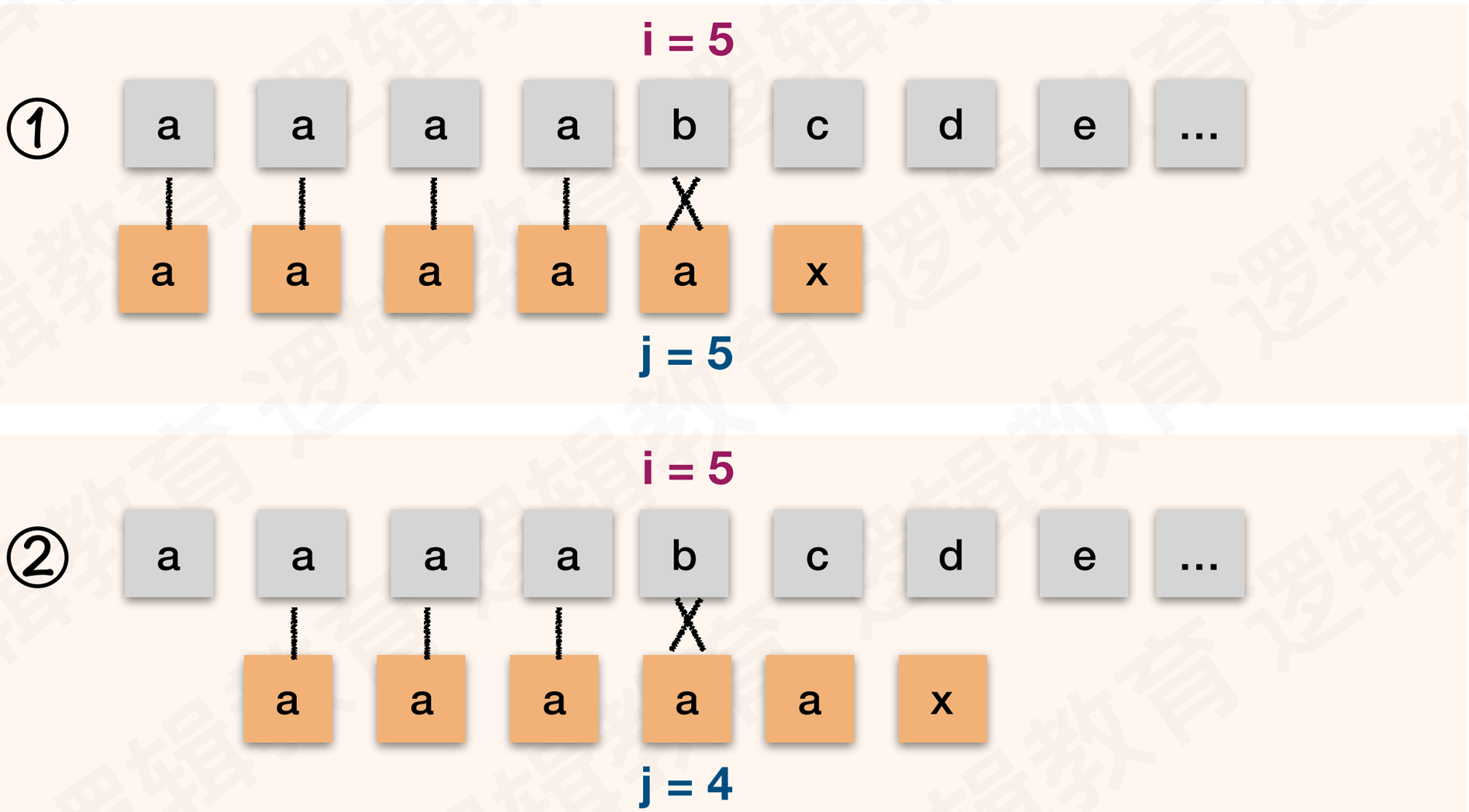
课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

012345

KMP 模式匹配算法

假设，主串S = “aaaabcde”；模式串 T = “aaaaax”

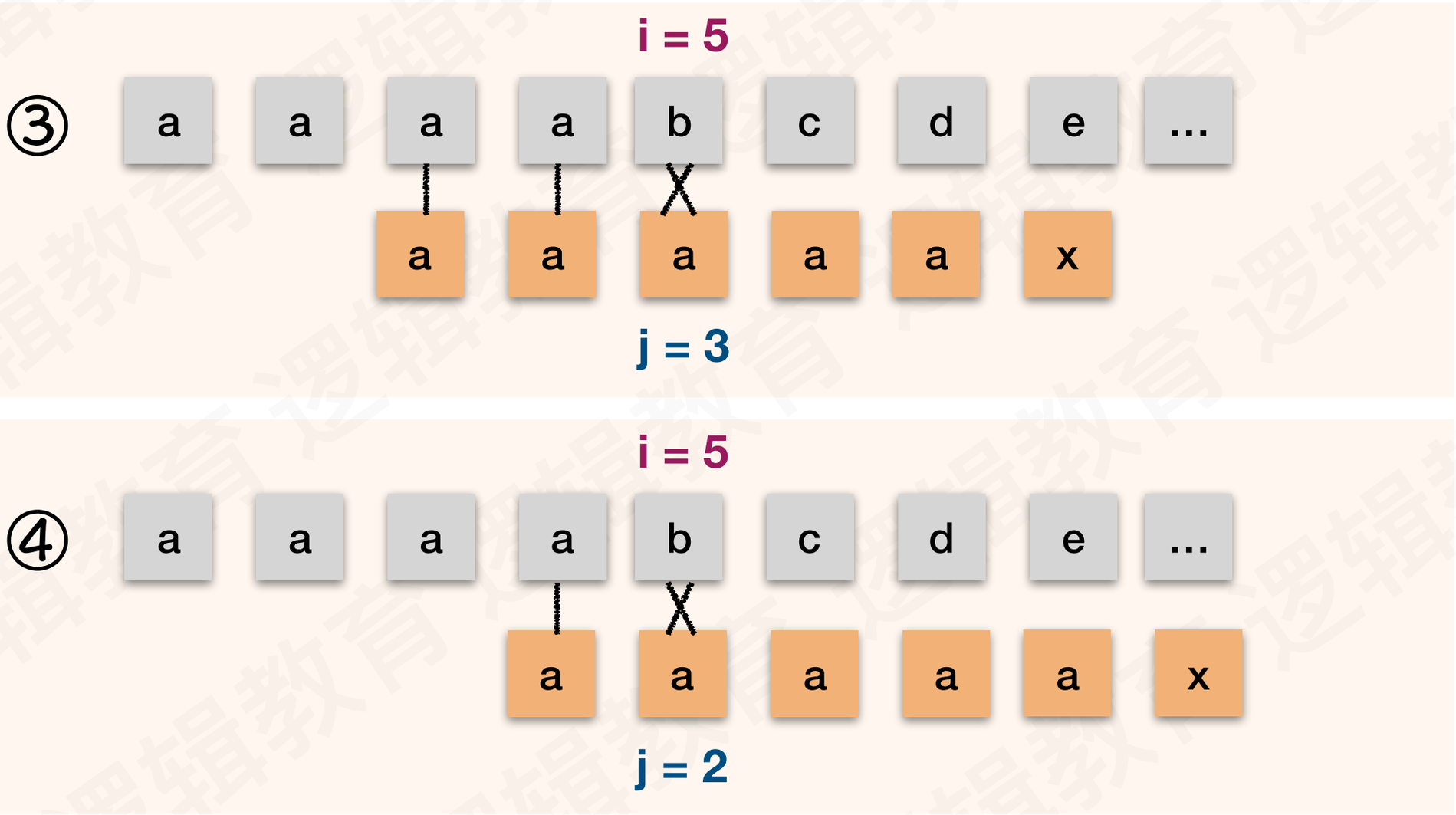


课程研发:CC老师
课程授课:CC老师

012345

KMP 模式匹配算法

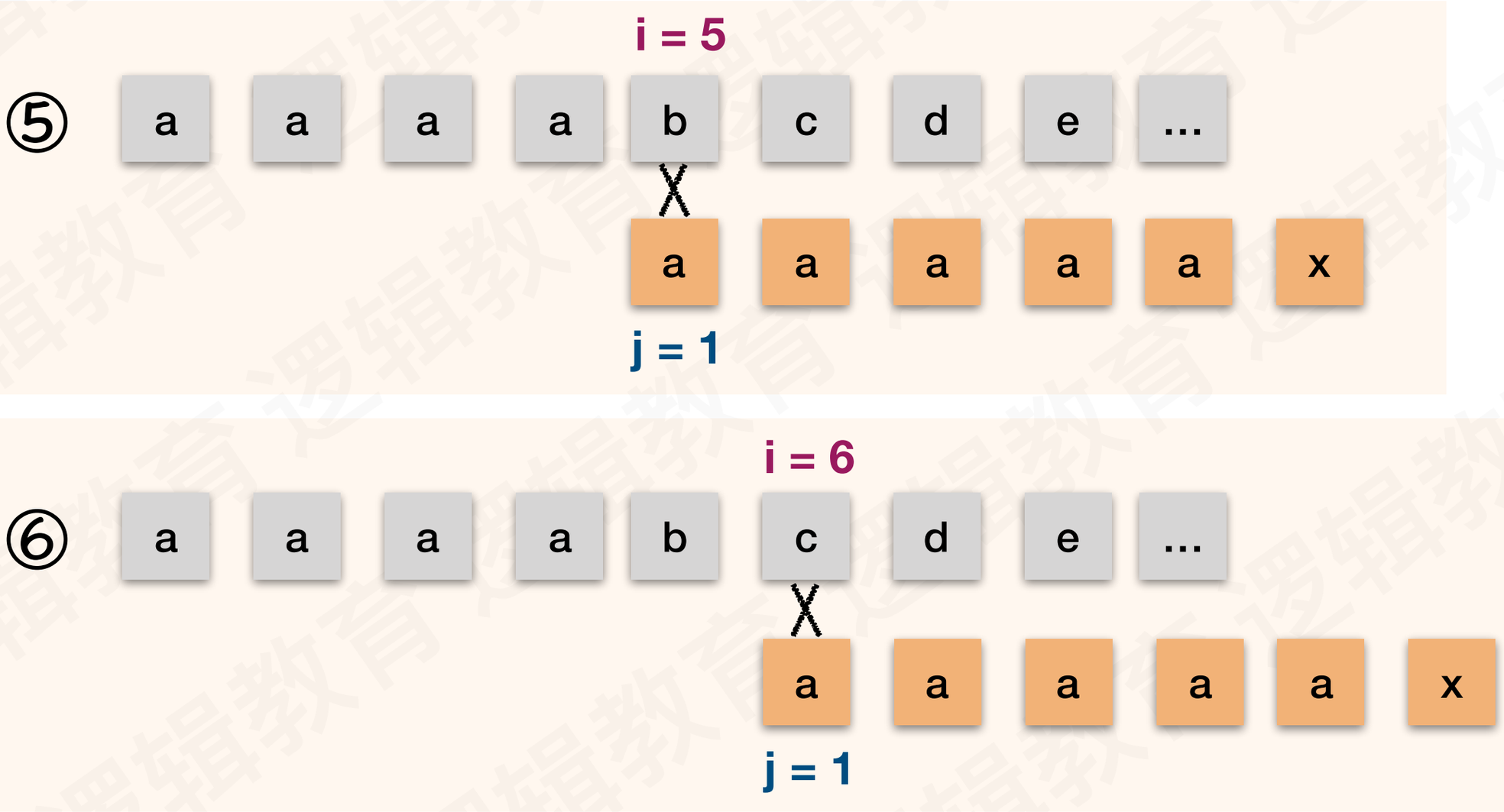
假设，主串S = “aaaabcde”；模式串 T = “aaaaax”



课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法

假设，主串S = “aaaabcde”；模式串 T = “aaaaax”



课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法 next 数组的优化

T = “abababaaaba”

j	123456789
模式串T	abababaaaba
next[j]	011234223
nextval[j]	010104210

解读:

- 当 j = 1, nextVal = 0;
- 当 j = 2, 因为第2个字符 “b” 的值next 值是1,而且第一个字符是”a”. 不相等. 所以 nextVal[2] = next[2] = 1;
- 当 j = 3, 因为第3个字符”a” 的next 值是1, 所以与第1位的”a”比较得知它们相等, 所以 nextval[3] = nextval[1] = 0;

KMP 模式匹配算法 next 数组的优化

- 当 j = 3, 因为第3个字符”a” 的next 值是1, 所以与第1位的”a”比较得知它们相等, 所以 nextval[3] = nextval[1] = 0;

j	1	2	3
T	a	b	a
next	0	1	1
nextval	0	1	0

next[3]为1 ,查看T[1] 与本位置上T[3] 是否相等

因为T[1] = T[3] . 所以nextVal[3] = nextVal[1] = 0;

课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法 next 数组的优化

- 当 j = 4, 因为第4个字符”b” 的next 值是2, 所以与第2位的”b”比较得知它们相等, 所以 nextval[4] = nextval[2] = 1;

j	1	2	3	4
T	a	b	a	b
next	0	1	1	2
nextval	0	1	0	1

next[4]为2 ,查看T[2] 与本位置上T[4] 是否相等

因为T[2] = T[4] . 所以nextVal[4] = nextVal[2] = 1;

课程研发:CC老师
课程授课:CC老师

KMP 模式匹配算法 next 数组的优化

T = “abababaaaba”

j	123456789
模式串T	abababaaaba
next[j]	011234223
nextval[j]	010104210

解读:

- 当 j = 5 时,next 值为3 , 第5个字符”a” 与第3个字符”a” 相等,则nextVal[5] = nextVal[3] = 0;
- 当 j = 6 时,next 值为4 , 第6个字符”a” 与第4个字符”b” 不相等,则nextVal[6] = 4;
- 当 j = 7 时,next 值为2 , 第7个字符”a” 与第2个字符”b” 不相等,则nextVal[7] = 2;
- 当 j = 8 时,next 值为2 , 第8个字符”b” 与第2个字符”b” 相等,则nextVal[6] = nextVal[2] = 1;
- 当 j = 9 时, next 值为3, 第9个字符”a” 与第3个字符”a” 相等,则nextVal[9] = nextVal[3] = 0;

逻辑教育
Logic education

KMP 模式匹配算法 next 数组的优化

在求解nextVal数组的5种情况:

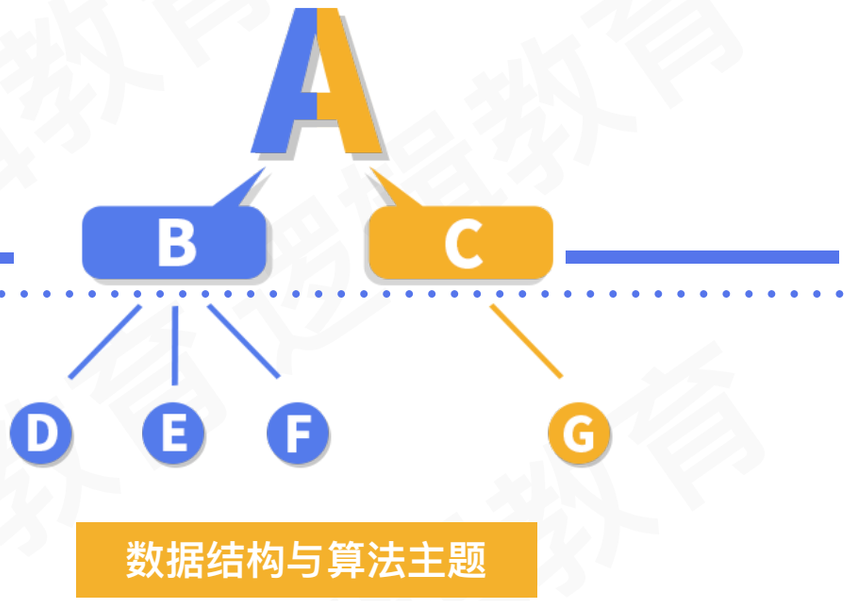
1. 默认next[1] = 0;
2. T[i] == T[j] 且++i,++j 后 T[i] 依旧等于 T[j] 则 nextval[i] = nextval[j]
3. i = 0, 表示从头开始i++,j++后,且T[i] != T[j] 则nextVal = j;
4. T[i] == T[j] 且++i,++j 后 T[i] != T[j] ,则nextVal = j;
5. 当 T[i] != T[j]` 表示不相等,则需要将i 退回到合理的位置. 则 i = next[i];

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

Class Ending !

thanks, see you next time



@CC老师

全力以赴，非同凡"想"

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

课程研发:CC老师
课程授课:CC老师