

本节内容

KMP算法

进一步优化



KMP算法

根据模式串T, 求出 next 数组

利用next数组进行匹配 (主串指针不回溯)

T = 'abaabc'

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
	0	1	1	2	2	3

```
if (S[i] != T[j]) j = next[j];
if (j == 0) { i++; j++; }
```

KMP算法, 最坏时间复杂度 $O(m+n)$

其中, 求 next 数组时间复杂度 $O(m)$
模式匹配过程最坏时间复杂度 $O(n)$

```
int Index_KMP(SString S, SString T, int next[]) {
    int i=1, j=1;
    while (i<=S.length && j<=T.length) {
        if (j==0 || S.ch[i]==T.ch[j]) {
            ++i;
            ++j; // 继续比较后继字符
        } else {
            j=next[j]; // 模式串向右移动
        }
    }
    if (j>T.length) return i-T.length; // 匹配成功
    else return 0;
}
```

手算求next数组的方法

根据模式串T, 求出 next 数组

T = 'abaabc'

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
	0	1	1	2	2	3

```
if (S[i] != T[j]) j = next[j];
if (j == 0) { i++; j++; }
```

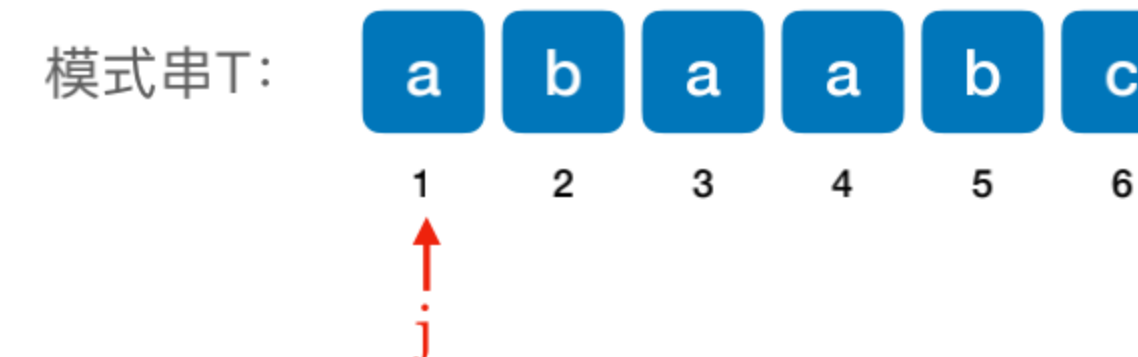
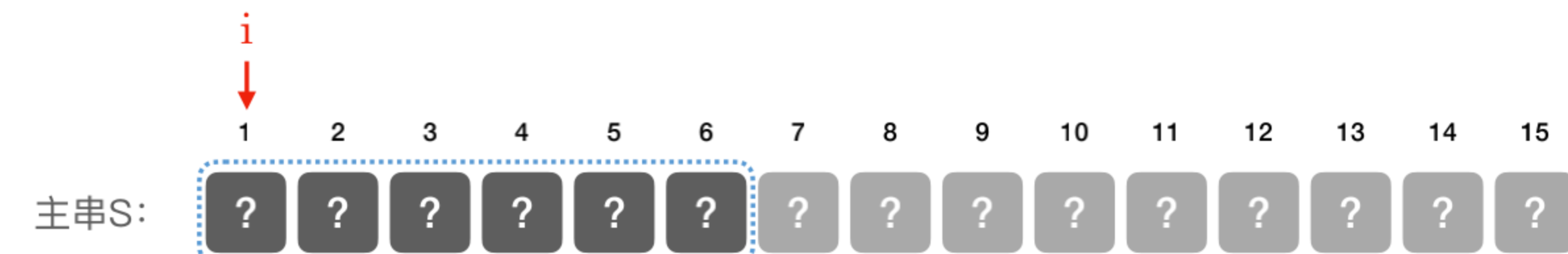
next[1]都无脑写 0
next[2]都无脑写 1

其他 next: 在不匹配的位置前, 划一根美丽的分界线
模式串一步一步往后退, 直到分界线之前“能对上”, 或模式串完全跨过分界线为止。此时 j 指向哪儿, next数组值就是多少

KMP算法, 最坏时间复杂度 $O(m+n)$

其中, 求 next 数组时间复杂度 $O(m)$
模式匹配过程最坏时间复杂度 $O(n)$

next数组的优化思路



next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
	0	1	1	2	2	3

next数组的优化思路

主串S: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

模式串T: 1 2 3 4 5 6

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
0	1	1	2	2	3	

王道考研/CSKAOYAN.COM

next数组的优化思路

主串S: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

模式串T: 1 2 3 4 5 6

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
0	1	1	2	2	3	

王道考研/CSKAOYAN.COM

next数组的优化思路

主串S: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

模式串T: 1 2 3 4 5 6

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
0	1	1	2	2	3	

王道考研/CSKAOYAN.COM

next数组的优化思路

主串S: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

模式串T: 1 2 3 4 5 6

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
0	1	0	2	2	3	

王道考研/CSKAOYAN.COM

next数组的优化思路

主串S: [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]

模式串T: [a] [b] [a] [a] [b] [c]

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
0	1	0	2	2	3	

王道考研/CSKAOYAN.COM

next数组的优化思路

主串S: [a] [b] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]

模式串T: [a] [b] [a] [a] [b] [c]

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
0	1	0	2	2	3	

王道考研/CSKAOYAN.COM

next数组的优化思路

主串S: [a] [b] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]

模式串T: [a] [b] [a] [a] [b] [c]

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
0	1	0	2	2	3	

王道考研/CSKAOYAN.COM

next数组的优化思路

主串S: [a] [b] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]

模式串T: [a] [b] [a] [a] [b] [c]

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
0	1	0	2	2	3	

王道考研/CSKAOYAN.COM

next数组的优化思路

主串S: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

模式串T: 1 2 3 4 5 6

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
0	1	0	2	2	3	

王道考研/CSKAOYAN.COM

next数组的优化思路

主串S: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

模式串T: 1 2 3 4 5 6

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
0	1	0	2	2	3	

王道考研/CSKAOYAN.COM

next数组的优化思路

主串S: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

模式串T: 1 2 3 4 5 6

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
0	1	0	2	2	3	

王道考研/CSKAOYAN.COM

next数组的优化思路

主串S: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

模式串T: 1 2 3 4 5 6

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
0	1	0	2	2	3	

王道考研/CSKAOYAN.COM

next数组的优化思路

一定不是 b

主串S: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 a b a a ? ? ? ? ? ? ? ? ? ?

模式串T: a b a a b c
 1 2 3 4 5 6
 j ↑

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
	0	1	0	2	2	3

王道考研/CSKAOYAN.COM

next数组的优化思路

主串S: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?

模式串T: a b a a b c
 1 2 3 4 5 6
 j ↑

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
	0	1	0	2	1	3

王道考研/CSKAOYAN.COM

next数组的优化思路

主串S: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?

模式串T: a b a a b c
 1 2 3 4 5 6
 j ↑

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
	0	1	0	2	1	3

王道考研/CSKAOYAN.COM

next数组的优化思路

一定不是 b

主串S: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 a b a a ? ? ? ? ? ? ? ? ? ?

模式串T: a b a a b c
 1 2 3 4 5 6
 j ↑

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
	0	1	0	2	1	3

王道考研/CSKAOYAN.COM

next数组的优化思路

主串S: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 a b a a ? ? ? ? ? ? ? ? ? ?

模式串T: a b a a b c
 1 2 3 4 5 6

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
0	1	0	2	1	3	

一定不是 b

王道考研/CSKAOYAN.COM

next数组的优化思路

主串S: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?

模式串T: a b a a b c
 1 2 3 4 5 6

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
0	1	0	2	1	3	

王道考研/CSKAOYAN.COM

next数组的优化思路

主串S: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 ? ? ? ? ? ? ? ? ? ? ? ? ? ?

模式串T: a b a a b c
 1 2 3 4 5 6

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
0	1	0	2	1	3	

王道考研/CSKAOYAN.COM

next数组的优化思路

主串S: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 a b a a b ? ? ? ? ? ? ? ? ? ?

模式串T: a b a a b c
 1 2 3 4 5 6

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
0	1	0	2	1	3	

一定不是 c

王道考研/CSKAOYAN.COM

next数组的优化思路

但有可能是 a

主串: a b a a b ? ? ? ? ? ? ? ? ? ?

模式串: a b a a b c

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
0	1	0	2	1	3	

王道考研/CSKAOYAN.COM

KMP算法的进一步优化

根据模式串T, 求出 next 数组

利用next数组进行匹配 (主串指针不回溯)

使用nextval数组

T = 'abaabc'

next数组:

next[0]	next[1]	next[2]	next[3]	next[4]	next[5]	next[6]
0	1	1	2	2	3	

优化

nextval数组:

nextval[0]	nextval[1]	nextval[2]	nextval[3]	nextval[4]	nextval[5]	nextval[6]
0	1	0	2	1	3	

```
int Index_KMP(SString S, SString T, int next[]) {
    int i=1, j=1;
    while(i<=S.length&& j<=T.length){
        if(j==0 || S.ch[i]==T.ch[j]){
            ++i;
            ++j; //继续比较后继字符
        }
        else
            j=next[j]; //模式串向右移动
    }
    if(j>T.length)
        return i-T.length; //匹配成功
    else
        return 0;
}
```

王道考研/CSKAOYAN.COM

练习1: 求nextval数组

模式串 T = ababaa

ababaa

序号j	1	2	3	4	5	6
模式串	a	b	a	b	a	a
next[j]	0	1	1	2	3	4

手算解: 先求next数组, 再由next数组求nextval数组

```
nextval[1]=0;
for (int j=2; j<=T.length; j++) {
    if(T.ch[next[j]]==T.ch[j])
        nextval[j]=nextval[next[j]];
    else
        nextval[j]=next[j];
}
```

王道考研/CSKAOYAN.COM

练习1: 求nextval数组

模式串 T = ababaa

ababaa

序号j	1	2	3	4	5	6
模式串	a	b	a	b	a	a
nextval[j]	0	1	0	1	0	4

手算解: 先求next数组, 再由next数组求nextval数组

```
nextval[1]=0;
for (int j=2; j<=T.length; j++) {
    if(T.ch[next[j]]==T.ch[j])
        nextval[j]=nextval[next[j]];
    else
        nextval[j]=next[j];
}
```

王道考研/CSKAOYAN.COM

练习2: 求nextval数组

aaaab

模式串 T = aaaab

序号j	1	2	3	4	5
模式串	a	a	a	a	b
next[j]	0	1	2	3	4

手算解题: 先求next数组, 再由next数组求nextval数组

```
nextval[1]=0;
for (int j=2; j<=T.length; j++) {
    if(T.ch[next[j]]==T.ch[j])
        nextval[j]=nextval[next[j]];
    else
        nextval[j]=next[j];
}
```

王道考研/CSKAOYAN.COM

练习2: 求nextval数组

aaaab

模式串 T = aaaab

序号j	1	2	3	4	5
模式串	a	a	a	a	b
next[j]	0	1	2	3	4

手算解题: 先求next数组, 再由next数组求nextval数组

```
nextval[1]=0;
for (int j=2; j<=T.length; j++) {
    if(T.ch[next[j]]==T.ch[j])
        nextval[j]=nextval[next[j]];
    else
        nextval[j]=next[j];
}
```

王道考研/CSKAOYAN.COM

例: KMP算法 (优化前)

主串S: a a a c a a a a d a a a a b

模式串T: a a a a b

序号j	1	2	3	4	5
模式串	a	a	a	a	b
next[j]	0	1	2	3	4

王道考研/CSKAOYAN.COM

例: KMP算法 (优化前)

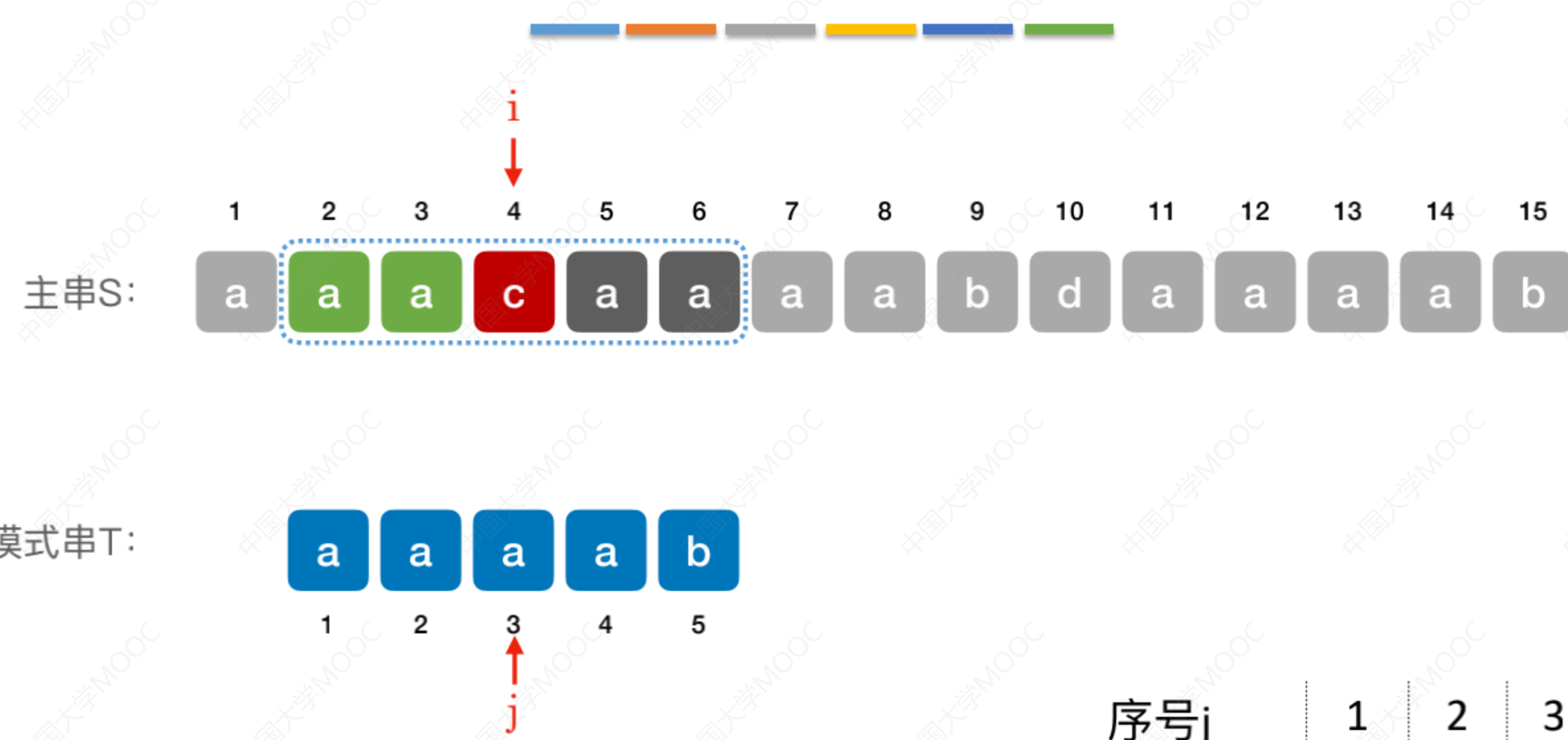
主串S: a a a c a a a a b d a a a a b

模式串T: a a a a b

序号j	1	2	3	4	5
模式串	a	a	a	a	b
next[j]	0	1	2	3	4

王道考研/CSKAOYAN.COM

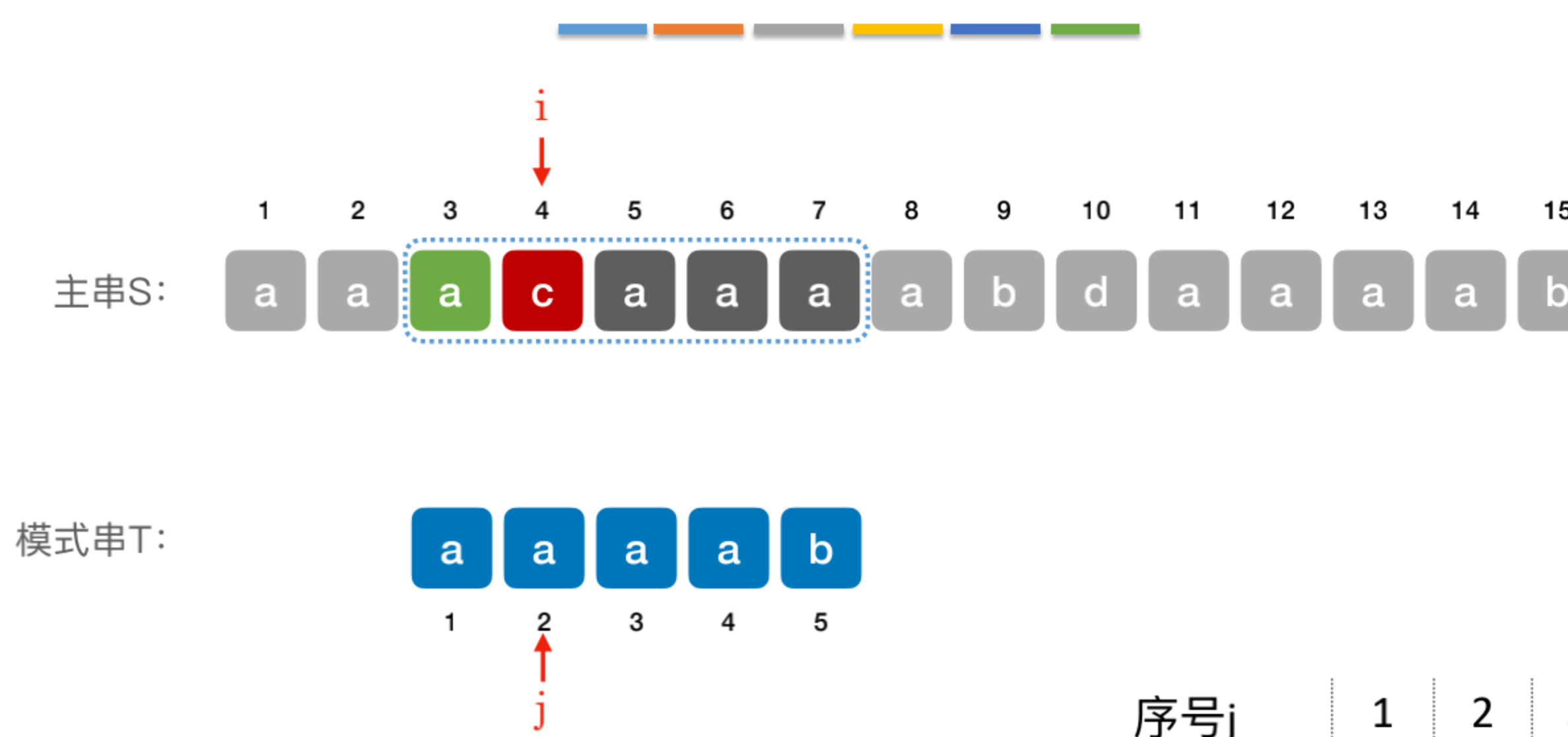
例：KMP算法（优化前）



序号j	1	2	3	4	5
模式串	a	a	a	a	b
next[j]	0	1	2	3	4

王道考研/CSKAOYAN.COM

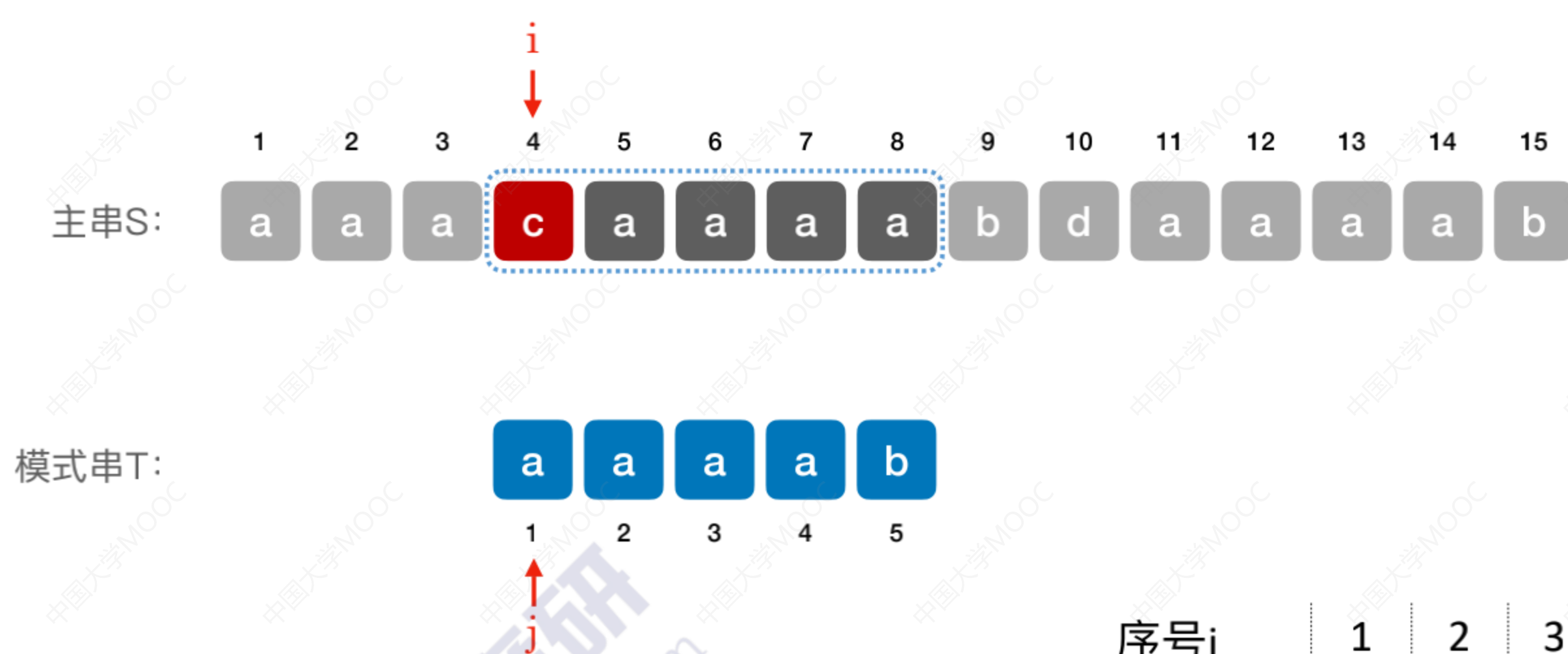
例：KMP算法（优化前）



序号j	1	2	3	4	5
模式串	a	a	a	a	b
next[j]	0	1	2	3	4

王道考研/CSKAOYAN.COM

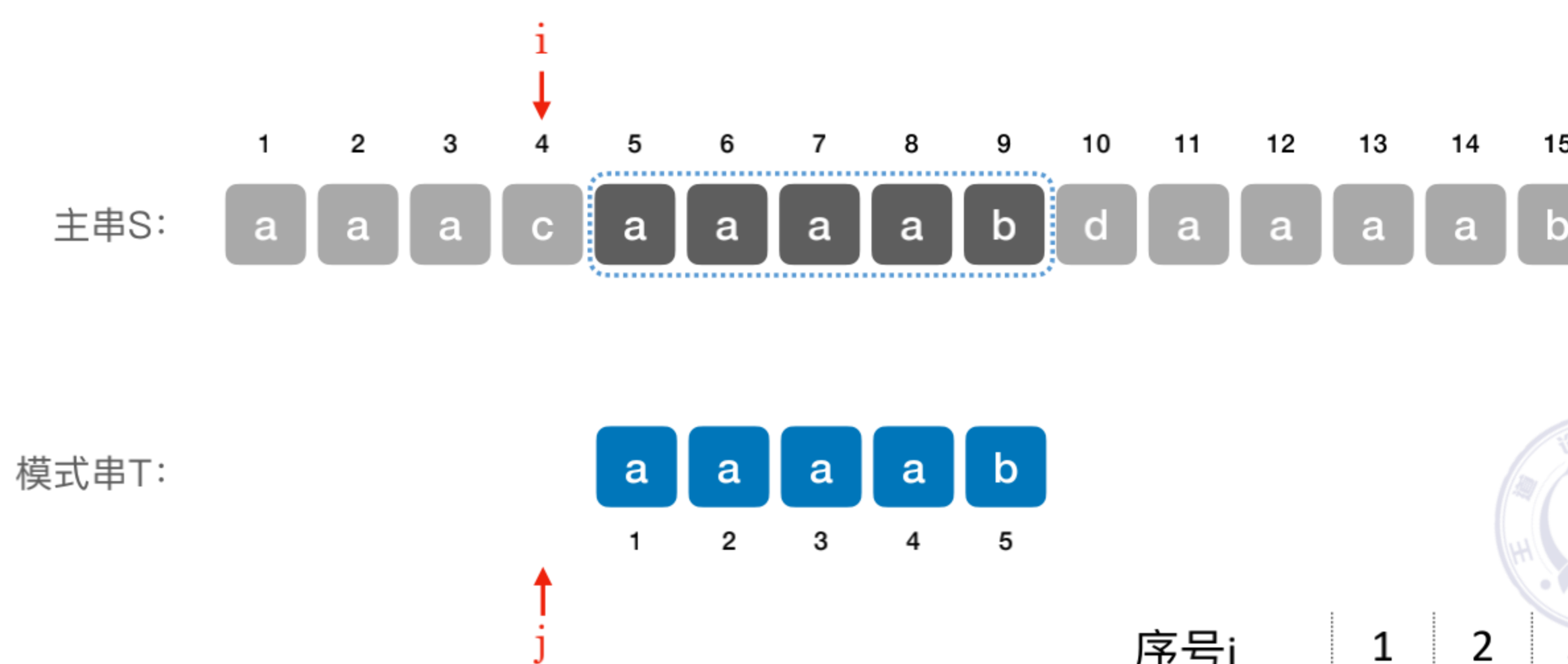
例：KMP算法（优化前）



序号j	1	2	3	4	5
模式串	a	a	a	a	b
next[j]	0	1	2	3	4

王道考研/CSKAOYAN.COM

例：KMP算法（优化前）



序号j	1	2	3	4	5
模式串	a	a	a	a	b
next[j]	0	1	2	3	4

王道考研/CSKAOYAN.COM

例：KMP算法（优化前）



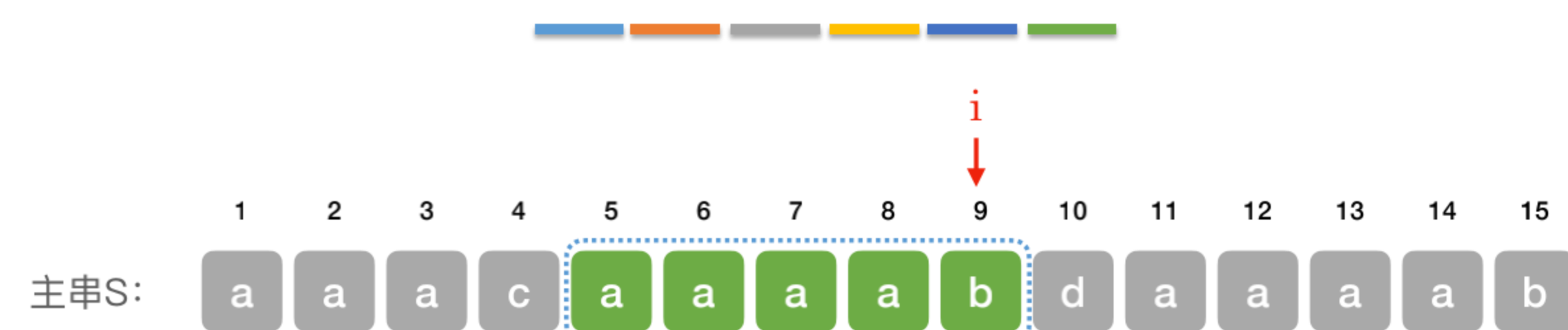
模式串T:



序号j	1	2	3	4	5
模式串	a	a	a	a	b
next[j]	0	1	2	3	4

王道考研/CSKAOYAN.COM

例：KMP算法（优化前）



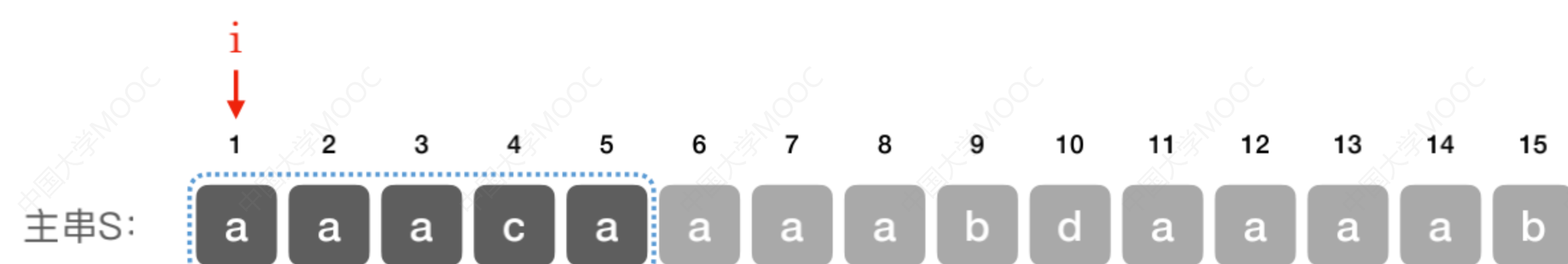
模式串T:



序号j	1	2	3	4	5
模式串	a	a	a	a	b
next[j]	0	1	2	3	4

王道考研/CSKAOYAN.COM

例：KMP算法（优化后）



模式串T:



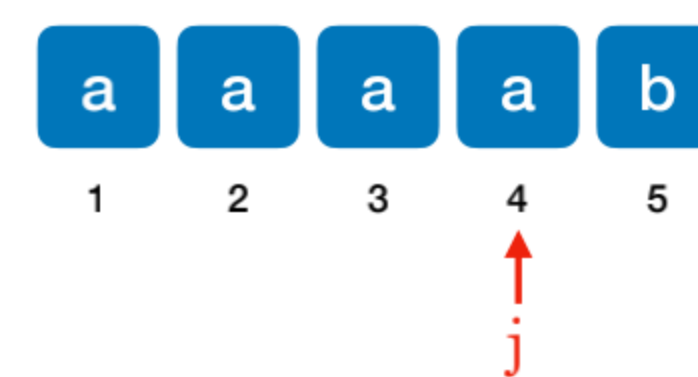
序号j	1	2	3	4	5
模式串	a	a	a	a	b
nextval[j]	0	0	0	0	4

王道考研/CSKAOYAN.COM

例：KMP算法（优化后）



模式串T:



序号j	1	2	3	4	5
模式串	a	a	a	a	b
nextval[j]	0	0	0	0	4

王道考研/CSKAOYAN.COM

例：KMP算法（优化后）

主串S: a a a c a a a a b d a a a a b

模式串T: a a a a b

序号j	1	2	3	4	5
模式串	a	a	a	a	b
nextval[j]	0	0	0	0	4

王道考研/CSKAOYAN.COM

例：KMP算法（优化后）

主串S: a a a c a a a a b d a a a a b

模式串T: a a a a b

序号j	1	2	3	4	5
模式串	a	a	a	a	b
nextval[j]	0	0	0	0	4

王道考研/CSKAOYAN.COM

例：KMP算法（优化后）

主串S: a a a c a a a a b d a a a a b

模式串T: a a a a b

序号j	1	2	3	4	5
模式串	a	a	a	a	b
nextval[j]	0	0	0	0	4

王道考研/CSKAOYAN.COM

补充学习：B站搜索“王道数据结构”-旧版 KMP

王道计算机考研 数据结构

441.0万播放 · 总弹幕数6.2万 2020-02-27 22:02:46

王道论坛 发消息
王道论坛www.cskaoyan.com
已关注 25.6万

弹幕列表 展开

视频选集 (37/90) 自动连播

P34 【旧版】4.2_1串的朴素模式匹配算法 11:33

P35 【旧版】4.2_2_KMP算法(上) 16:31

P36 【旧版】4.2_3_KMP算法(下) 17:31

H P37 【旧版】4.2_A_KMP 4.2_A_KMP算法(F)

P38 【旧版】5.1.1 树的定义和基本术语 15:17

P39 【旧版】5.1.2 树的性质 05:51

P40 【旧版】5.2_1 二叉树的定义和基本术语 12:46

P41 【旧版】5.2_2 二叉树的性质 02:30

P42 【旧版】5.2_3 二叉树的存储结构 10:45

P43 【旧版】5.3.1 二叉树的先中后序遍历 23:08

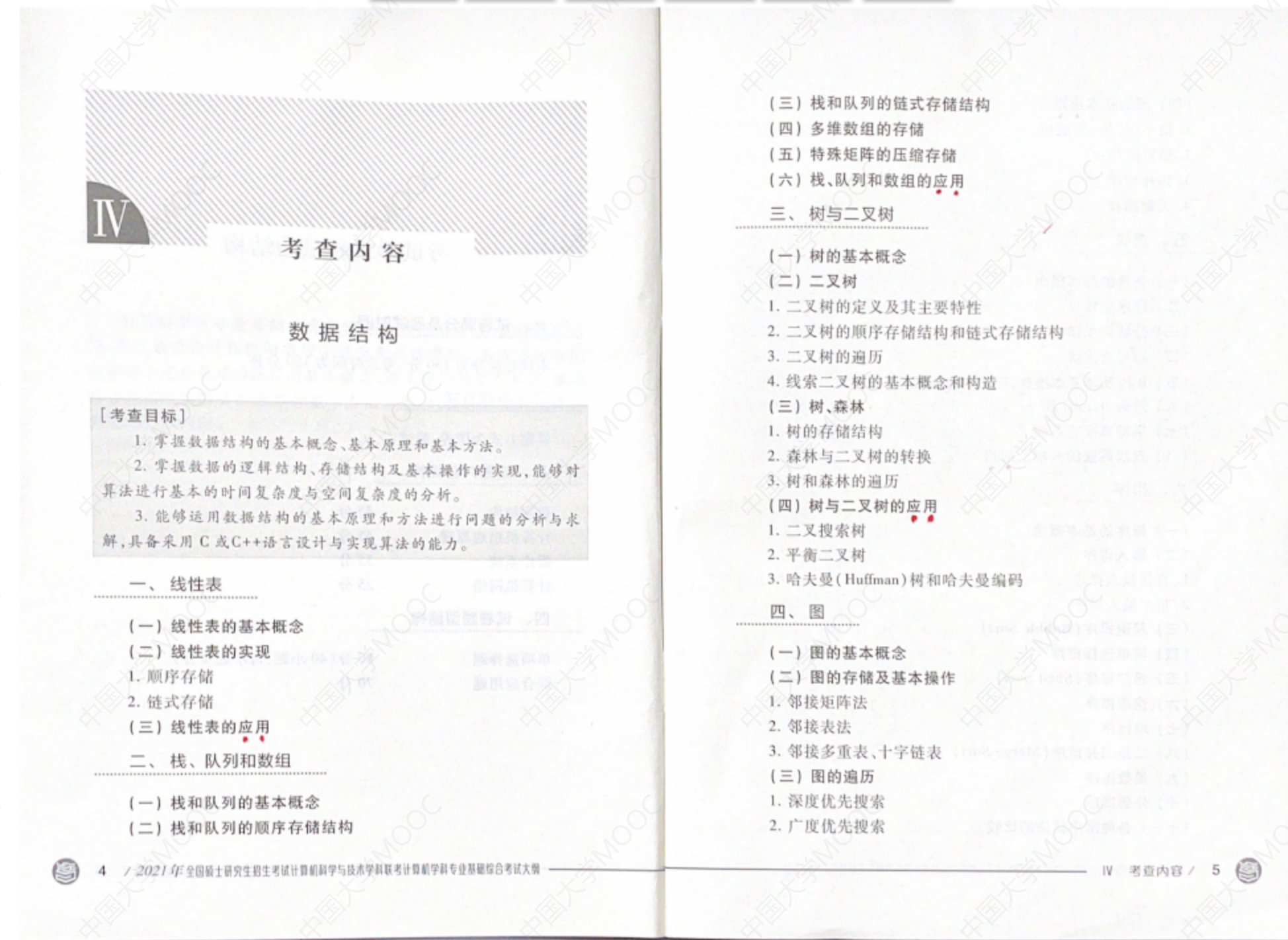
2023徐涛考研政治精讲【最新合集】 | B站独家 考研政治徐涛

序号j	1	2	3	4	5	6
模式串	g	o	o	g	l	e
next[j]	0	1	1	1	2	1

王道考研/CSKAOYAN.COM

王道考研/CSKAOYAN.COM

本章在408中的地位

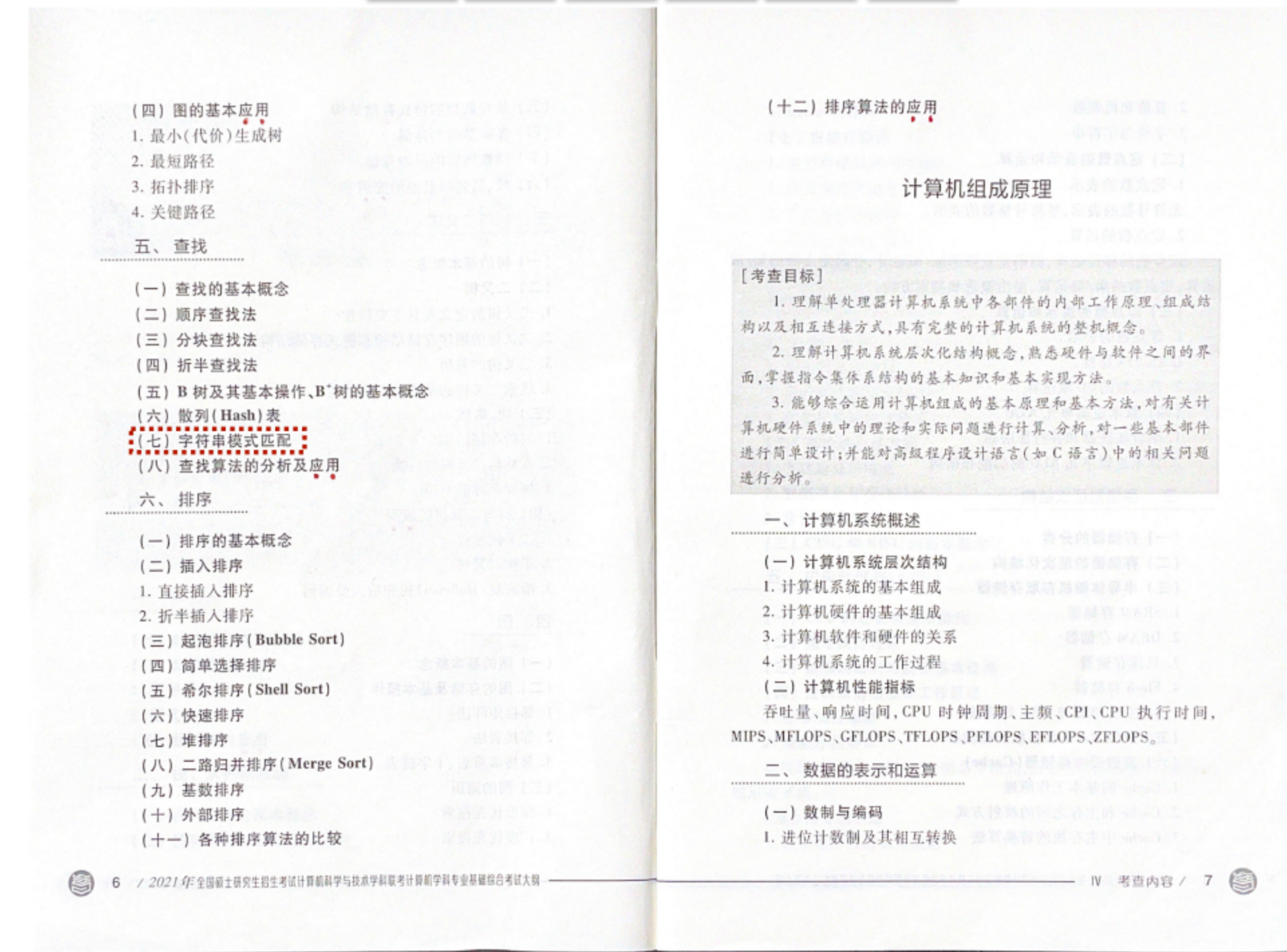


王道考研/CSKAOYAN.COM

本章在408中的地位



你好像没有任何牌面吧?



王道考研/CSKAOYAN.COM