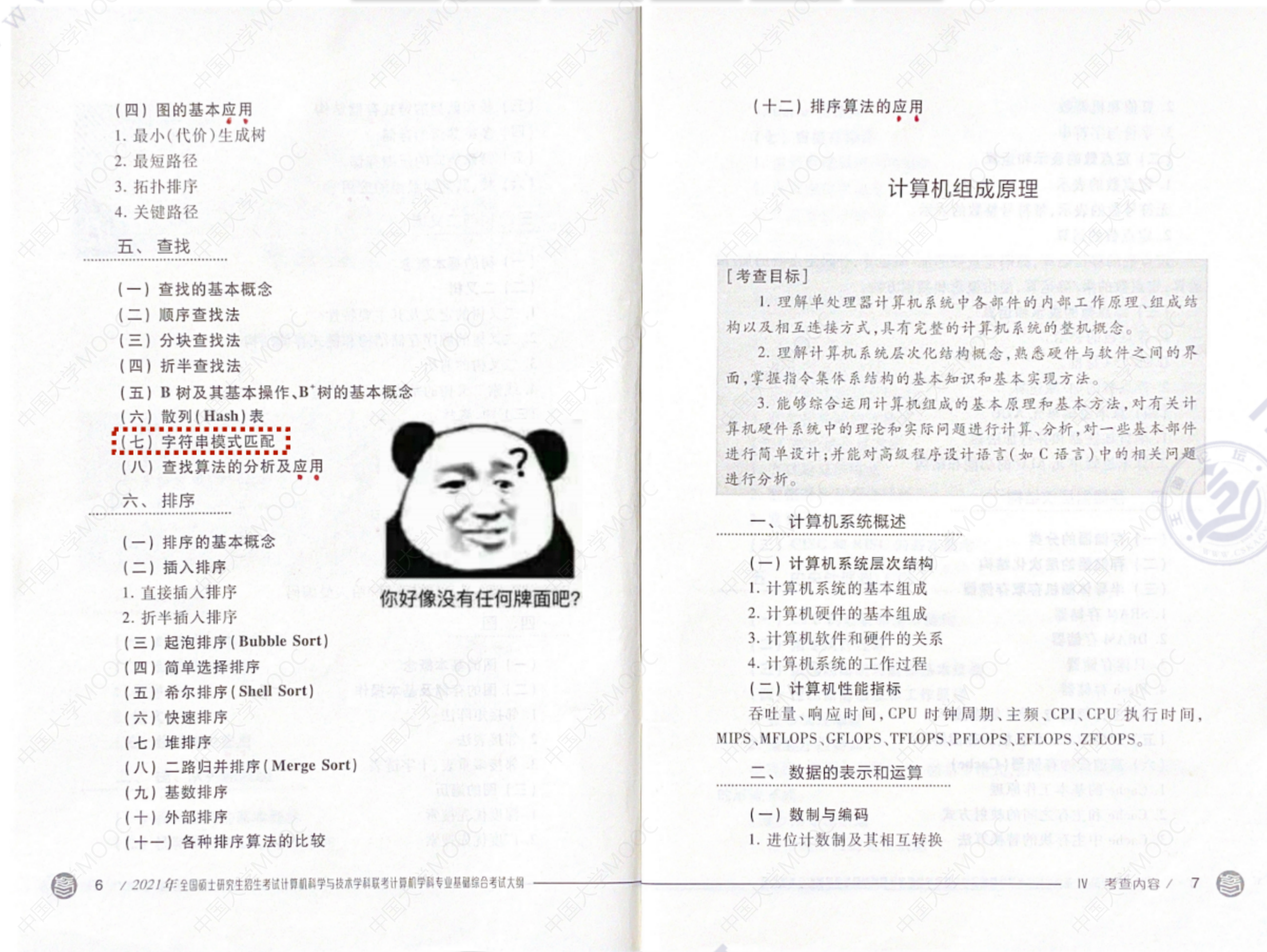


本章在408中的地位

小秘密：王道书每章开头有考纲



王道考研/CSKAOYAN.COM

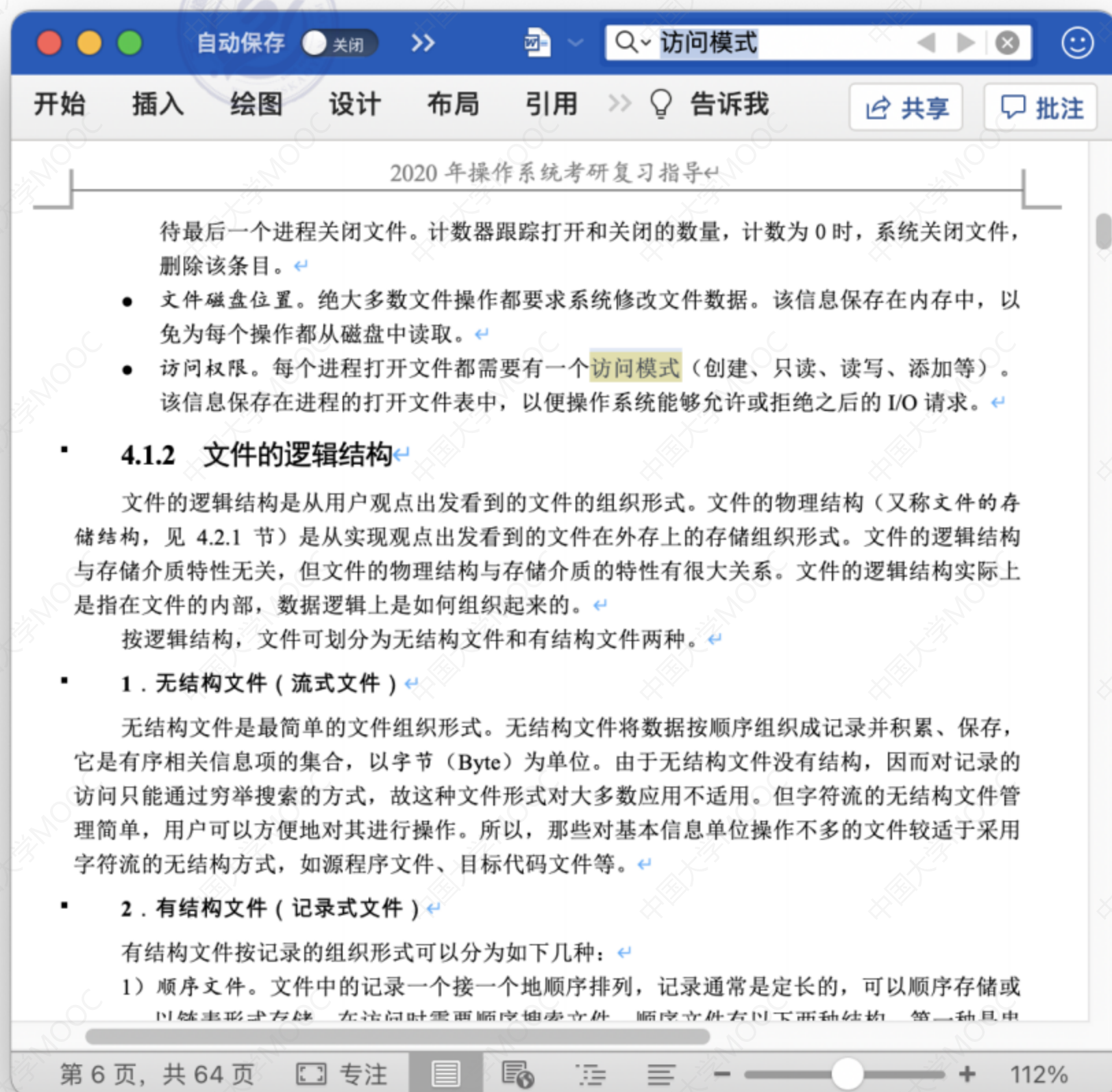
本节内容

字符串

朴素模式匹配算法

王道考研/CSKAOYAN.COM

什么是字符串的模式匹配



王道考研/CSKAOYAN.COM

什么是字符串的模式匹配

主串

‘嘿嘿嘿红红火火恍恍惚惚嗨皮开森猴开森笑出猪叫哈哈哈哈哈嗨森哈哈哈哈哈嗝’

模式串

‘笑出猪叫’

字符串模式匹配：在主串中找到与模式串相同的子串，并返回其所在位置。

王道考研/CSKAOYAN.COM

什么是字符串的模式匹配

与模式串匹配的子串

主串

‘嘿嘿嘿红红火火恍恍惚惚嗨皮开森猴开森笑出猪叫哈哈哈哈哈嗨森哈哈哈哈哈嗝’

模式串

‘笑出猪叫’

模式串

‘笑出喵叫’

子串——主串的一部分，一定存在
模式串——不一定能在主串中找到

字符串模式匹配：在主串中找到与模式串相同的子串，并返回其所在位置。

王道考研/CSKAOYAN.COM

两种模式匹配算法

字符串模式匹配

朴素模式匹配算法

KMP算法

王道考研/CSKAOYAN.COM

朴素模式匹配算法



主串S:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a	b	a	a	b	a	a	b	c	a	b	a	a	b	c

模式串T:

a	b	a	a	b	c
1	2	3	4	5	6



开始暴力解决问题了

王道考研/CSKAOYAN.COM

朴素模式匹配算法

主串S:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a	b	a	a	b	a	a	b	c	a	b	a	a	b	c

模式串T:

a	b	a	a	b	c
1	2	3	4	5	6



开始暴力解决问题了

王道考研/CSKAOYAN.COM

朴素模式匹配算法

主串S: a b a a b a a b c a b a a b c

模式串T: a b a a b c



开始暴力解决问题了

王道考研/CSKAOYAN.COM

朴素模式匹配算法

主串S: a b a a b a a b c a b a a b c

模式串T: a b a a b c



开始暴力解决问题了

王道考研/CSKAOYAN.COM

朴素模式匹配算法



主串S: a b a a b a a b c a b a a b c

模式串T: a b a a b c



开始暴力解决问题了

王道考研/CSKAOYAN.COM

朴素模式匹配算法

主串S: a b a a b a a b c a b a a b c

模式串T: a b a a b c



开始暴力解决问题了

王道考研/CSKAOYAN.COM

朴素模式匹配算法

主串S: a b a a b a a b c a b a a b c

模式串T: a b a a b c



开始暴力解决问题了

王道考研/CSKAOYAN.COM

朴素模式匹配算法

主串S: a b a a b a a b c a b a a b c

模式串T: a b a a b c



开始暴力解决问题了

王道考研/CSKAOYAN.COM

朴素模式匹配算法



主串S: a b a a b a a b c a b a a b c

模式串T: a b a a b c



开始暴力解决问题了

王道考研/CSKAOYAN.COM

朴素模式匹配算法

主串S: a b a a b a a b c a b a a b c

模式串T: a b a a b c



开始暴力解决问题了

王道考研/CSKAOYAN.COM

朴素模式匹配算法

主串S: a b a a b a a b c a b a a b c

模式串T: a b a a b c



开始暴力解决问题了

王道考研/CSKAOYAN.COM

朴素模式匹配算法

主串S: a b a a b a a b c a b a a b c

模式串T: a b a a b c



开始暴力解决问题了

主串长度为 n ，模式串长度为 m

最多对比 $n-m+1$ 个子串

朴素模式匹配算法：将主串中所有长度为 m 的子串依次与模式串对比，直到找到一个完全匹配的子串，或所有的子串都不匹配为止。

王道考研/CSKAOYAN.COM

朴素模式匹配算法

Index(S,T): 定位操作。若主串S中存在与串T值相同的子串, 则返回它的主串S中第一次出现的位置; 否则函数值为0。

```
int Index(SString S, SString T){
    int i=1, n=StrLength(S), m=StrLength(T);
    SString sub; //用于暂存子串
    while(i<=n-m+1){
        SubString(sub,S,i,m);
        if(StrCompare(sub, T)!=0) ++i;
        else return i; //返回子串在主串中的位置
    }
    return 0; //S中不存在与T相等的子串
}
```

最多对比 $n-m+1$ 个子串

取出从位置 i 开始, 长度为 m 的子串

子串和模式串对比, 若不匹配, 则匹配下一个子串

原来是这样啊



接下来: 不使用字符串的基本操作, 直接通过数组下标实现朴素模式匹配算法

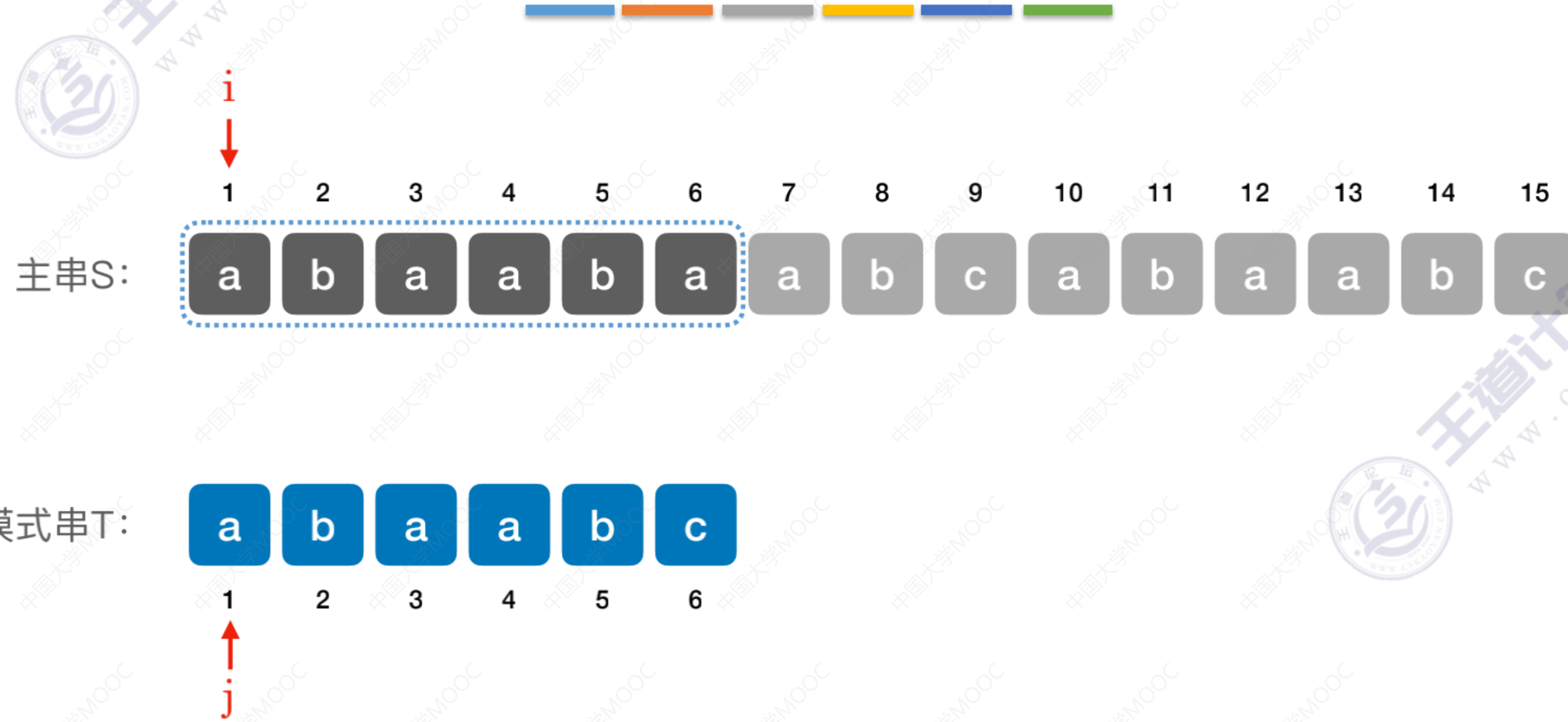
王道考研/CSKAOYAN.COM

朴素模式匹配算法

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

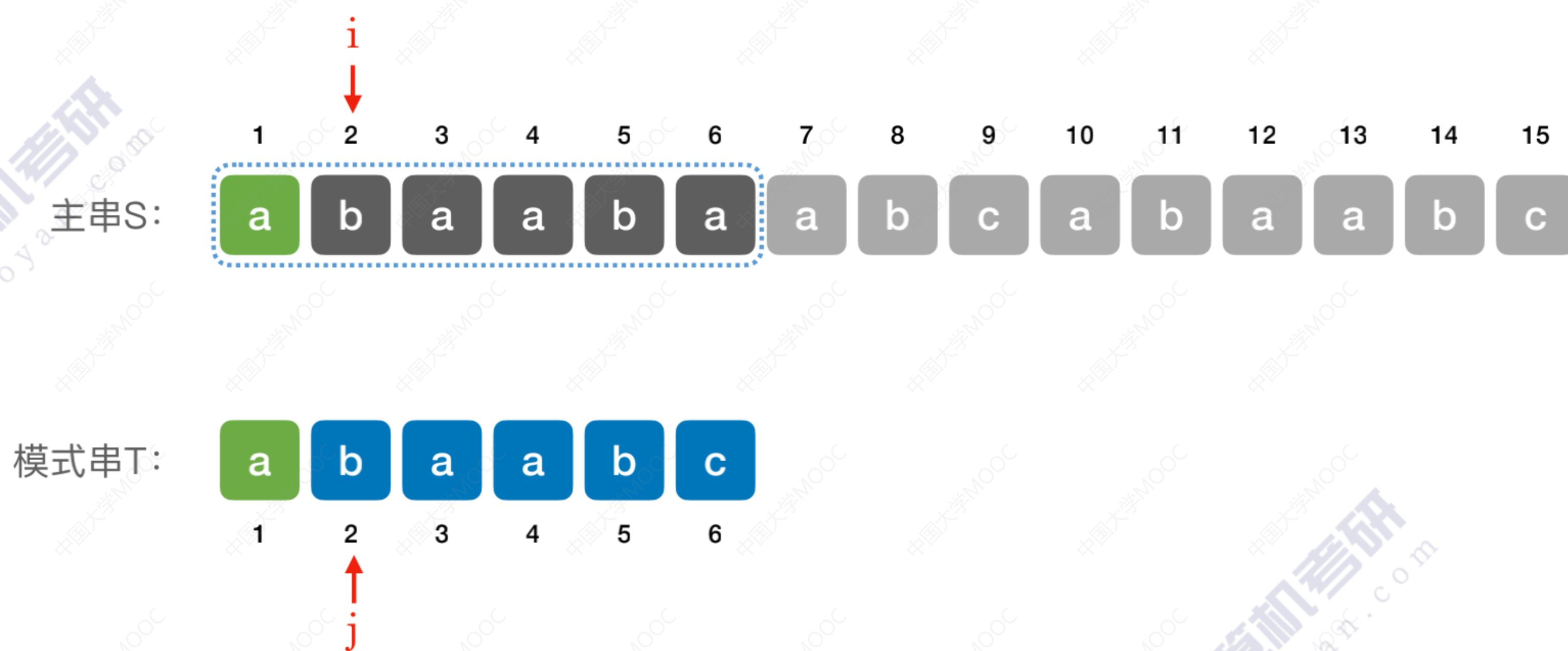
王道考研/CSKAOYAN.COM

朴素模式匹配算法



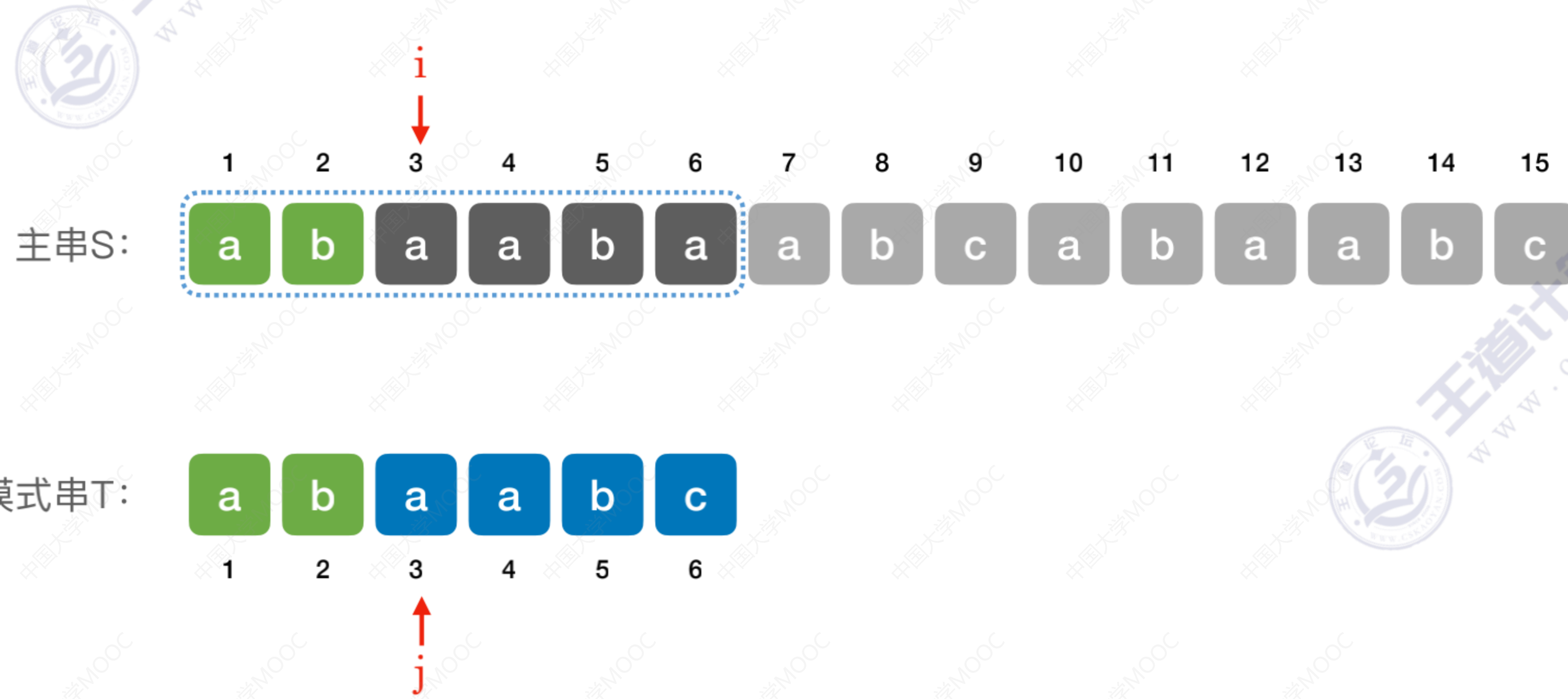
王道考研/CSKAOYAN.COM

朴素模式匹配算法



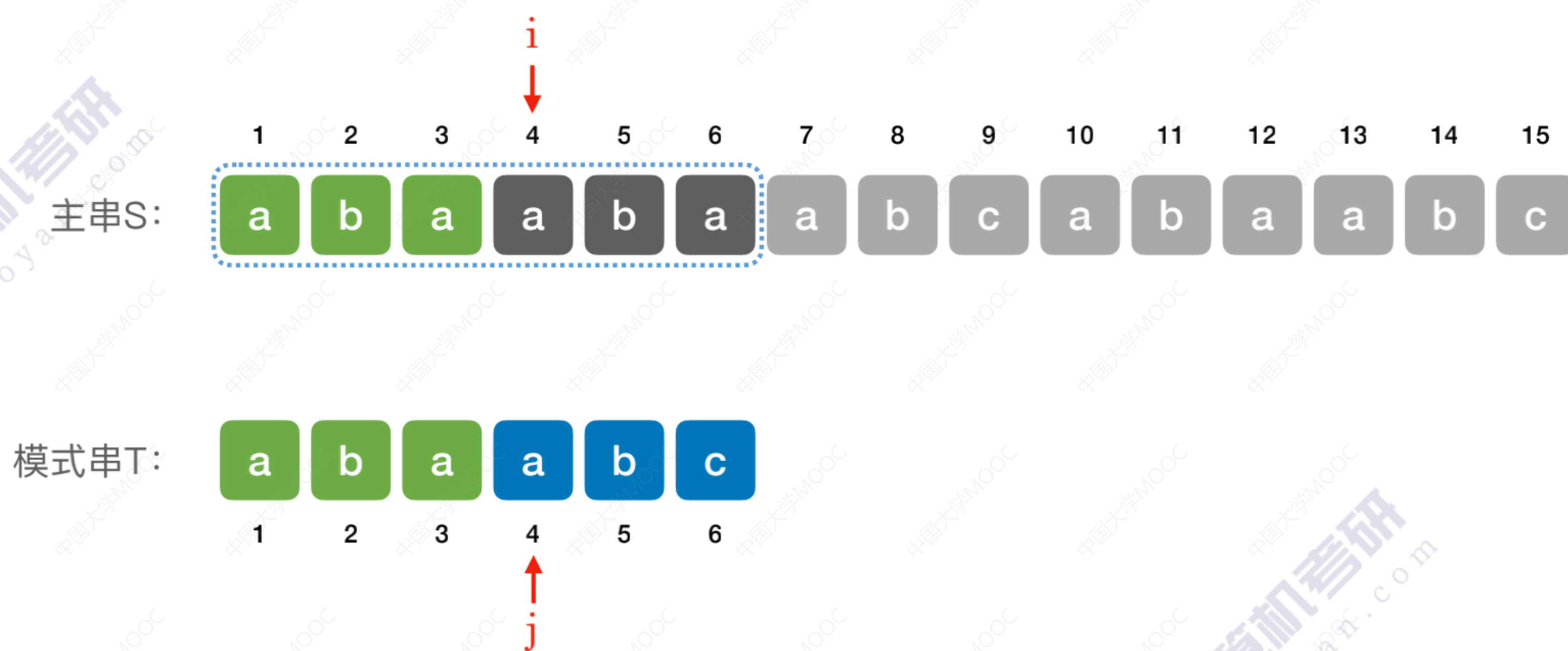
王道考研/CSKAOYAN.COM

朴素模式匹配算法



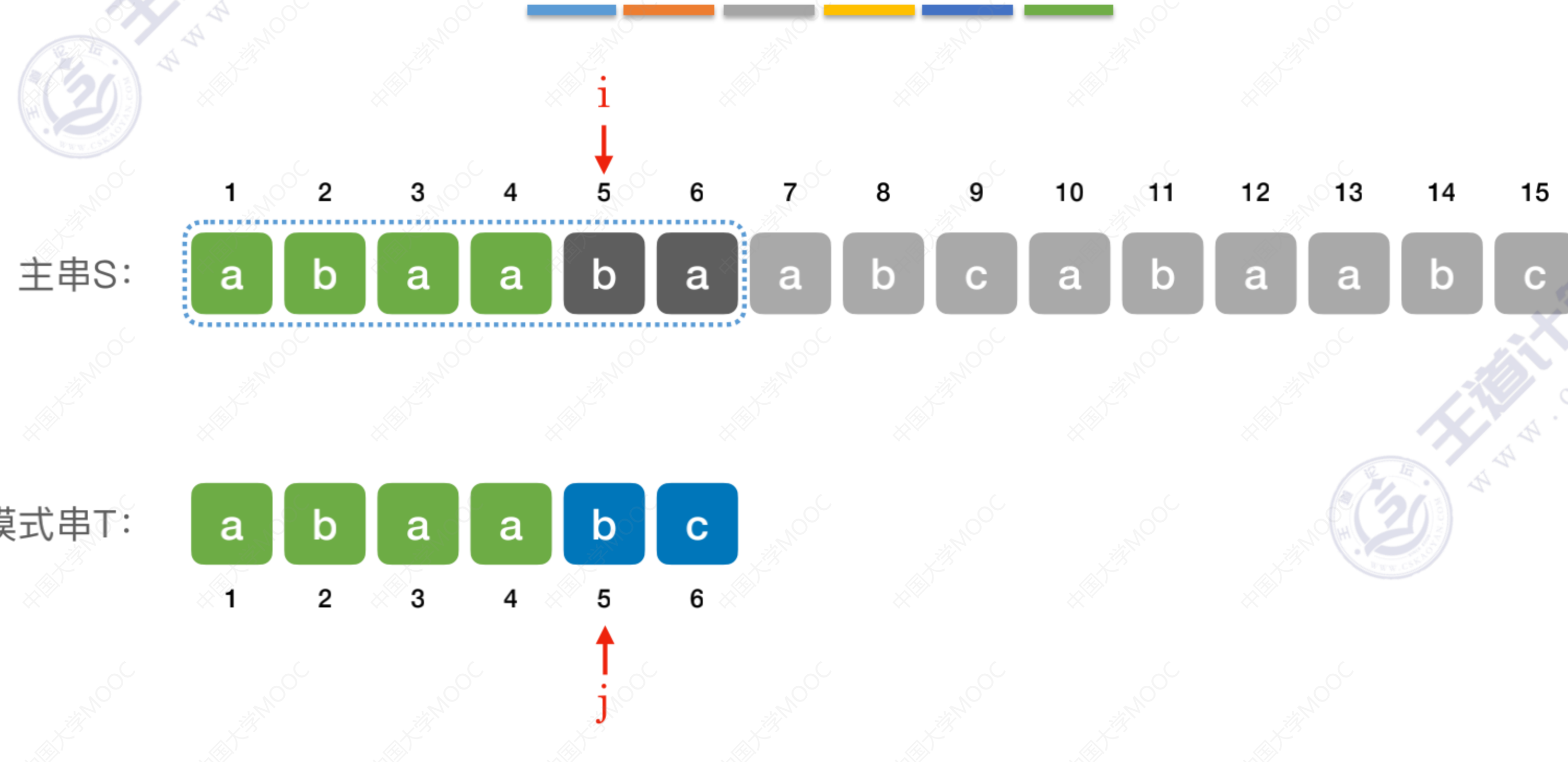
王道考研/CSKAOYAN.COM

朴素模式匹配算法

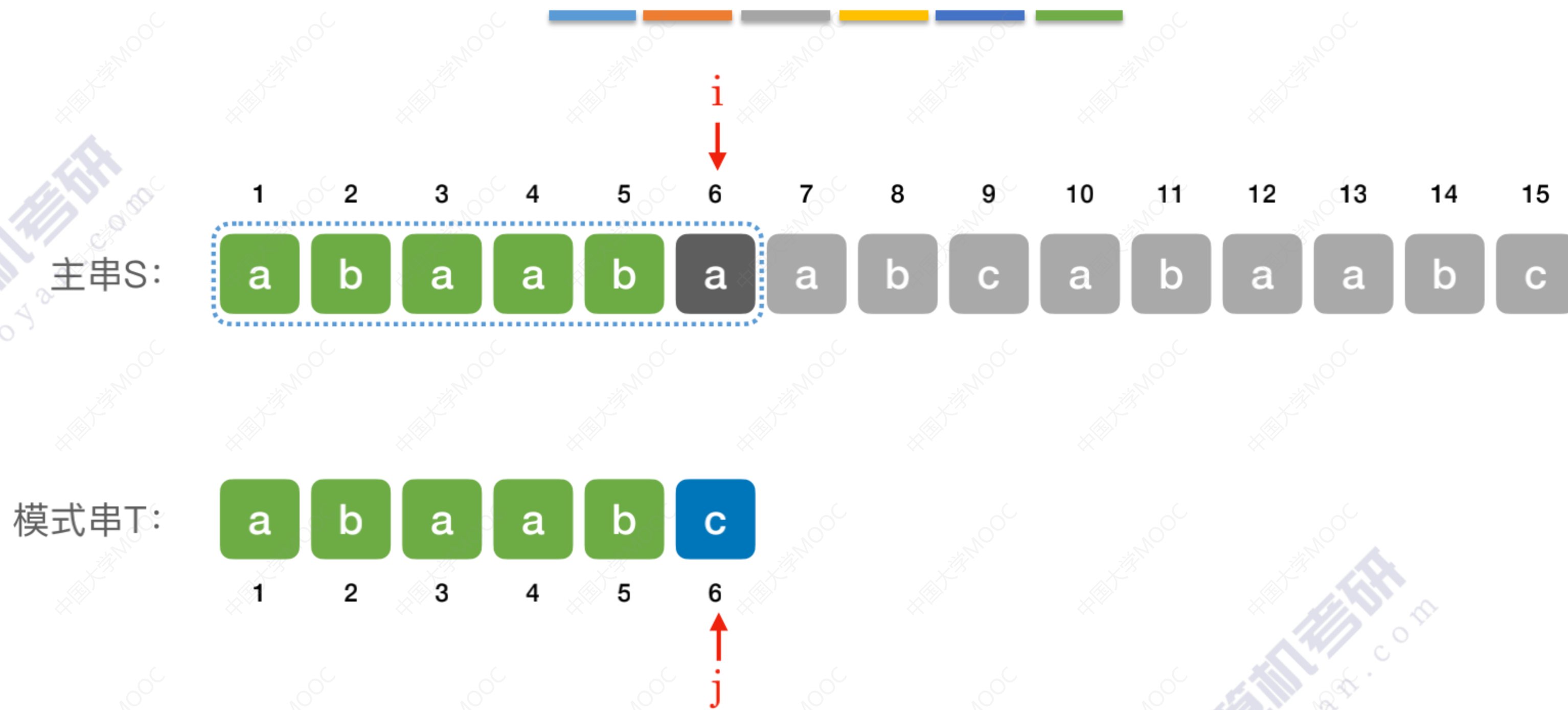


王道考研/CSKAOYAN.COM

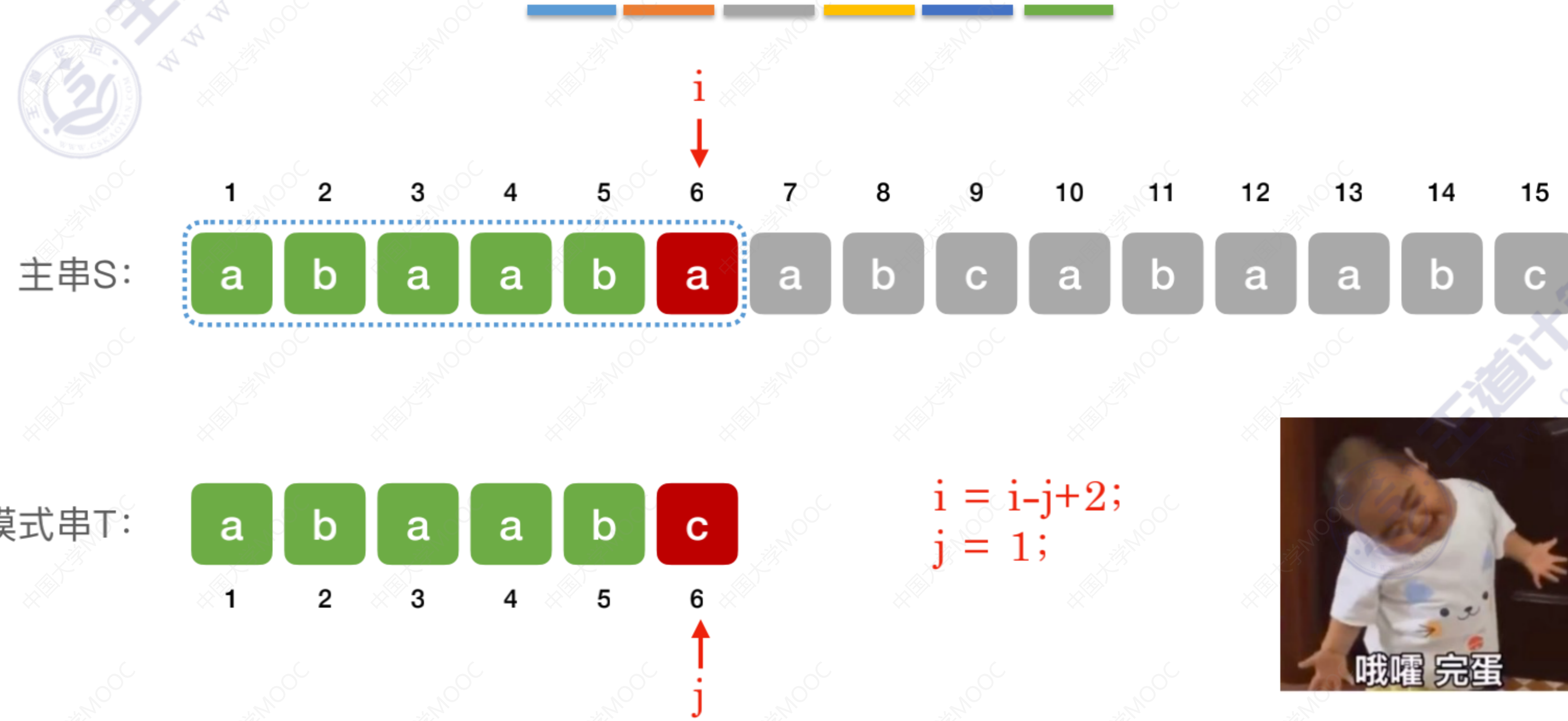
朴素模式匹配算法



朴素模式匹配算法



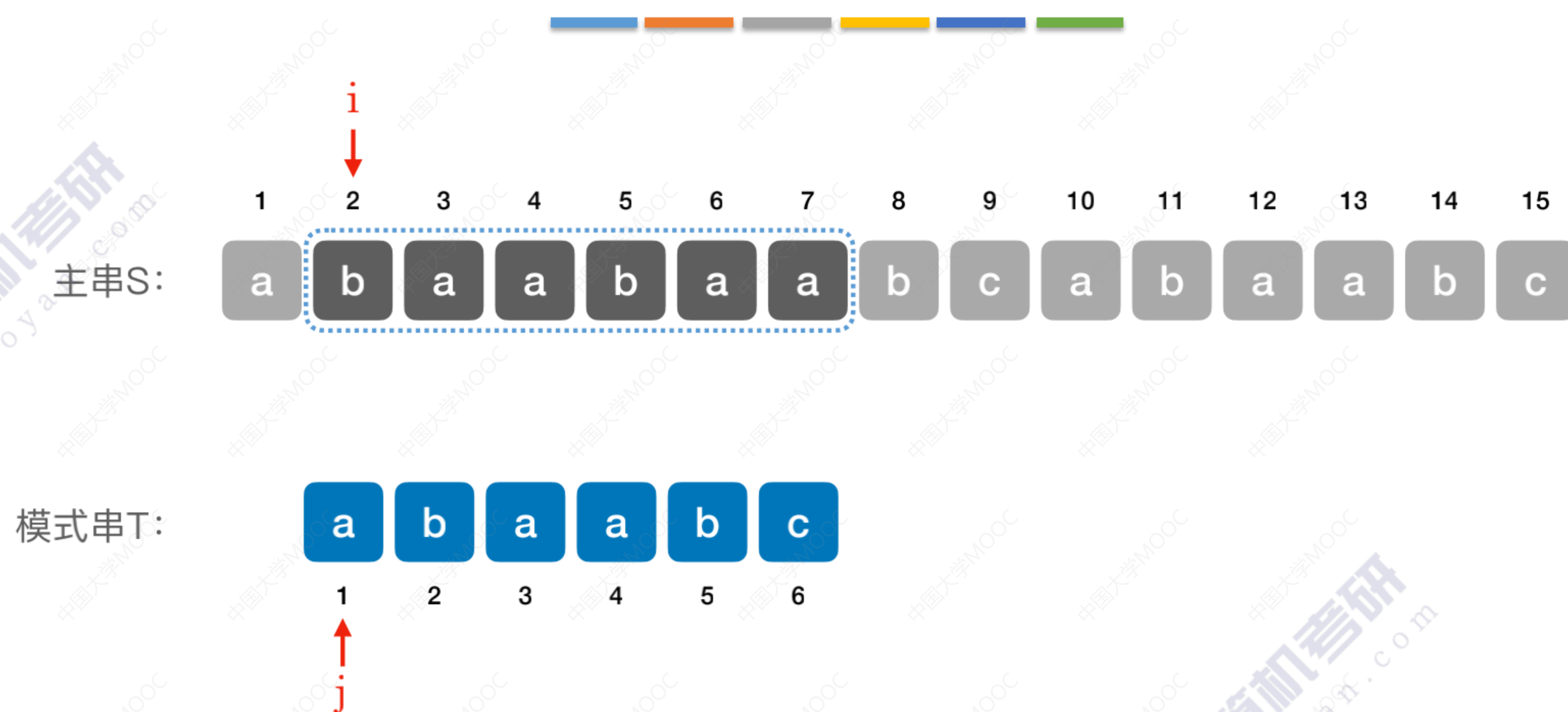
朴素模式匹配算法



若当前子串匹配失败，则主串指针 i 指向下一个子串的第一个位置，模式串指针 j 回到模式串的第一个位置

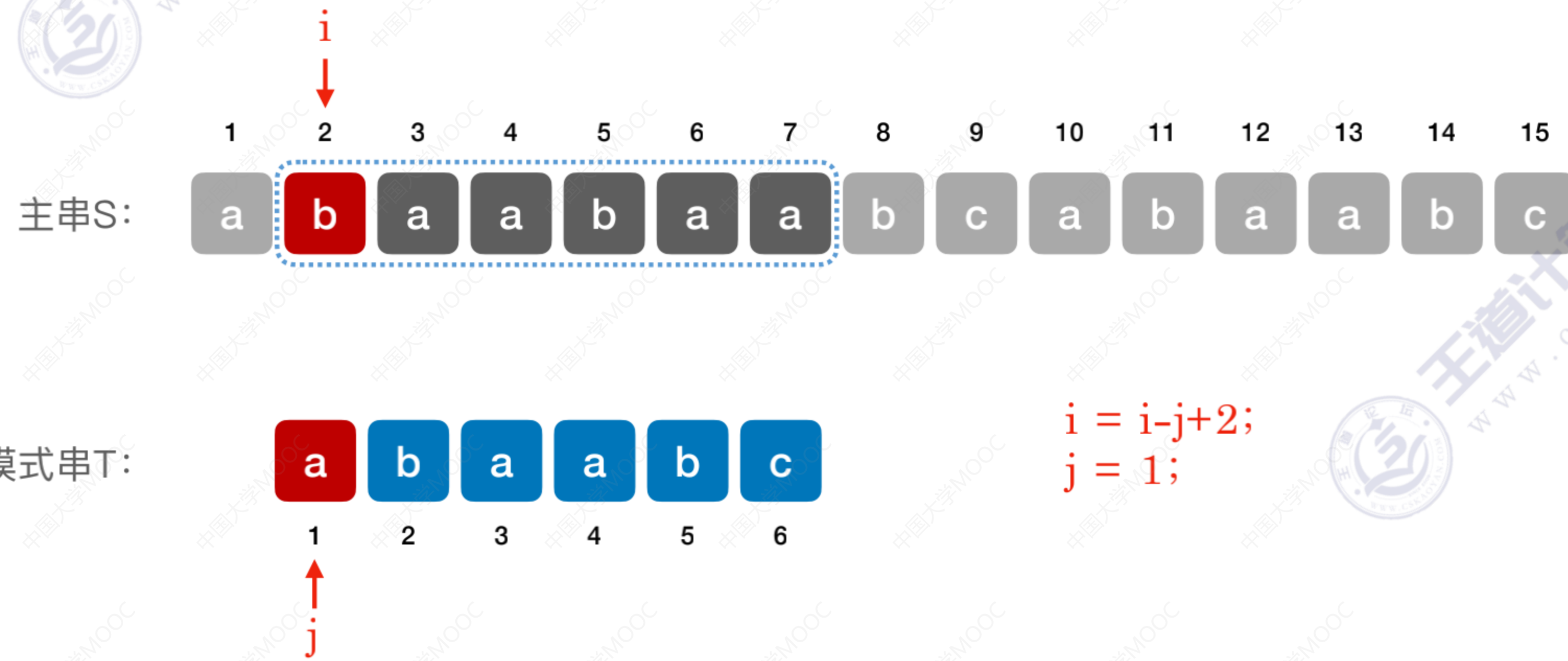
王道考研/CSKAOYAN.COM

朴素模式匹配算法



王道考研/CSKAOYAN.COM

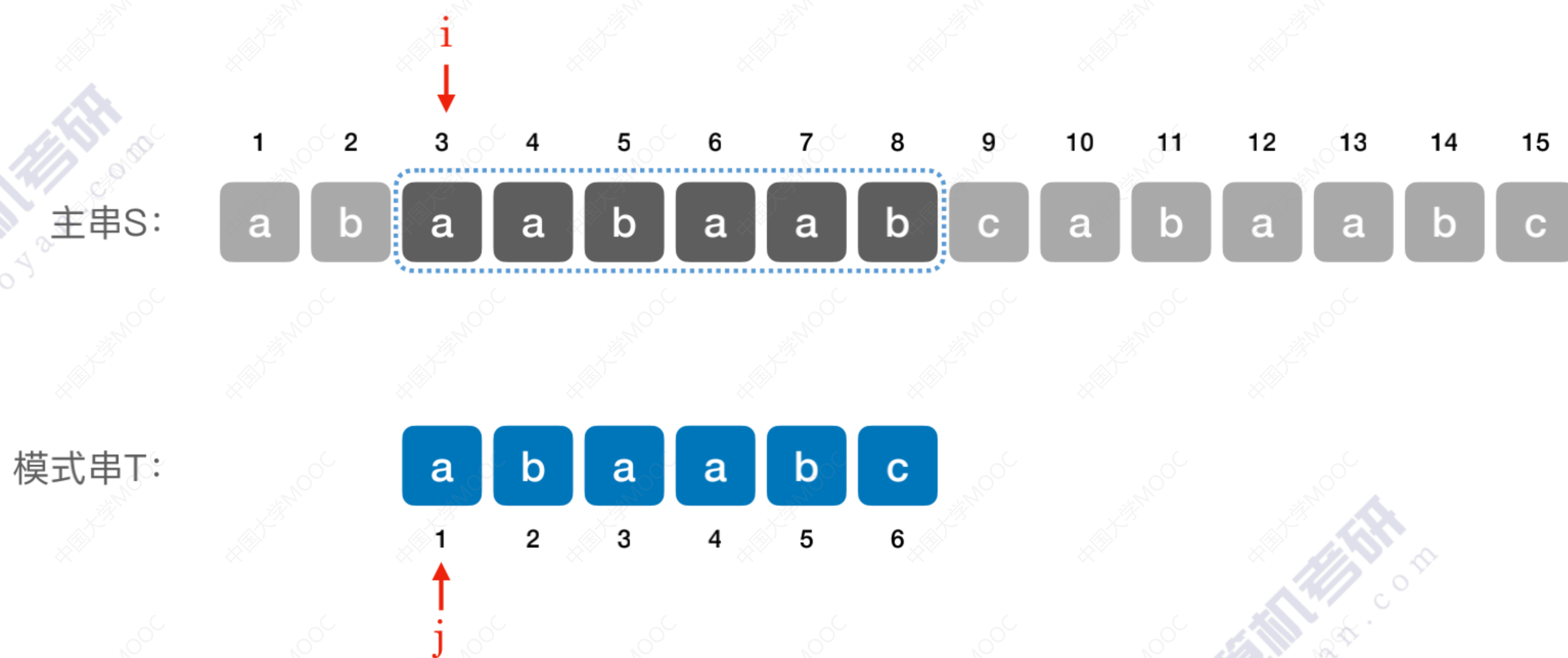
朴素模式匹配算法



若当前子串匹配失败，则主串指针 i 指向下一个子串的第一个位置，模式串指针 j 回到模式串的第一个位置

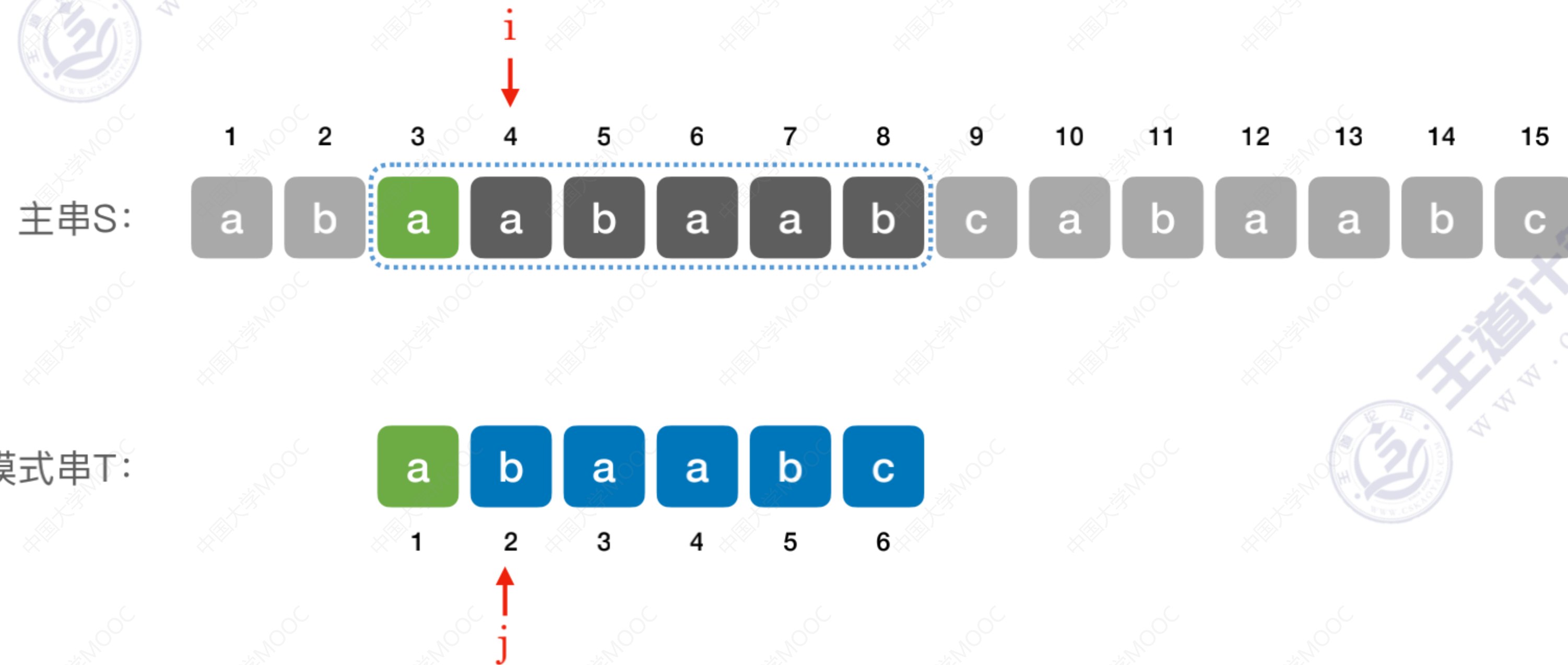
王道考研/CSKAOYAN.COM

朴素模式匹配算法



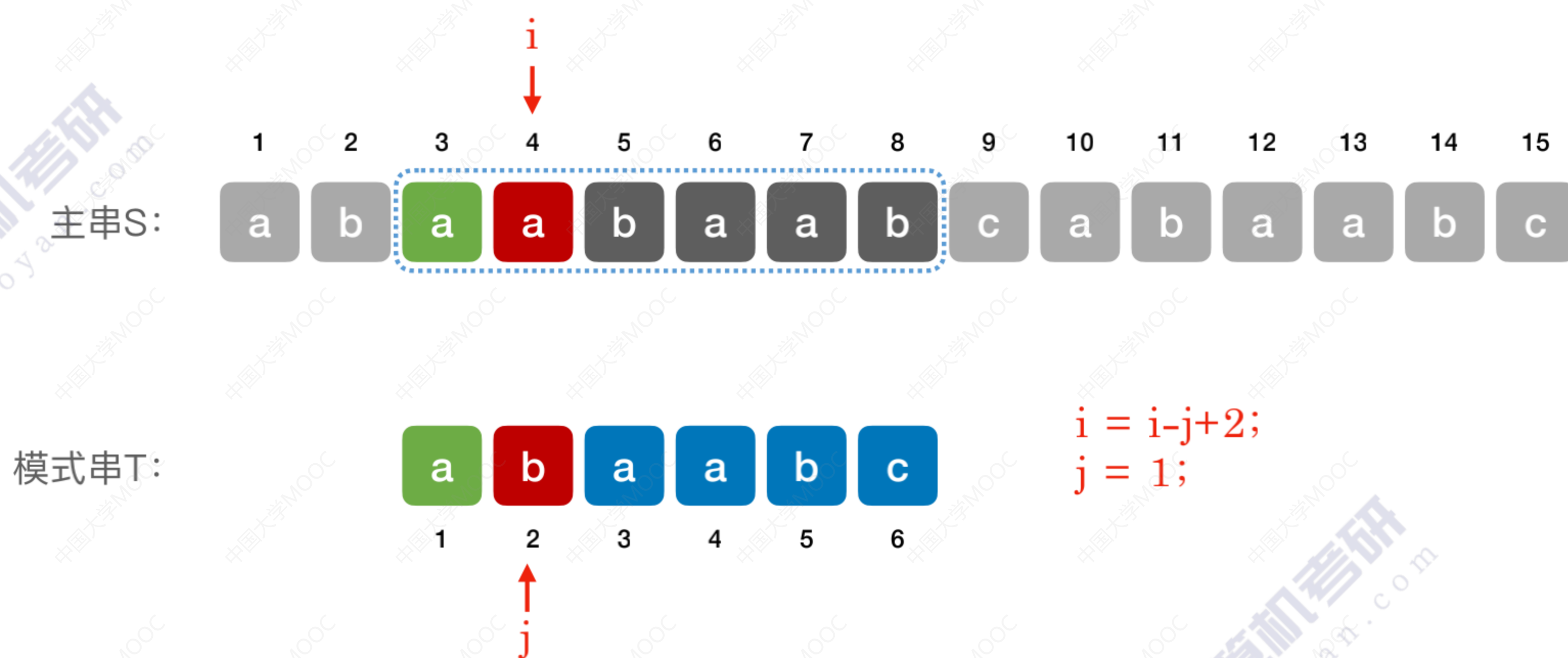
王道考研/CSKAOYAN.COM

朴素模式匹配算法



王道考研/CSKAOYAN.COM

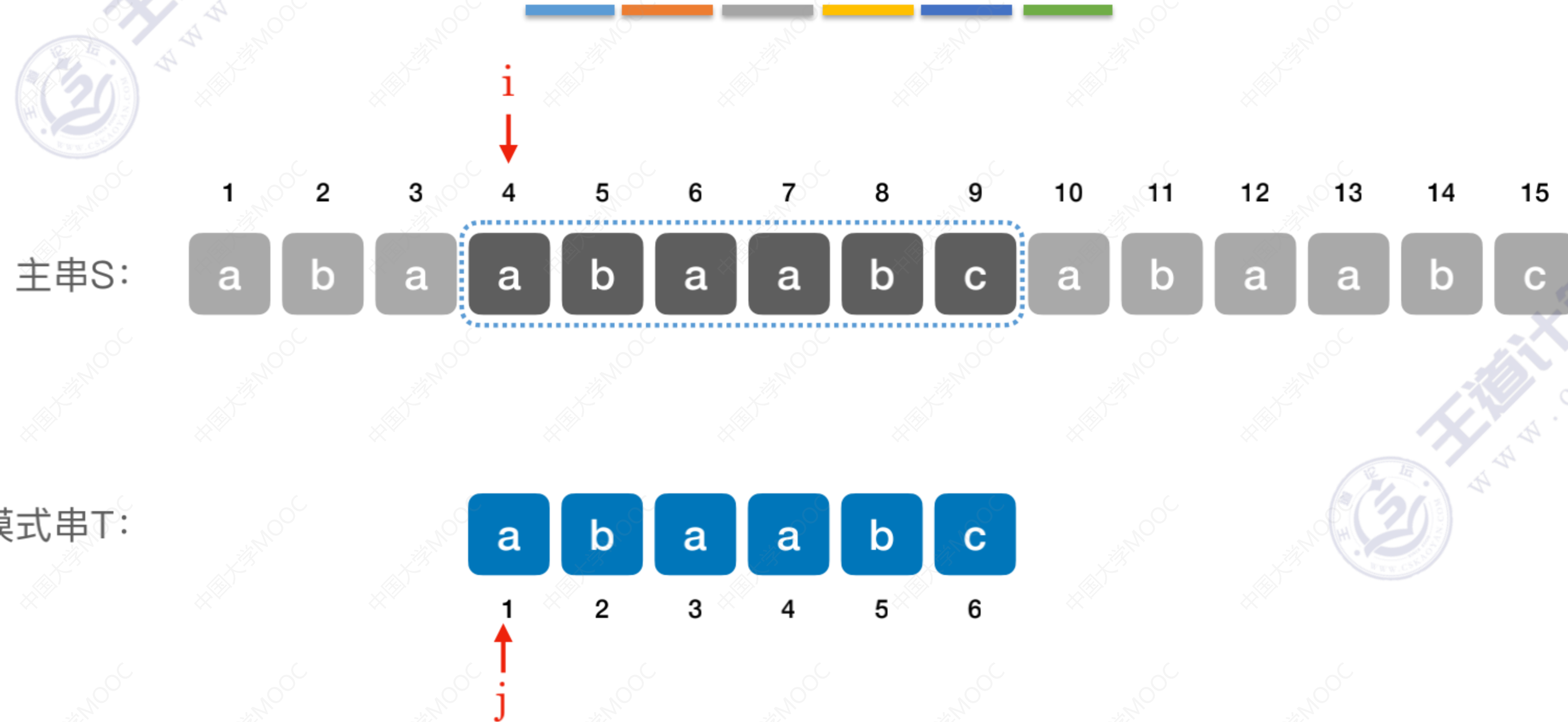
朴素模式匹配算法



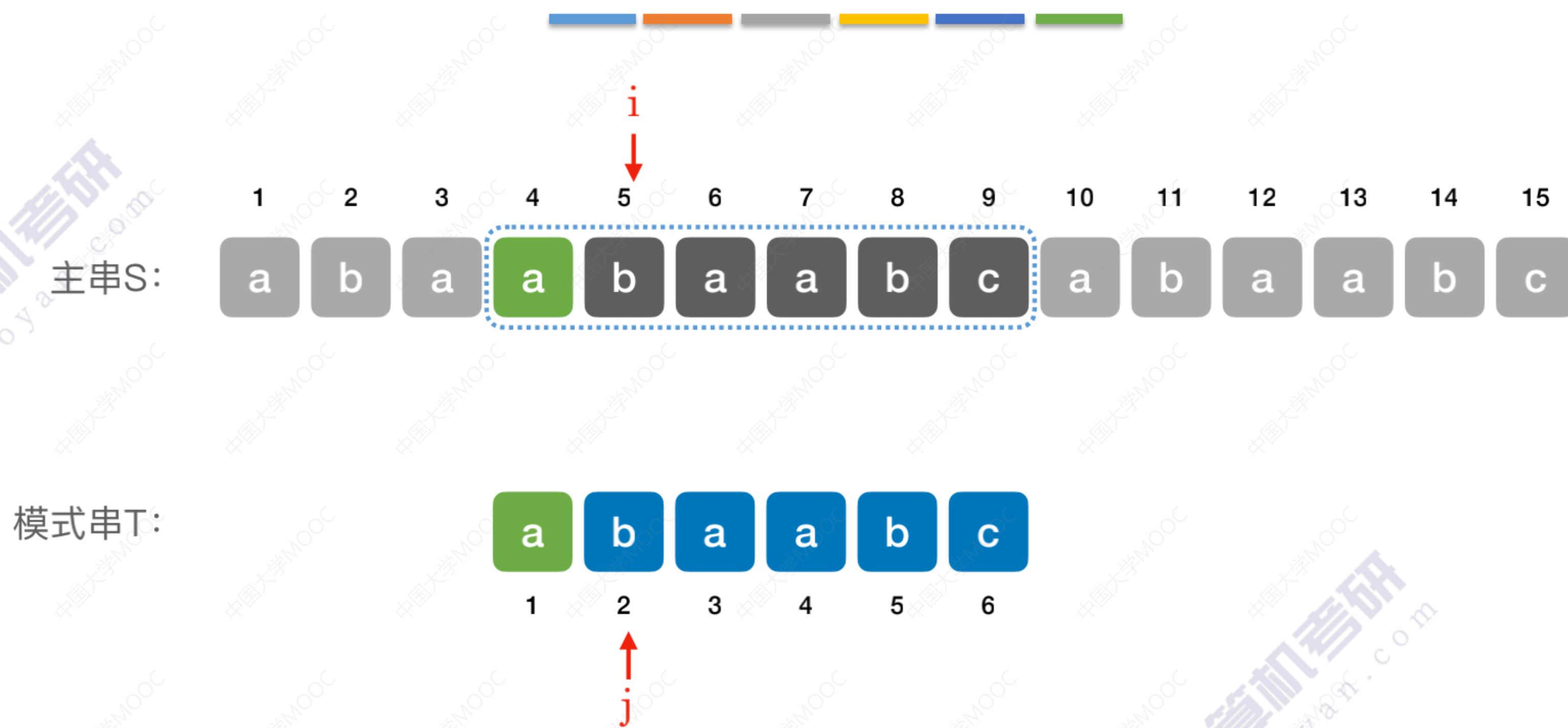
若当前子串匹配失败，则主串指针 i 指向下一个子串的第一个位置，模式串指针 j 回到模式串的第一个位置

王道考研/CSKAOYAN.COM

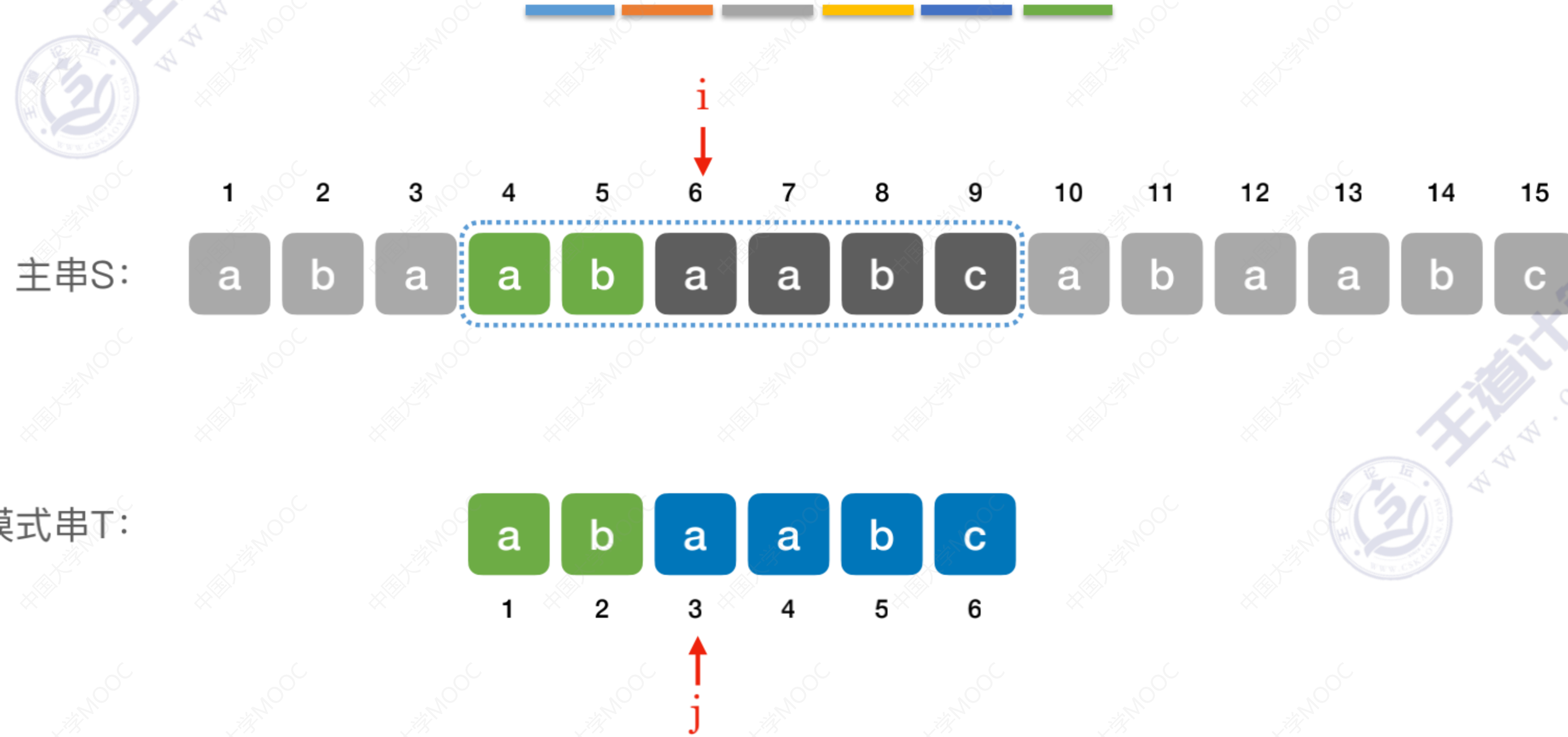
朴素模式匹配算法



朴素模式匹配算法

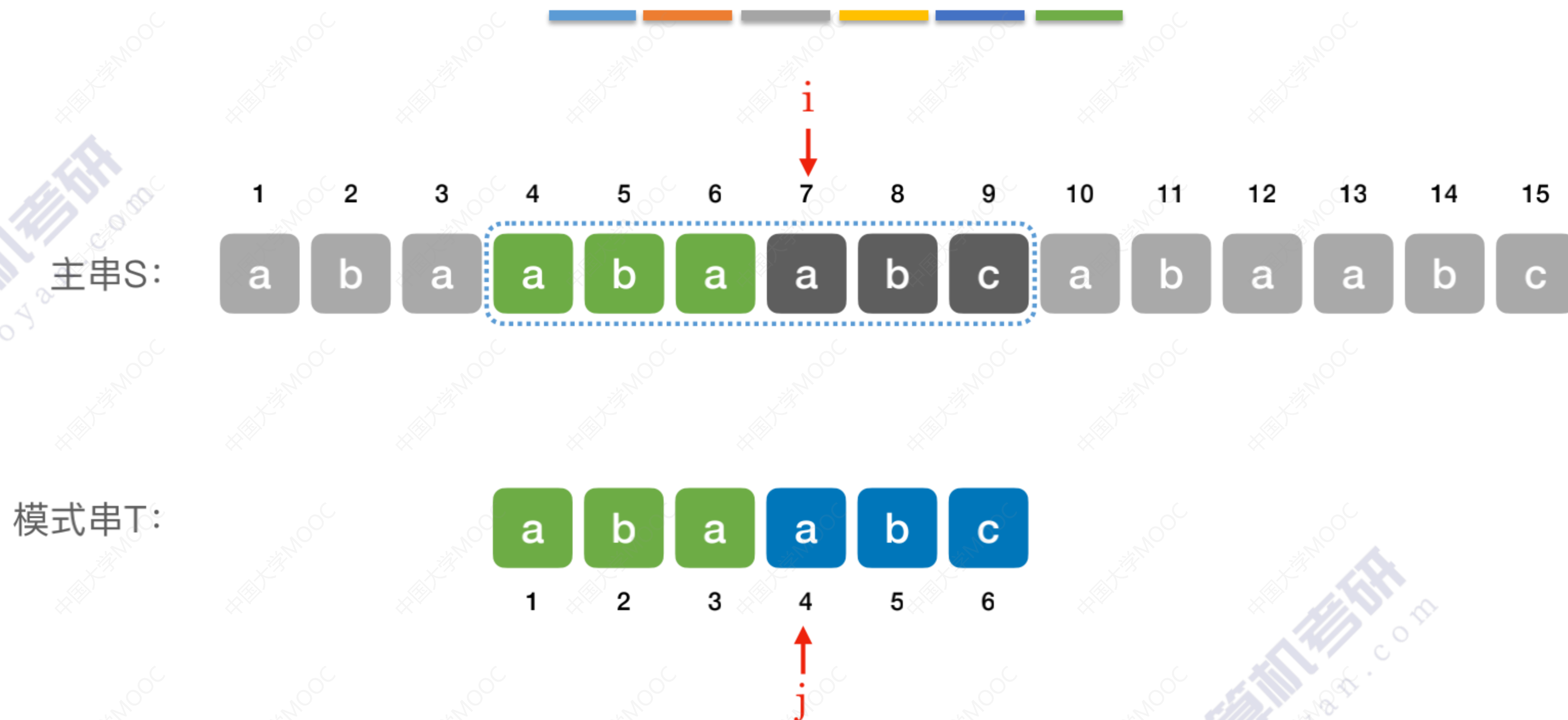


朴素模式匹配算法



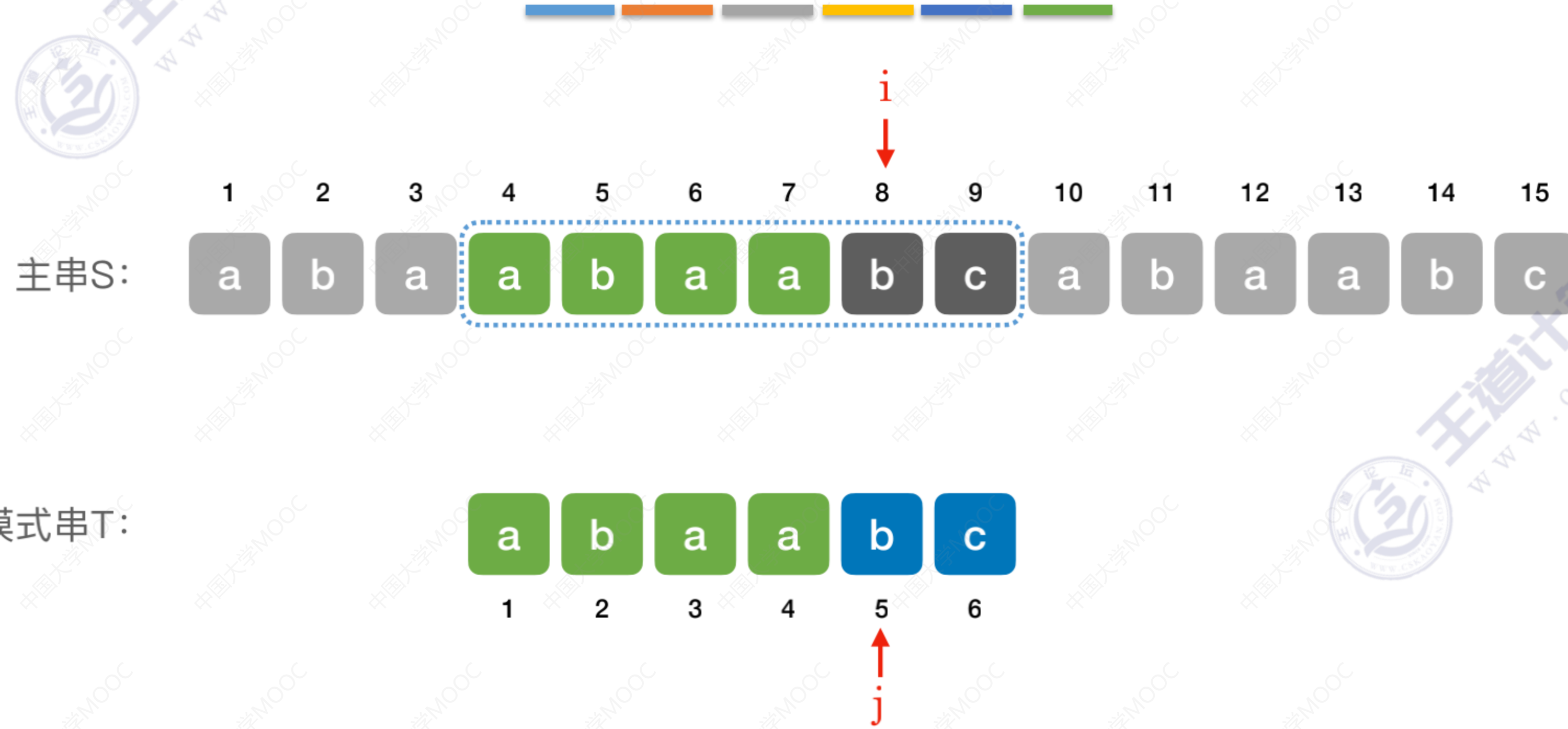
王道考研/CSKAOYAN.COM

朴素模式匹配算法



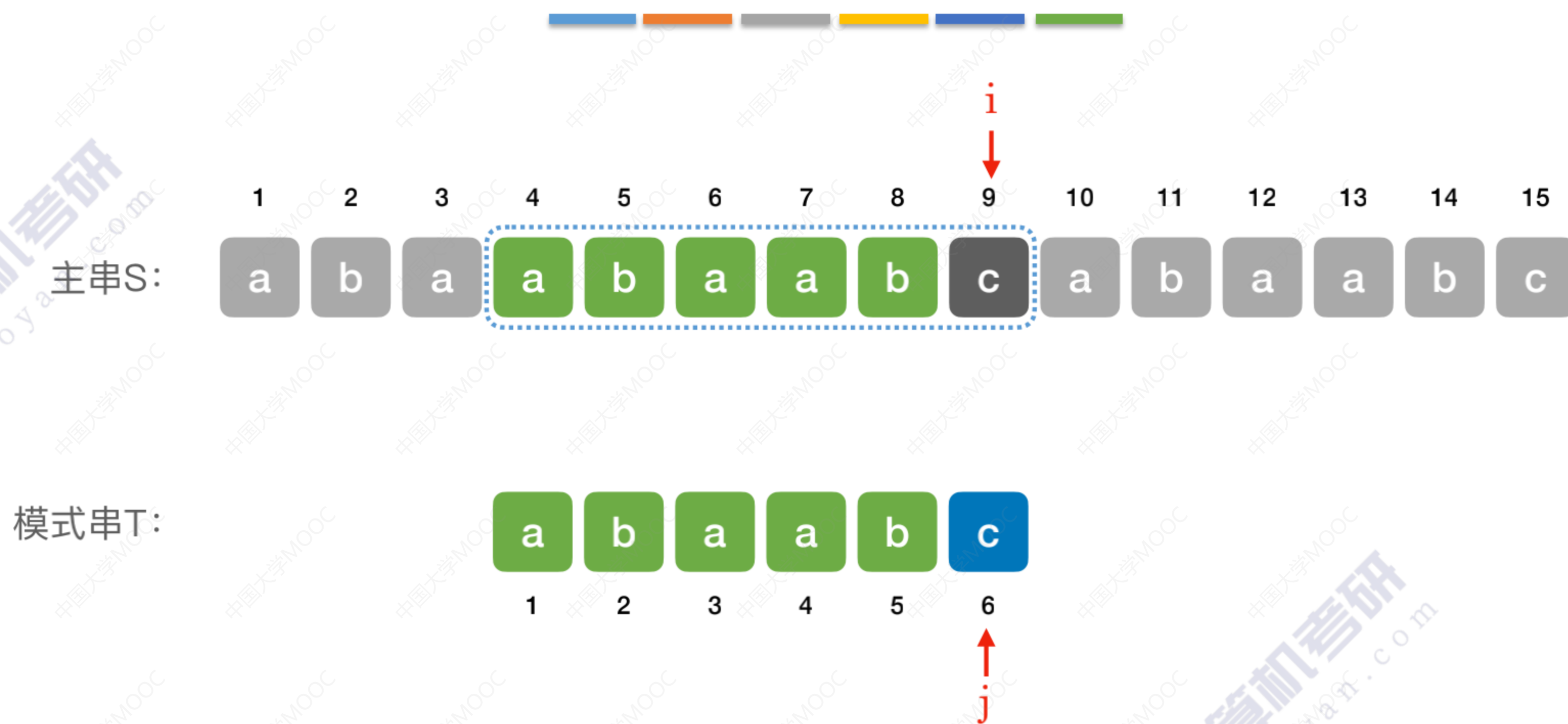
王道考研/CSKAOYAN.COM

朴素模式匹配算法



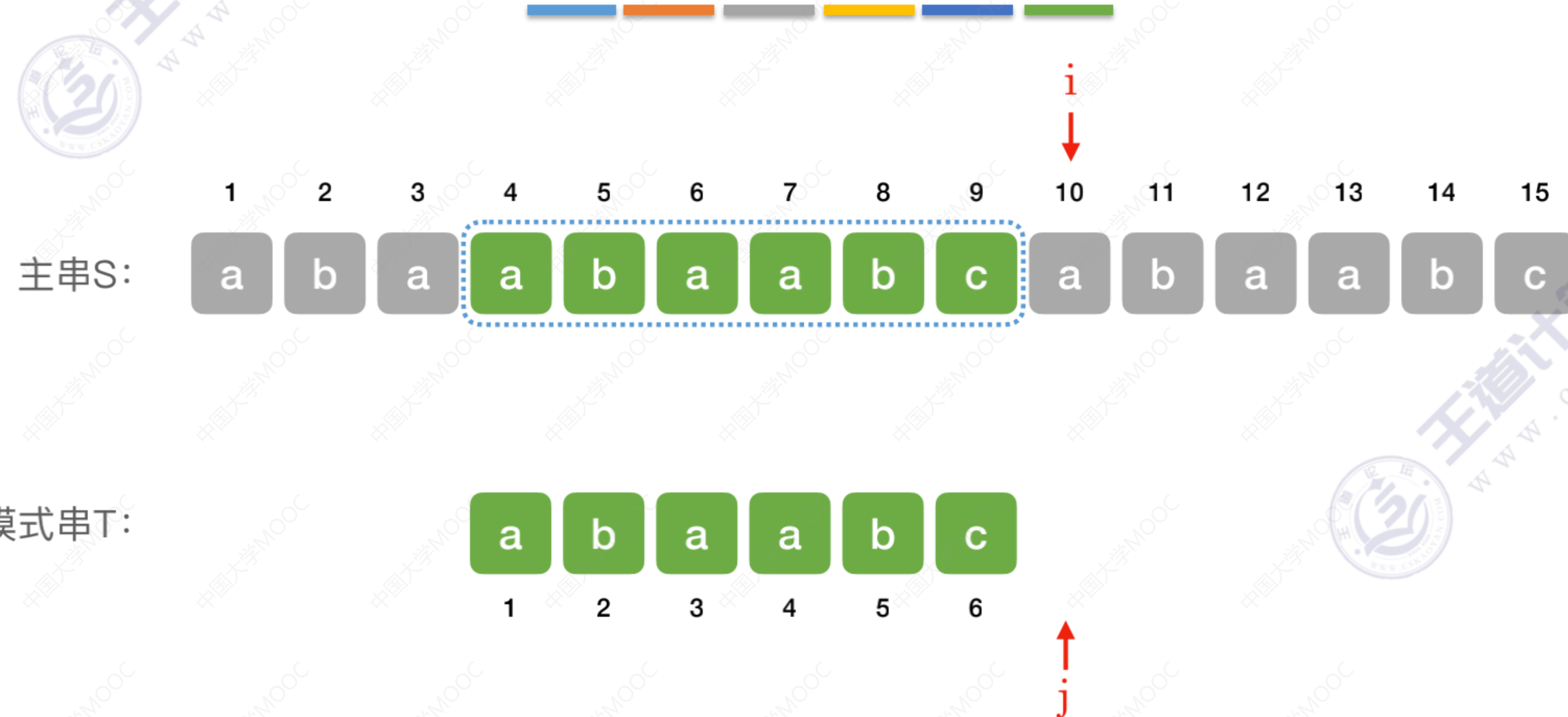
王道考研/CSKAOYAN.COM

朴素模式匹配算法



王道考研/CSKAOYAN.COM

朴素模式匹配算法



若 $j > T.length$, 则当前子串匹配成功, 返回当前子串第一个字符的位置 $-- i - T.length$

王道考研/CSKAOYAN.COM

朴素模式匹配算法

```
int Index(SString S, SString T){
    int i=1, j=1;
    while(i<=S.length && j<=T.length){
        if(S.ch[i]==T.ch[j]){
            ++i; ++j; //继续比较后继字符
        }
        else{
            i=i-j+2;
            j=1; //指针后退重新开始匹配
        }
    }
    if(j>T.length)
        return i-T.length;
    else
        return 0;
}
```

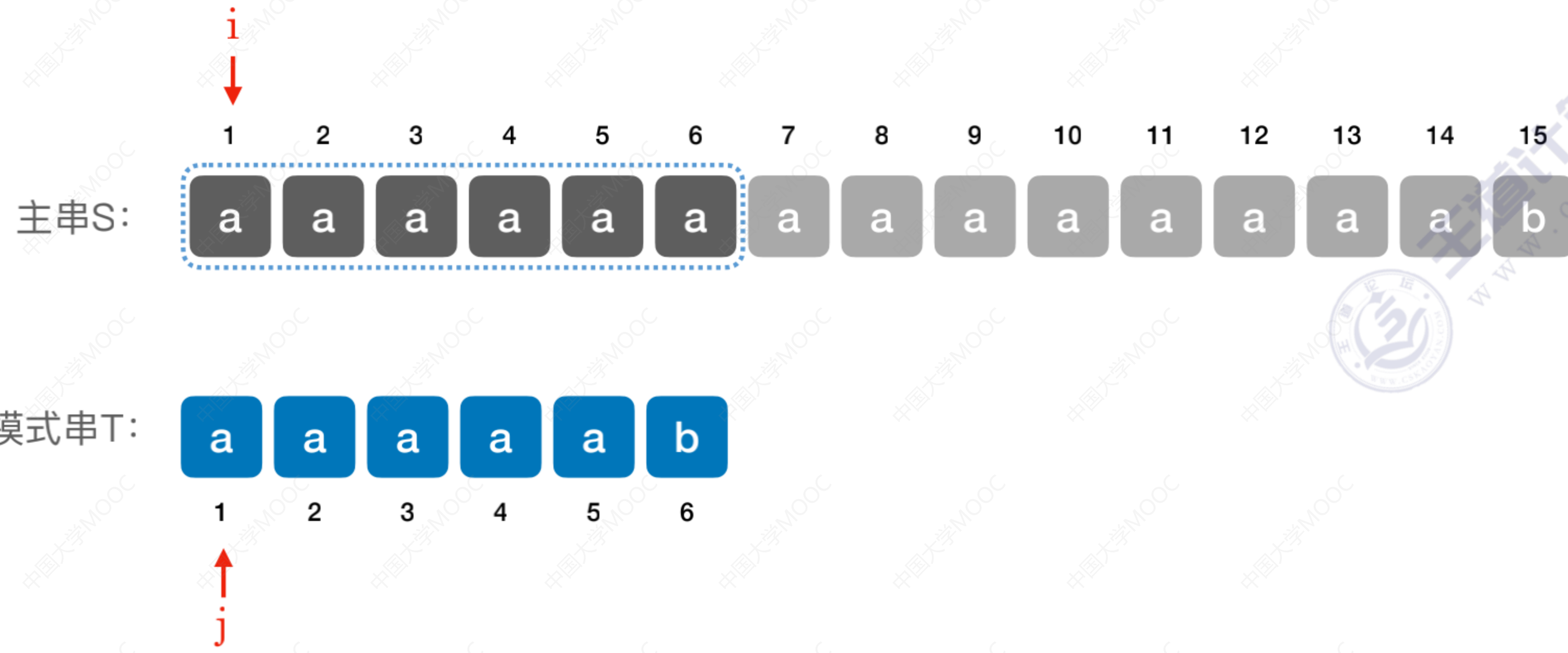
设主串长度为 n , 模式串长度为 m , 则

最坏时间复杂度 = $O(nm)$

王道考研/CSKAOYAN.COM

朴素模式匹配算法

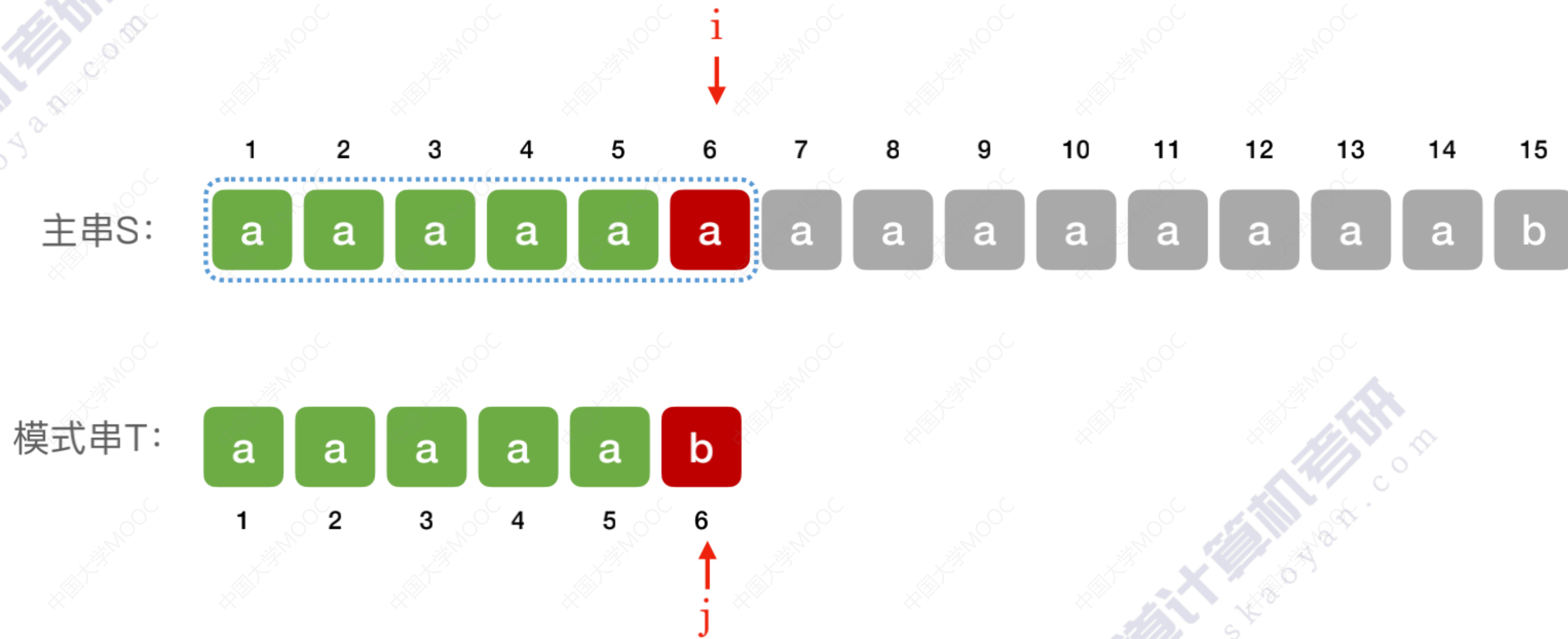
最坏时间复杂度 = $O(nm)$



王道考研/CSKAOYAN.COM

朴素模式匹配算法

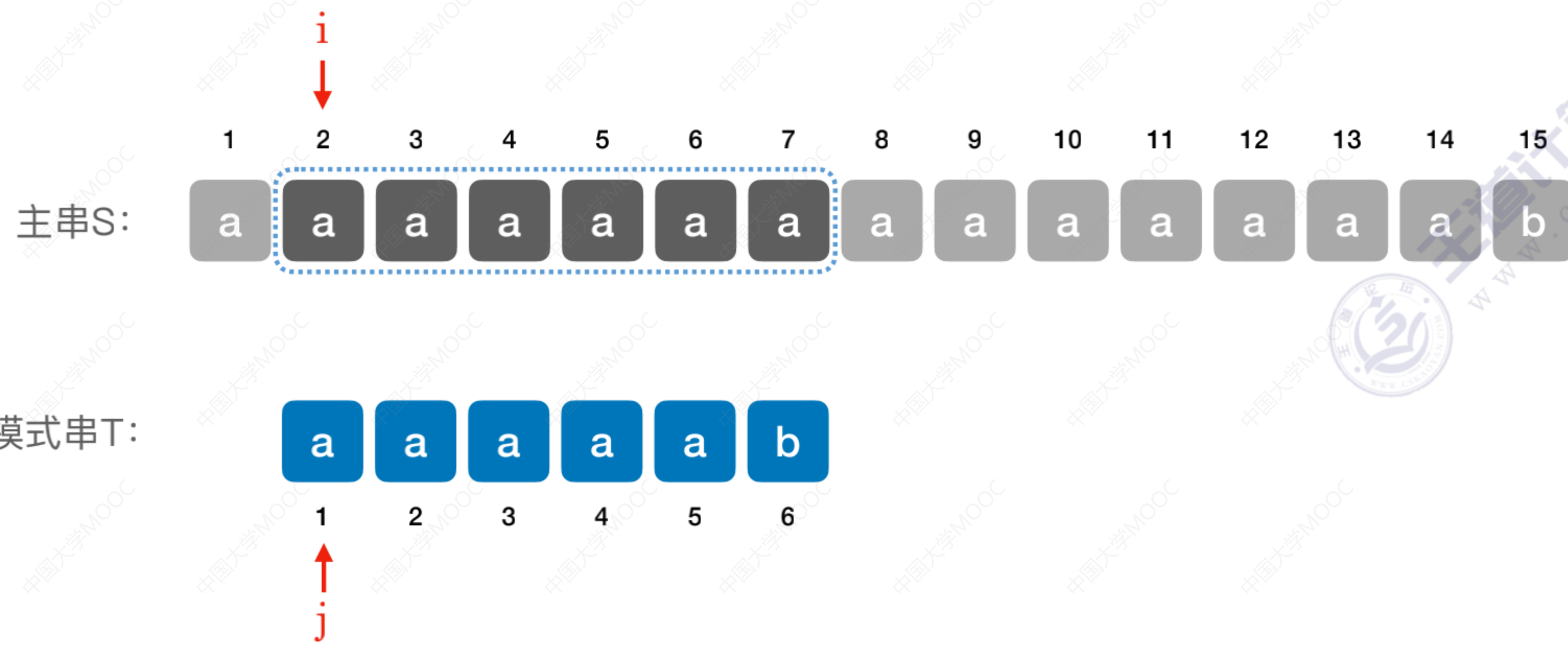
最坏时间复杂度 = $O(nm)$



王道考研/CSKAOYAN.COM

朴素模式匹配算法

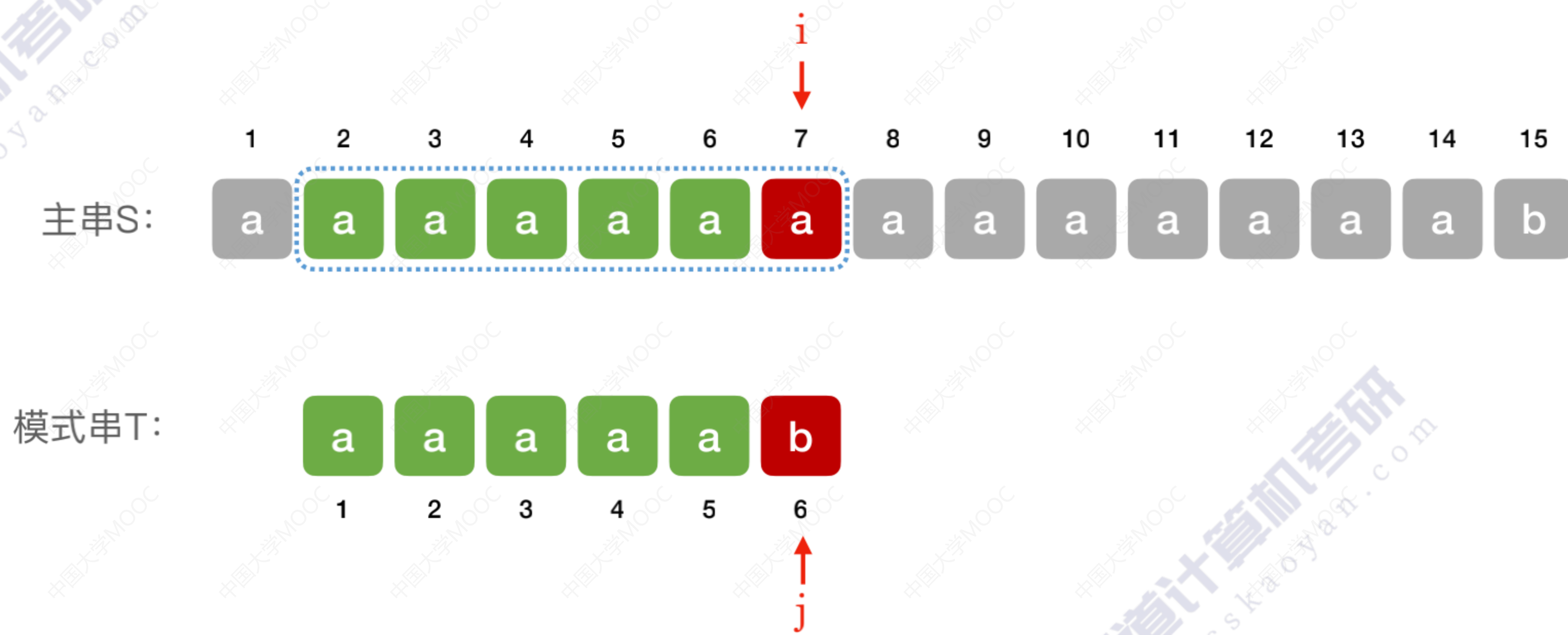
最坏时间复杂度 = $O(nm)$



王道考研/CSKAOYAN.COM

朴素模式匹配算法

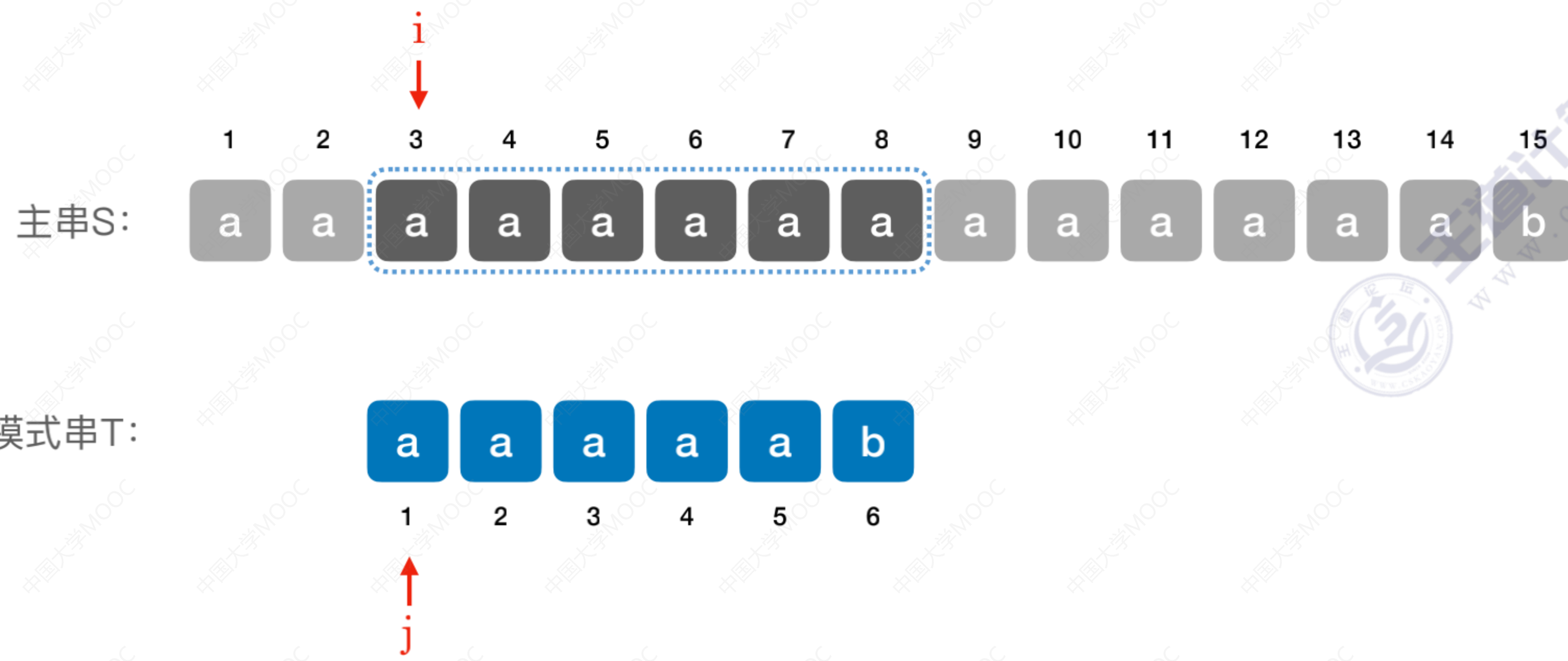
最坏时间复杂度 = $O(nm)$



王道考研/CSKAOYAN.COM

朴素模式匹配算法

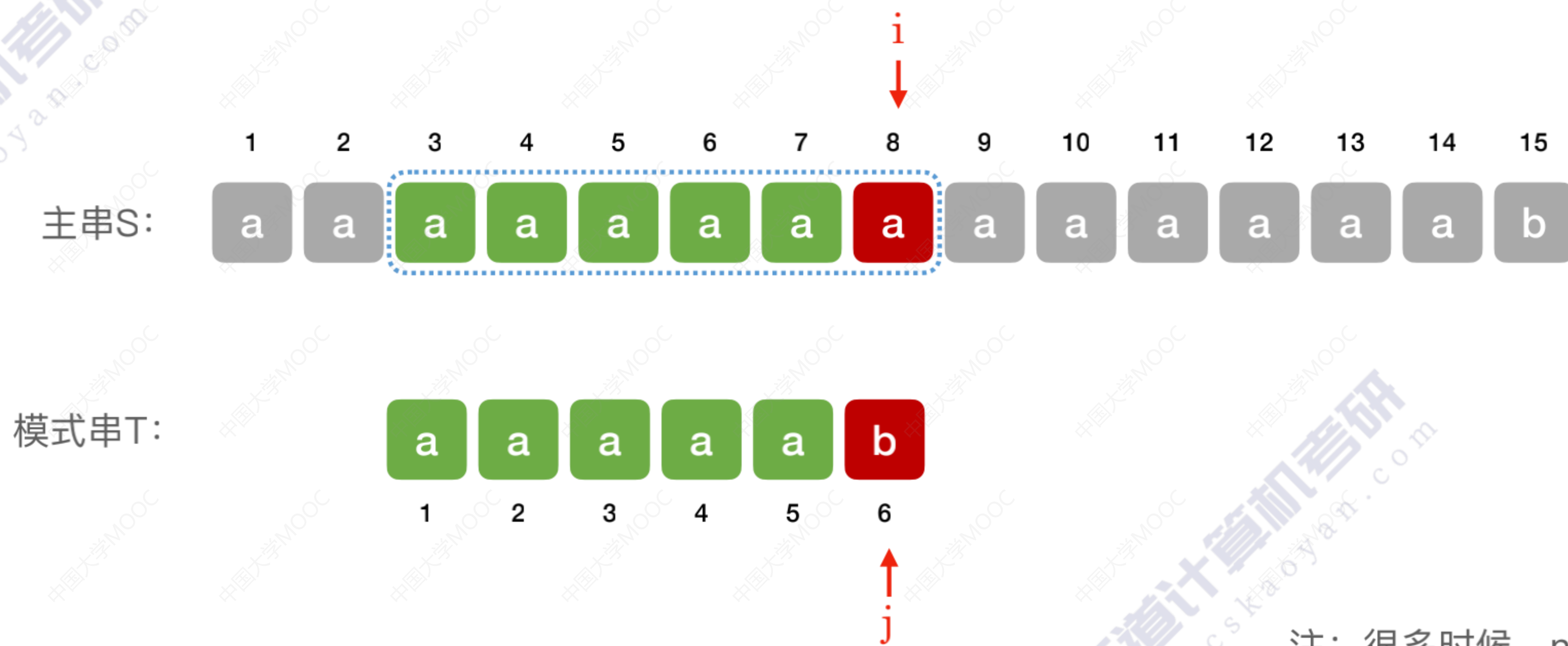
最坏时间复杂度 = $O(nm)$



王道考研/CSKAOYAN.COM

朴素模式匹配算法

最坏时间复杂度 = $O(nm)$



注: 很多时候, $n \gg m$

最坏的情况, 每个子串都要对比 m 个字符, 共 $n-m+1$ 个子串, 复杂度 = $O((n-m+1)m) = O(nm)$

王道考研/CSKAOYAN.COM

知识回顾与重要考点

朴素模式匹配算法

算法思想

主串长度 n ，模式串长度 m

将主串中所有长度为 m 的子串与模式串对比

找到第一个与模式串匹配的子串，并返回子串起始位置

若所有子串都不匹配，则返回0

最坏时间复杂度= $O(nm)$

王道考研/CSKAOYAN.COM



@王道论坛



@王道计算机考研备考



@王道咸鱼老师-计算机考研

@王道楼楼老师-计算机考研



@王道计算机考研



知乎

@王道计算机考研

微信视频号

@王道计算机考研

微信公众平台

@王道在线