字符串匹配

胡船长
初航我带你，远航靠自己

一、单模匹配问题

1. 易学易懂：Brute Force 算法
2. 高效方便：Sunday 算法
3. 经典回顾：Boyer Moore 算法
4. 变化多端：KMP 算法

二、多模匹配问题

1. 基于哈希：Rabin-Karp 算法
2. 初探 NFA：Shift-and/or 算法
3. 神兵利器：Trie 字典树
4. 飞升蜕变：AC 自动机

三、字符串匹配-课后实战题

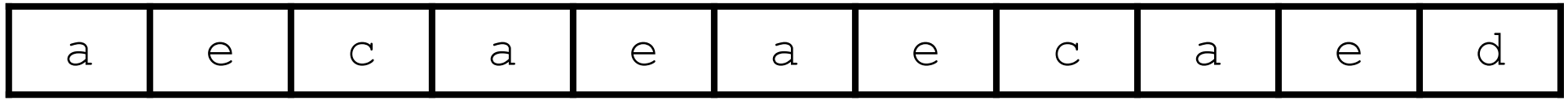
1. HZOJ-278：循环的字符串
2. HZOJ-279：项链的主人
3. HZOJ-281：前缀统计
4. HZOJ-282：最大异或对
5. HZOJ-283：拨号
6. P3370：【模板】字符串哈希
7. P5410：【模板】扩展 KMP
8. P1470：最长前缀
9. P8306：【模板】字典树
10. P2292：L 语言

一、单模匹配问题

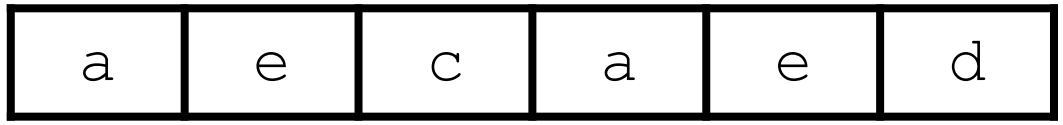
1. 易学易懂：Brute Force 算法
2. 高效方便：Sunday 算法
3. 经典回顾：Boyer Moore 算法
4. 变化多端：KMP 算法

暴力匹配

母串 S

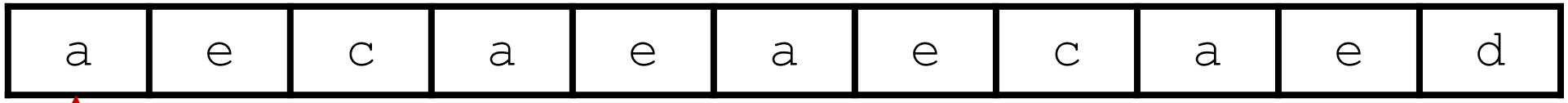


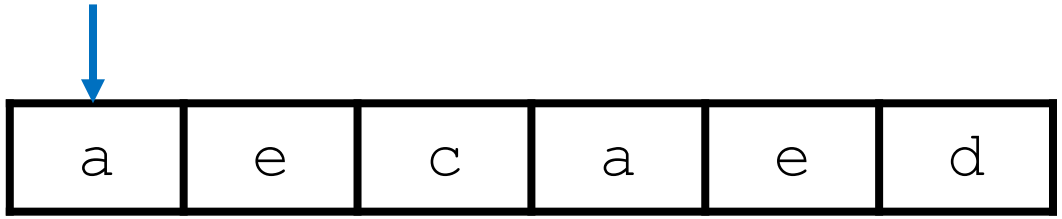
模式串 T



暴力匹配

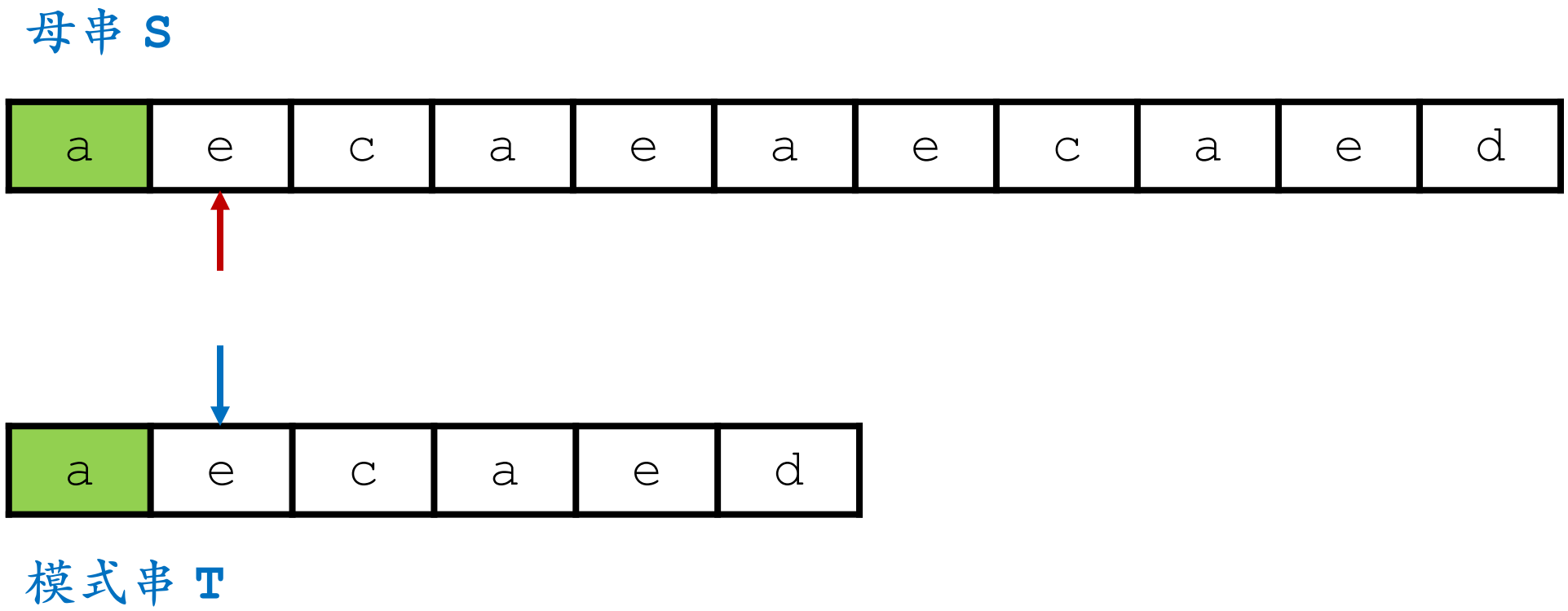
母串 S





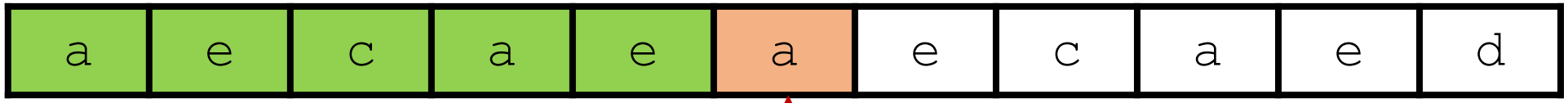
模式串 T

暴力匹配



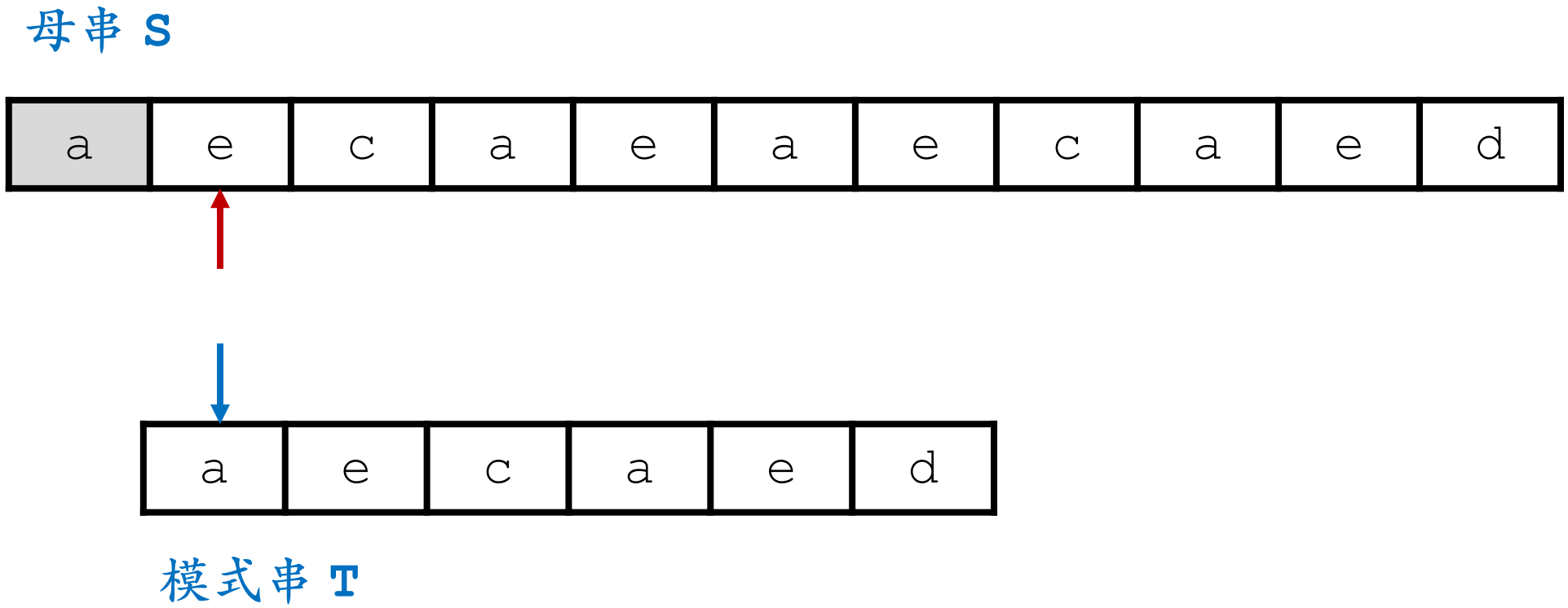
暴力匹配

母串 S

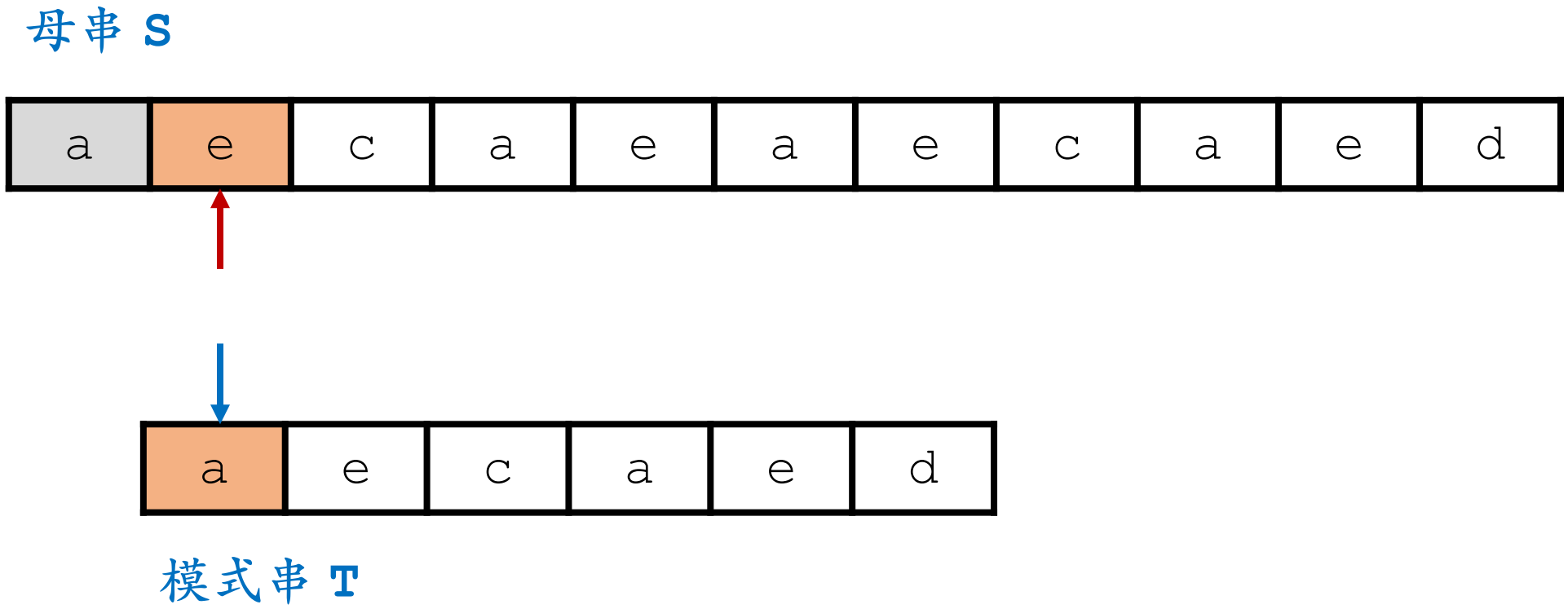


模式串 T

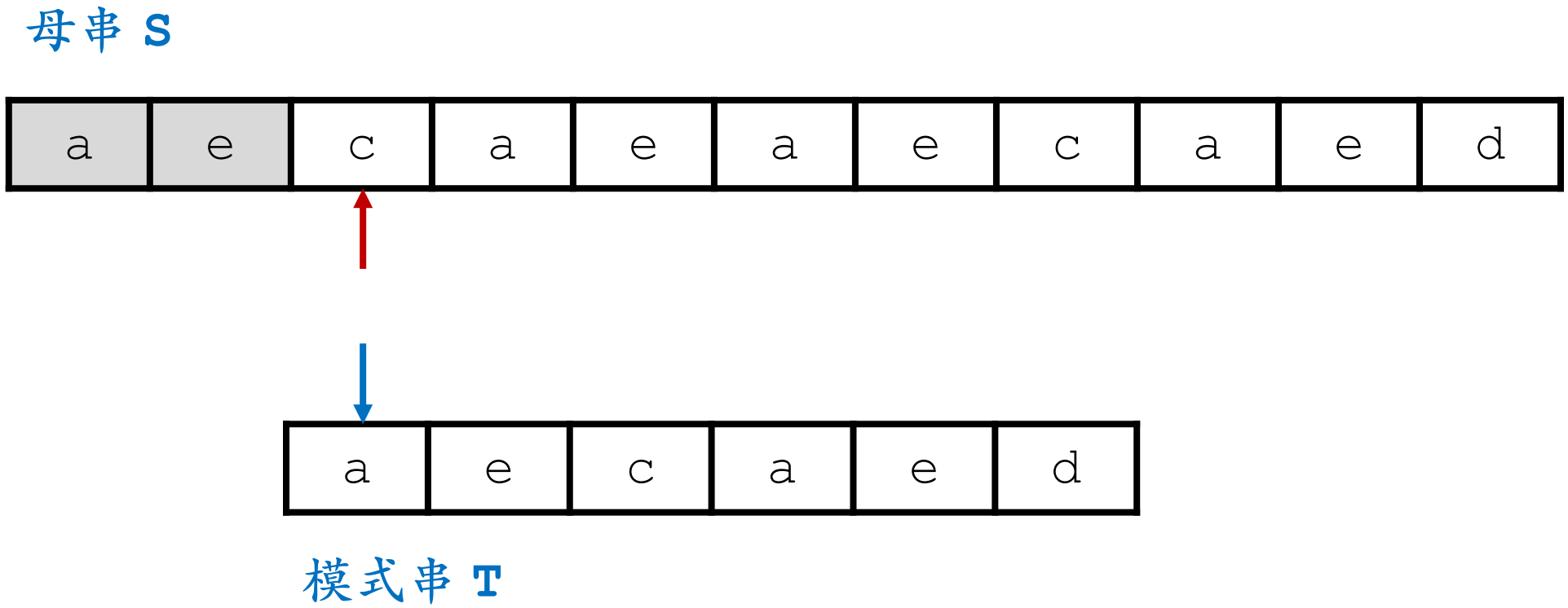
暴力匹配



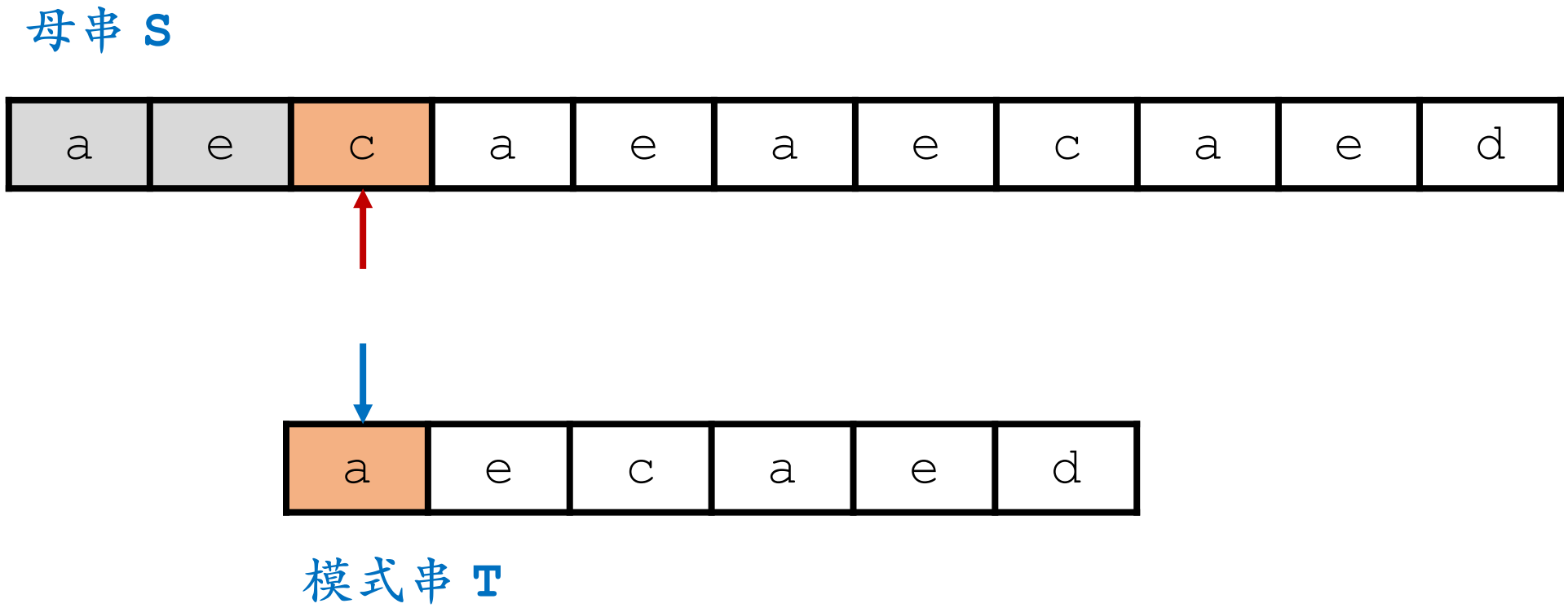
暴力匹配



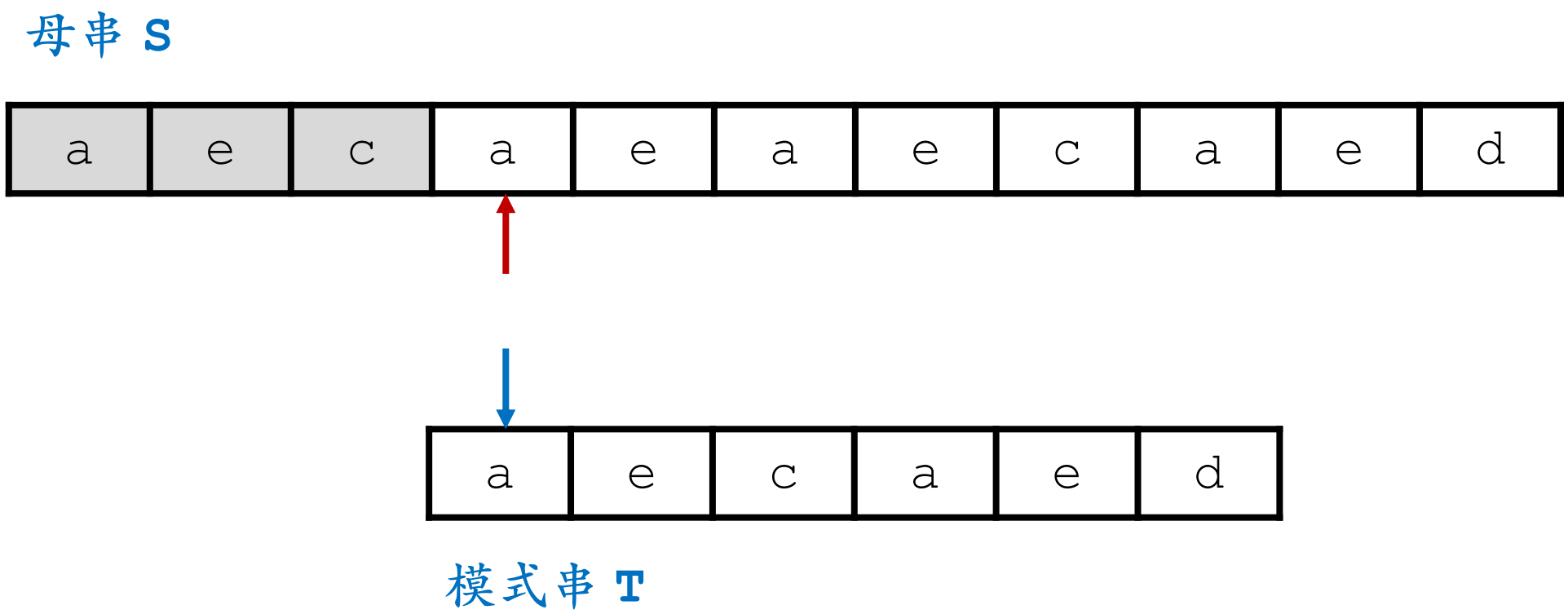
暴力匹配



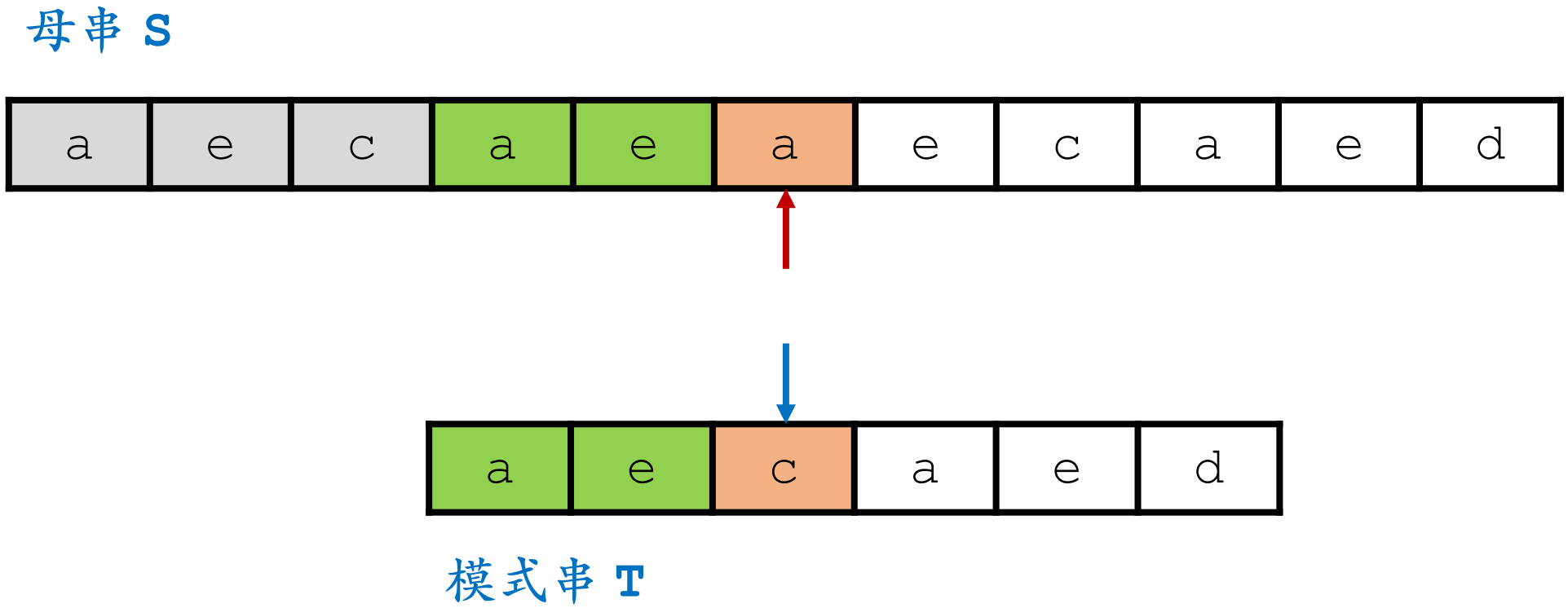
暴力匹配



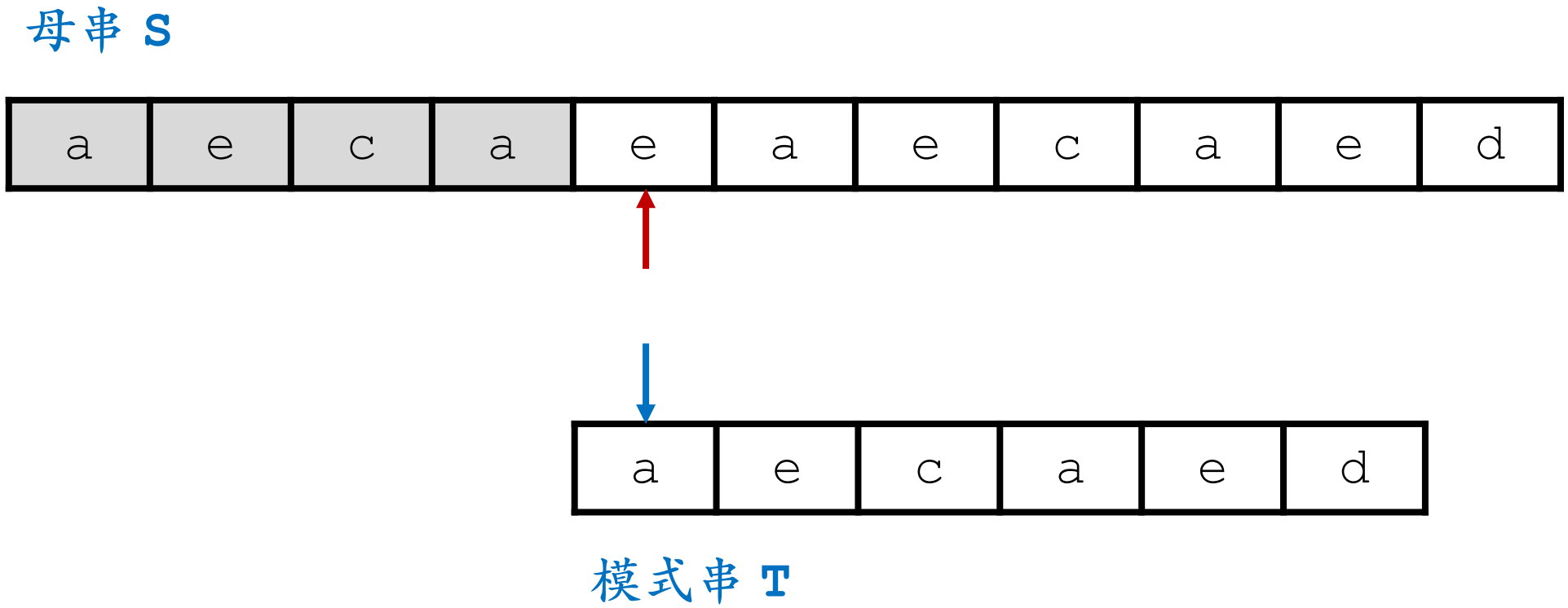
暴力匹配



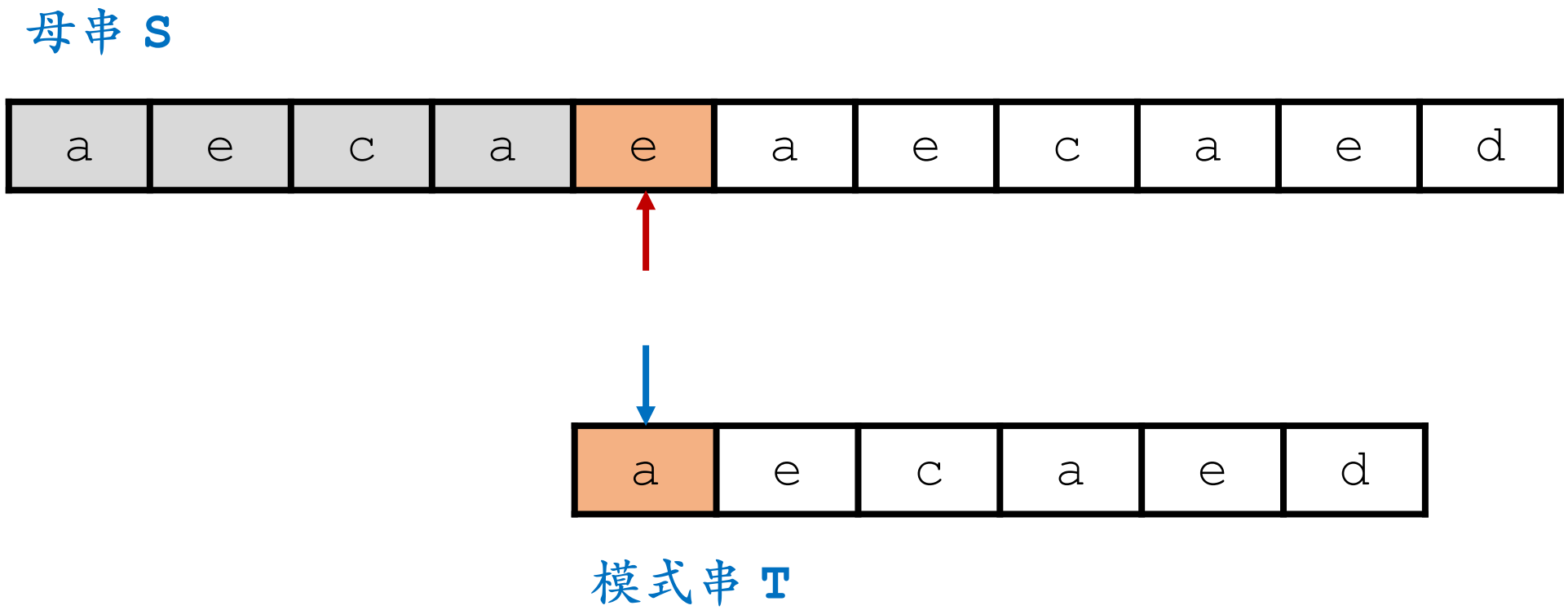
暴力匹配



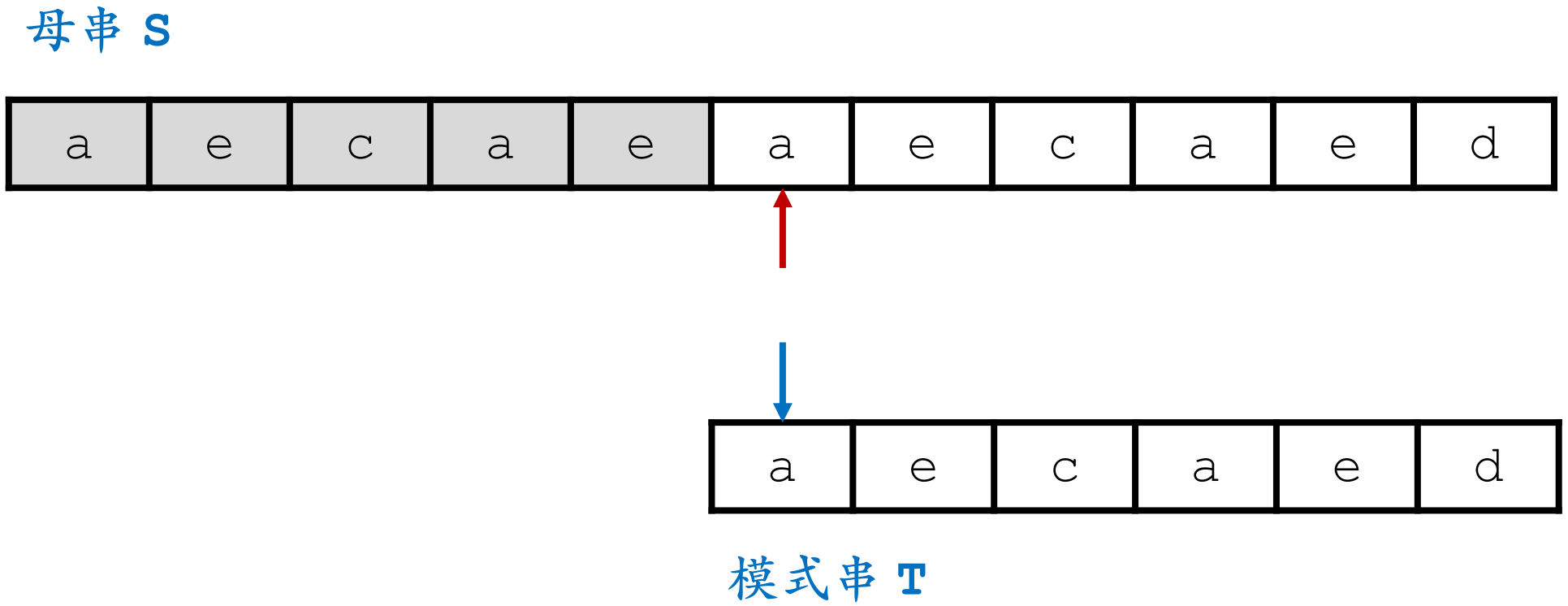
暴力匹配



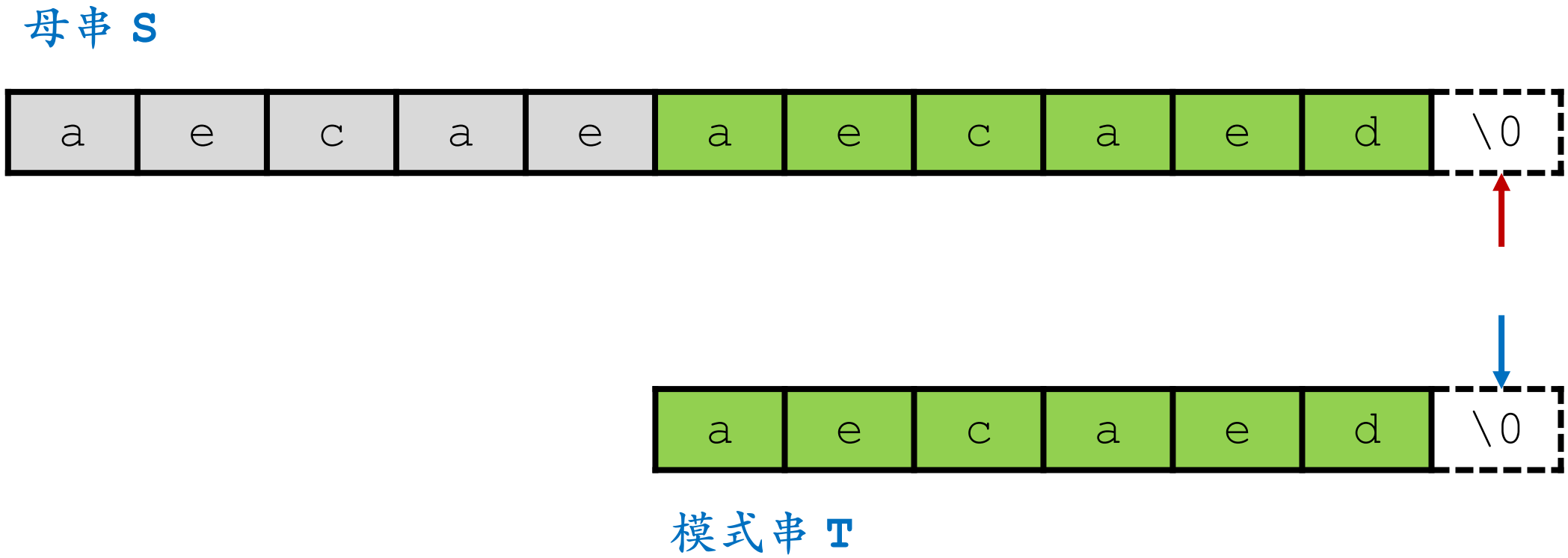
暴力匹配



暴力匹配



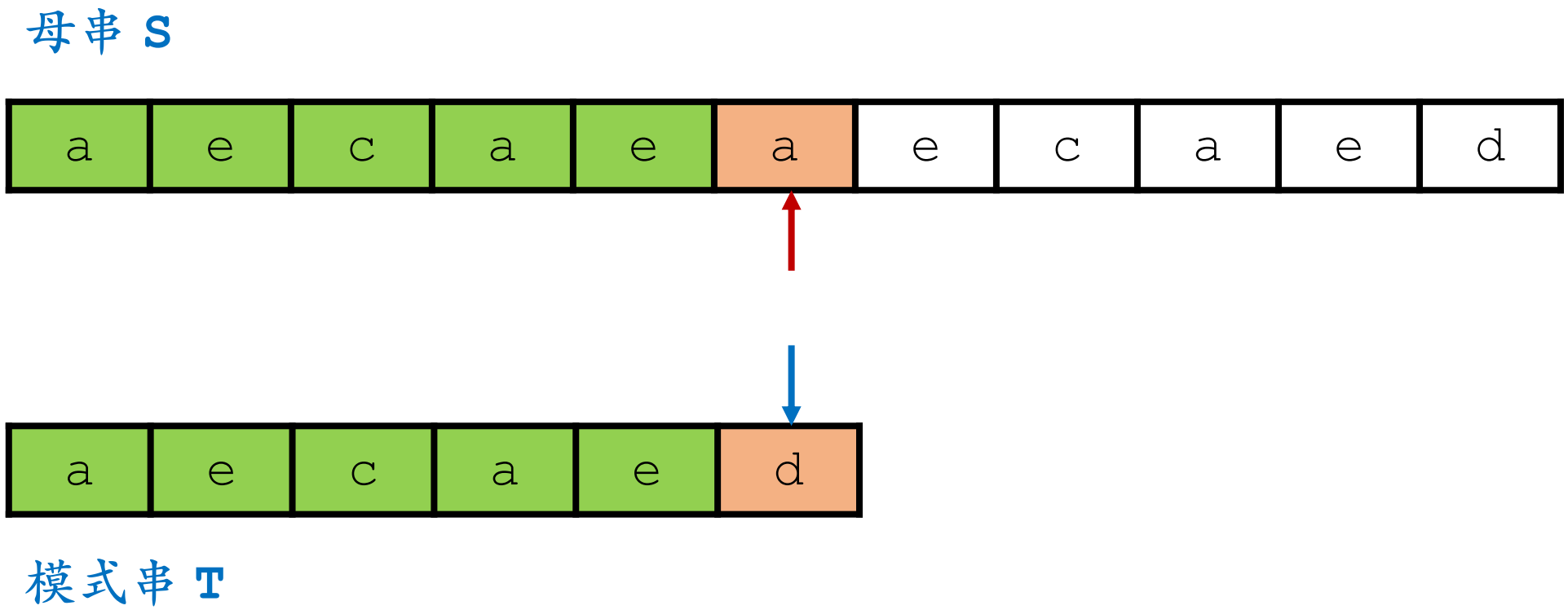
暴力匹配



一、单模匹配问题

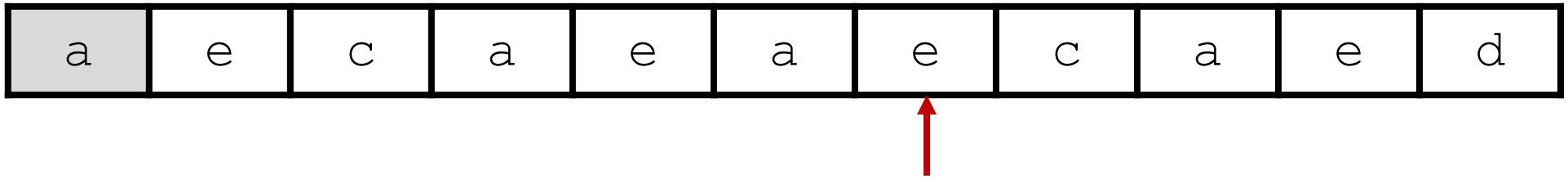
1. 易学易懂：Brute Force 算法
2. 高效方便：Sunday 算法
3. 经典回顾：Boyer Moore 算法
4. 变化多端：KMP 算法

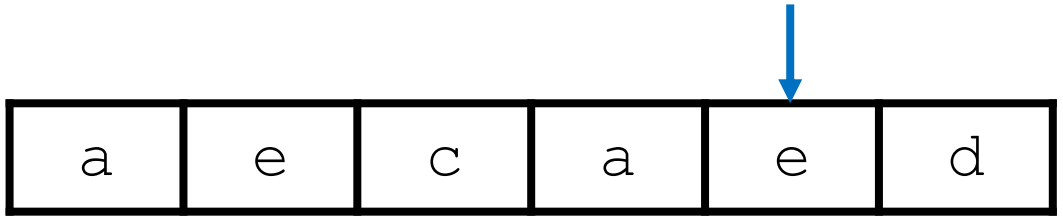
Sunday 算法



Sunday 算法

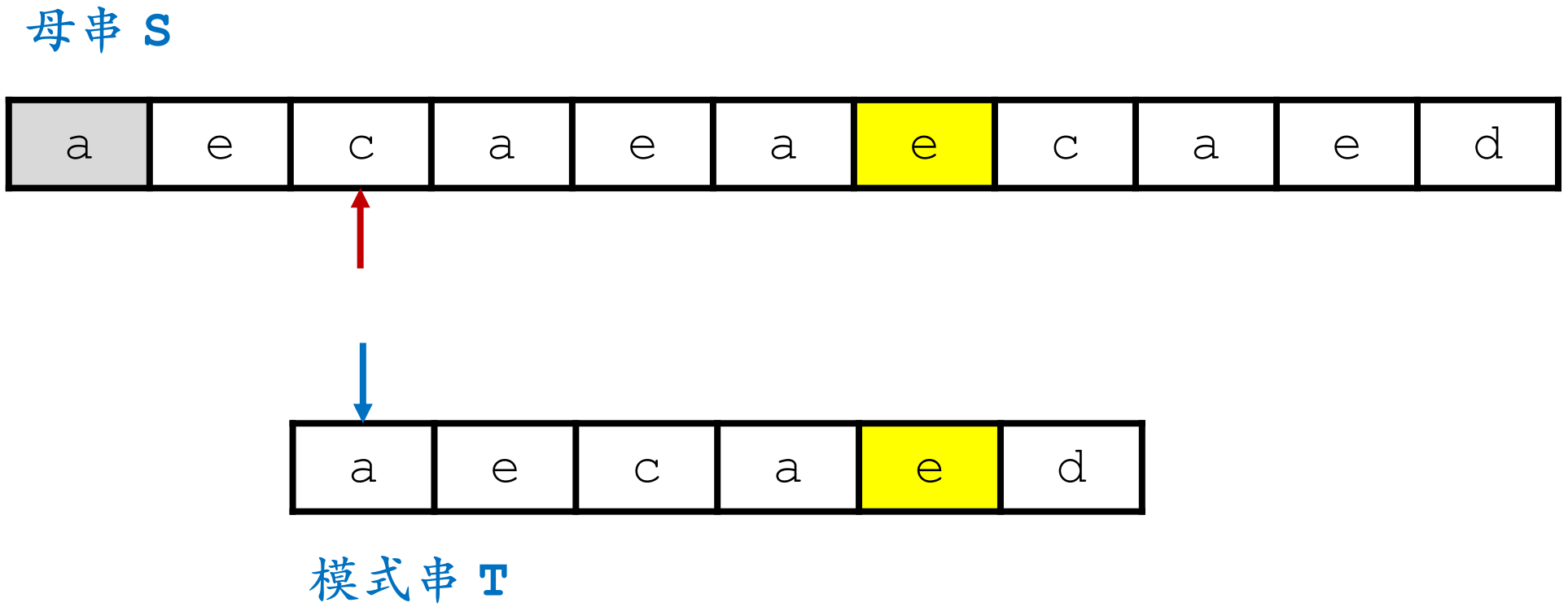
母串 S





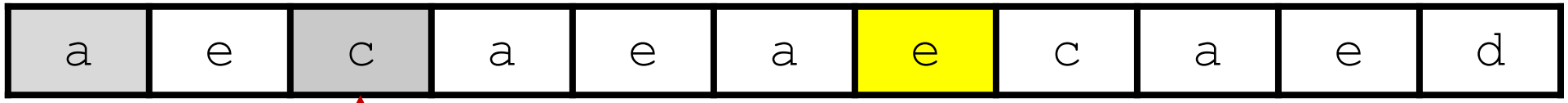
模式串 T

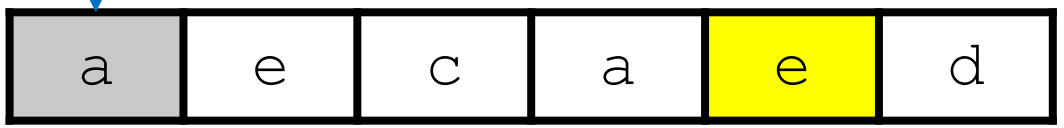
Sunday 算法



Sunday 算法

母串 S

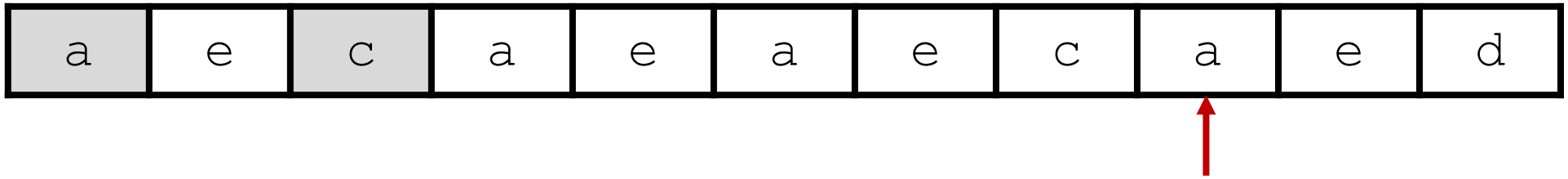


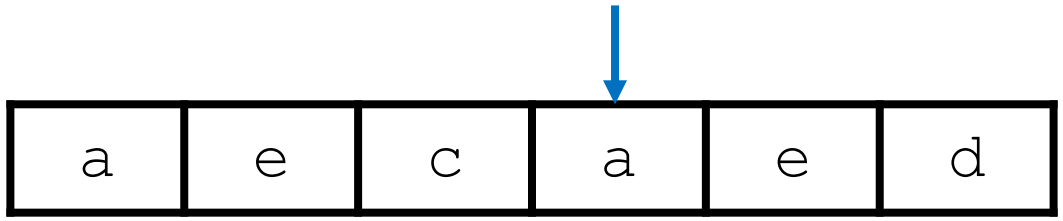


模式串 T

Sunday 算法

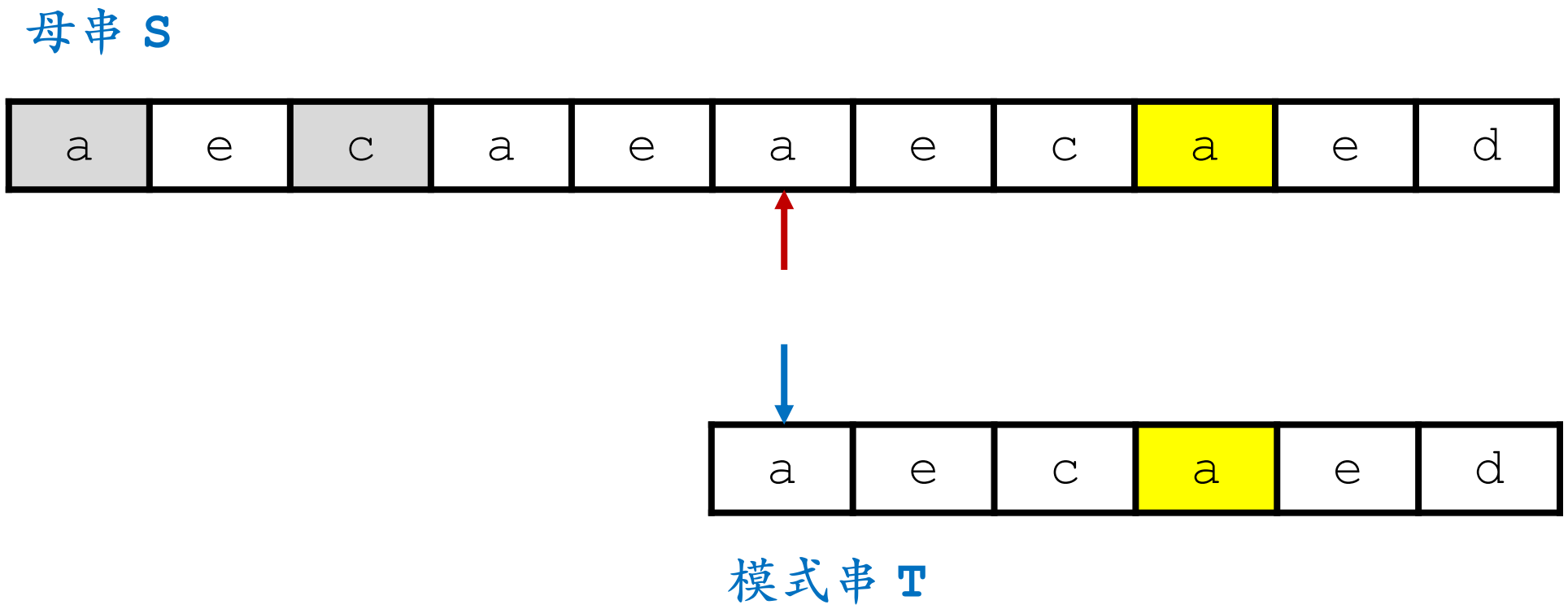
母串 S



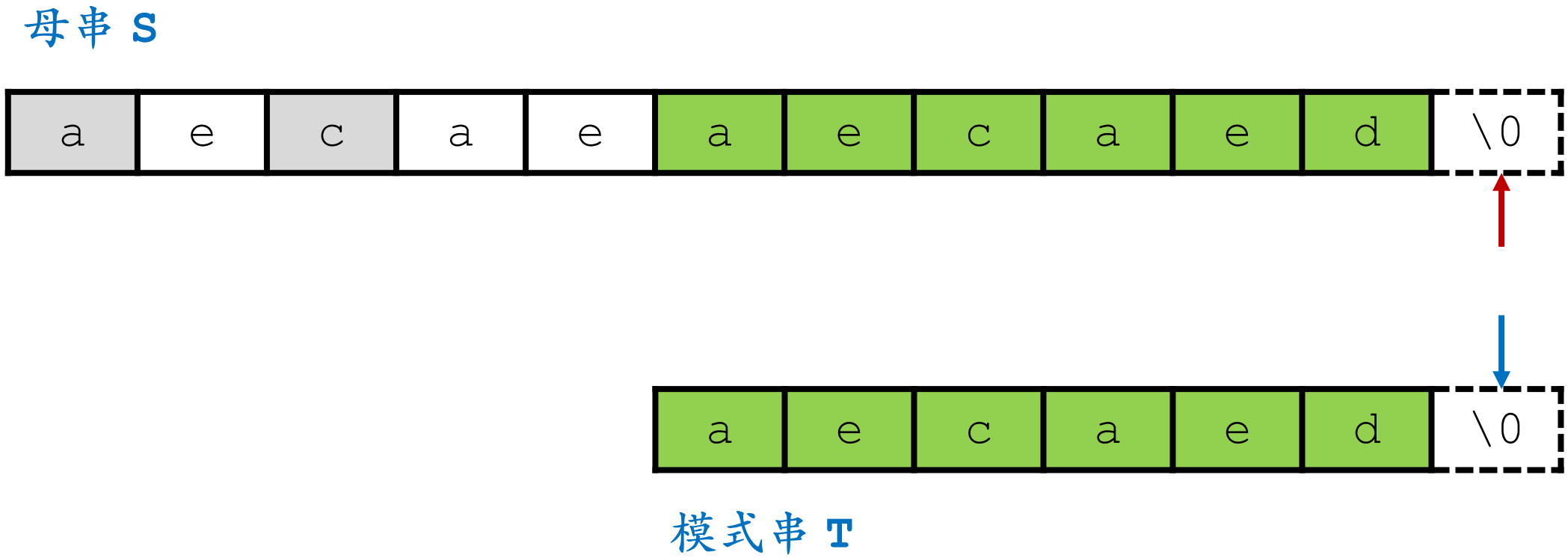


模式串 T

Sunday 算法



Sunday 算法



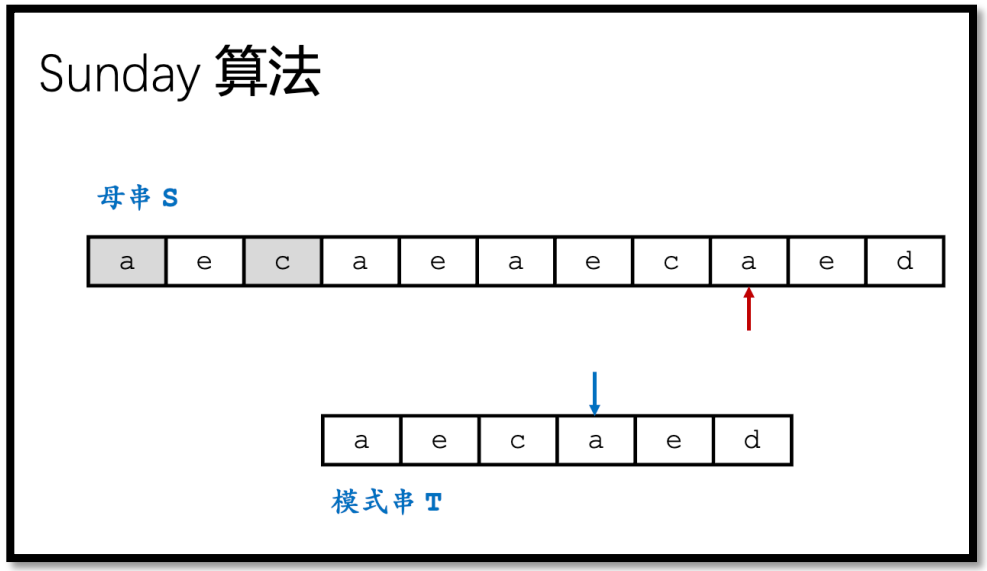
一、单模匹配问题

1. 易学易懂：Brute Force 算法
2. 高效方便：Sunday 算法
3. 经典回顾：Boyer Moore 算法
4. 变化多端：KMP 算法

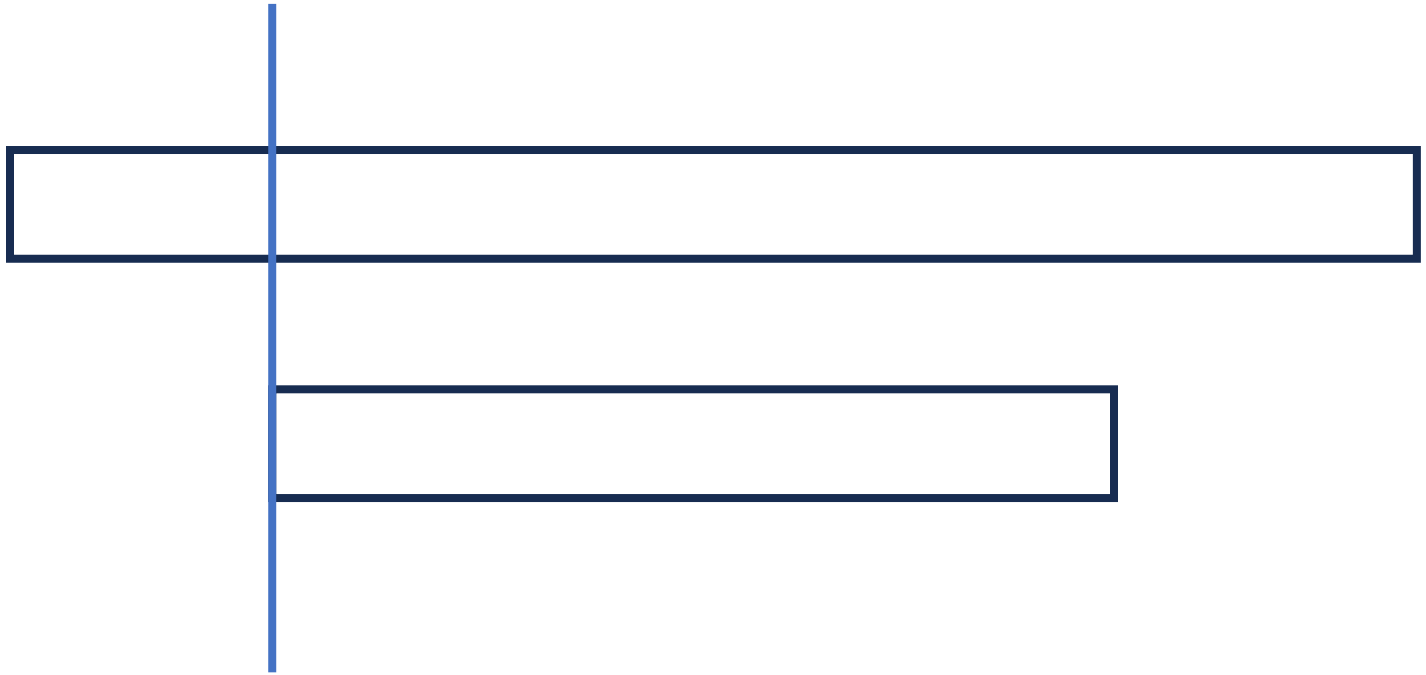
Boyer Moore 算法

理解 BM 算法的核心法门：

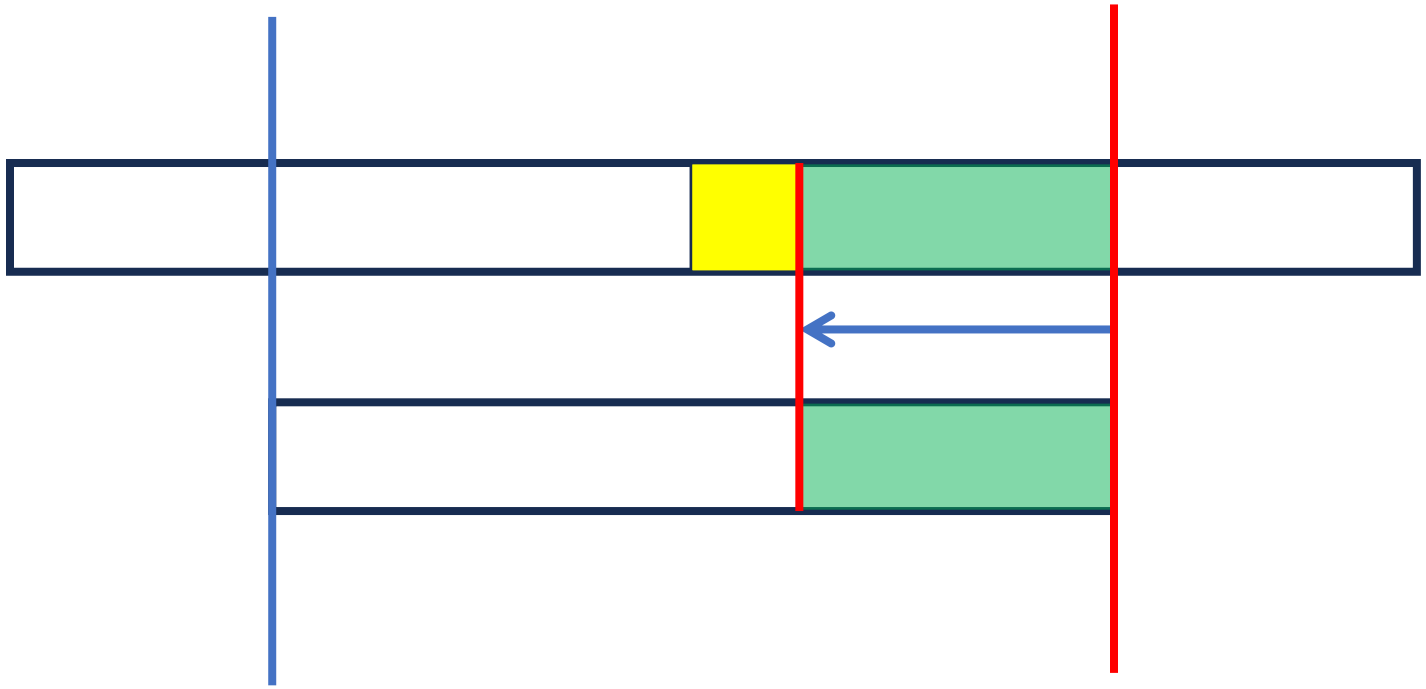
1. 失配时，模式串尽可能向后移动最大长度
2. 移动的长度取决于2条规则中的较大值
3. 规则1：坏字符规则 delta1
4. 规则2：好后缀规则 delta2



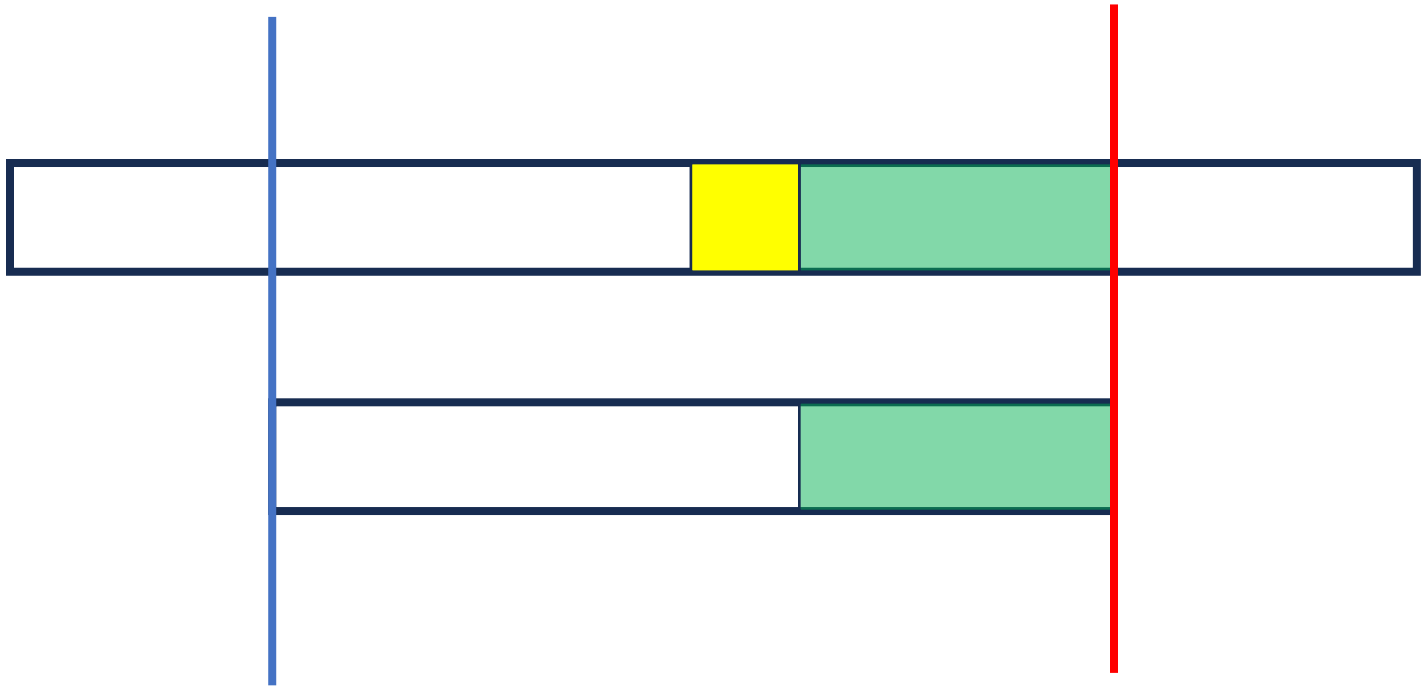
Boyer Moore 算法-匹配方向



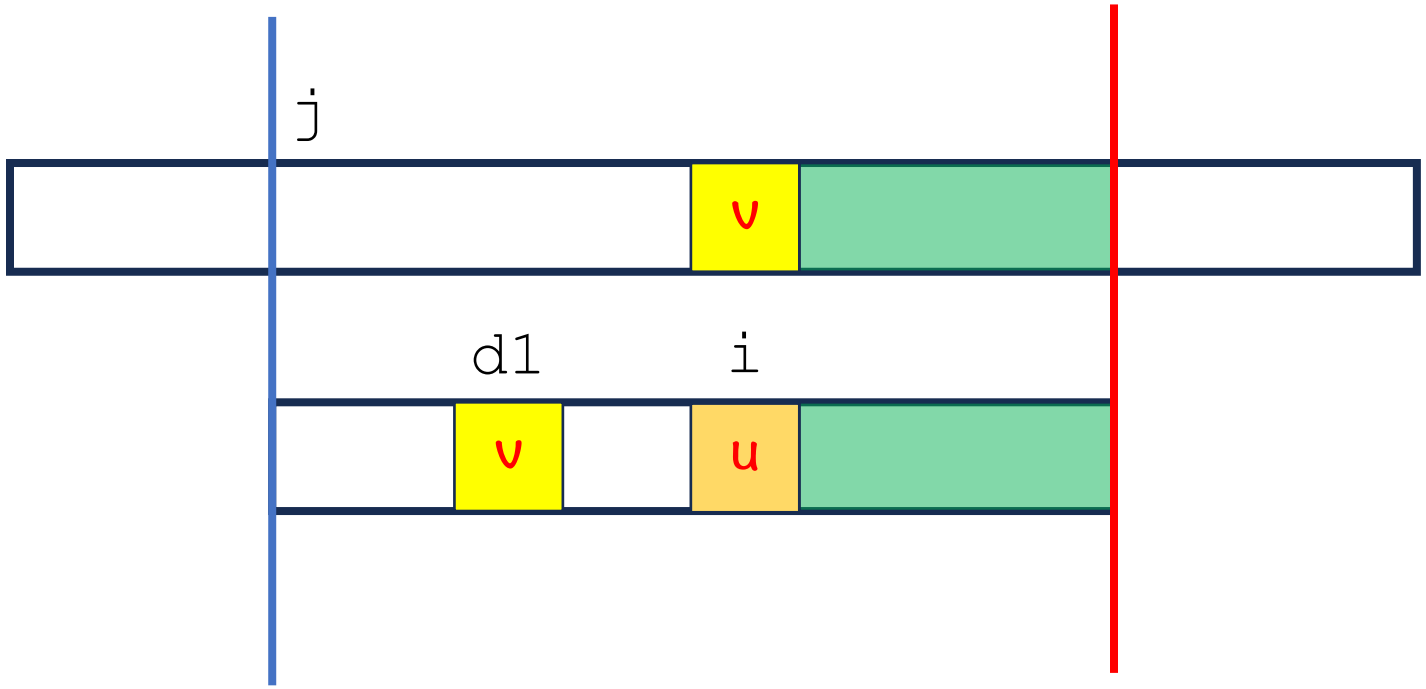
Boyer Moore 算法-匹配方向



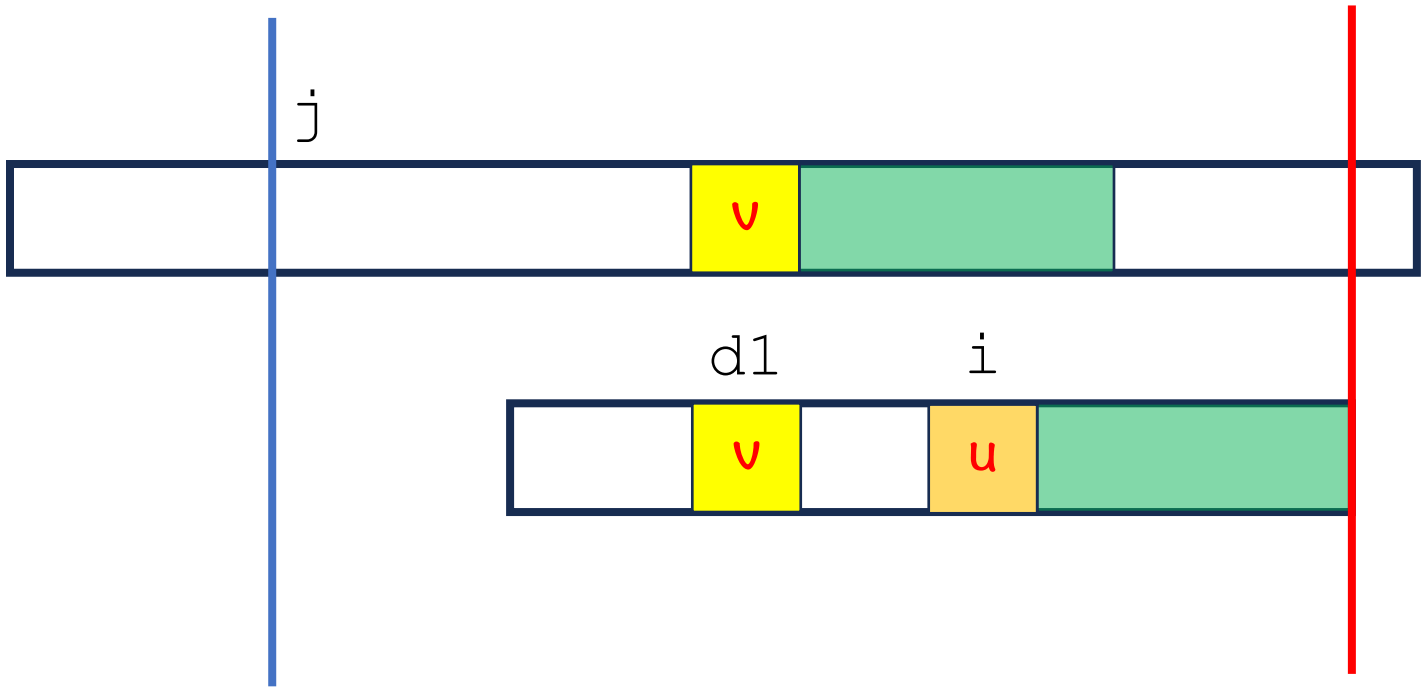
Boyer Moore 算法-坏字符规则



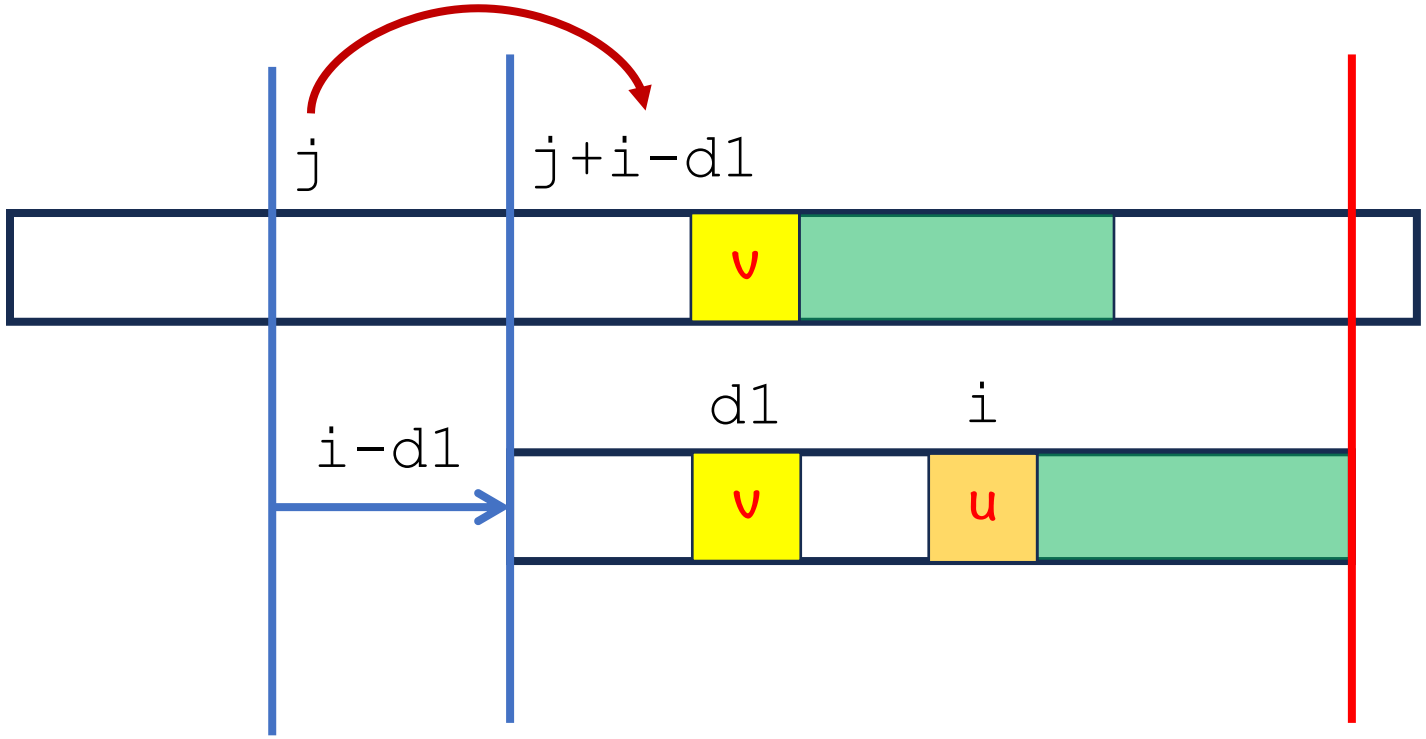
Boyer Moore 算法-坏字符规则



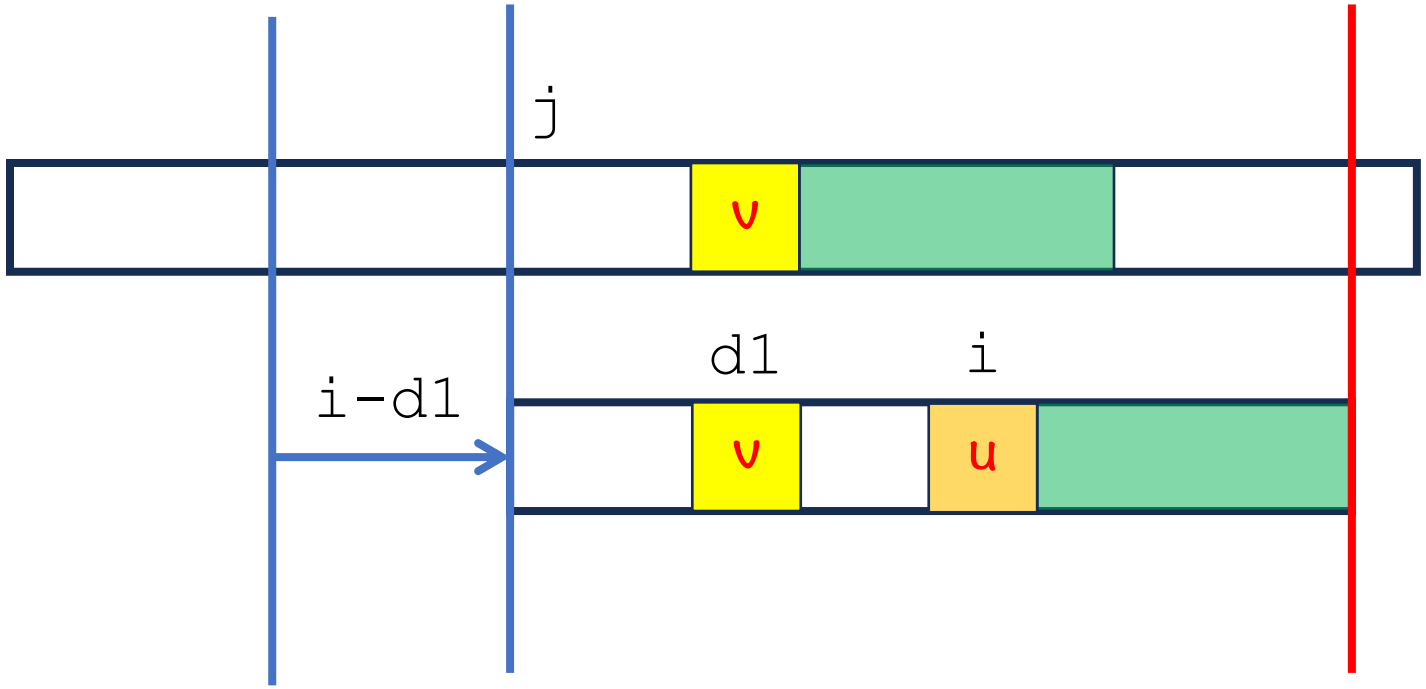
Boyer Moore 算法-坏字符规则



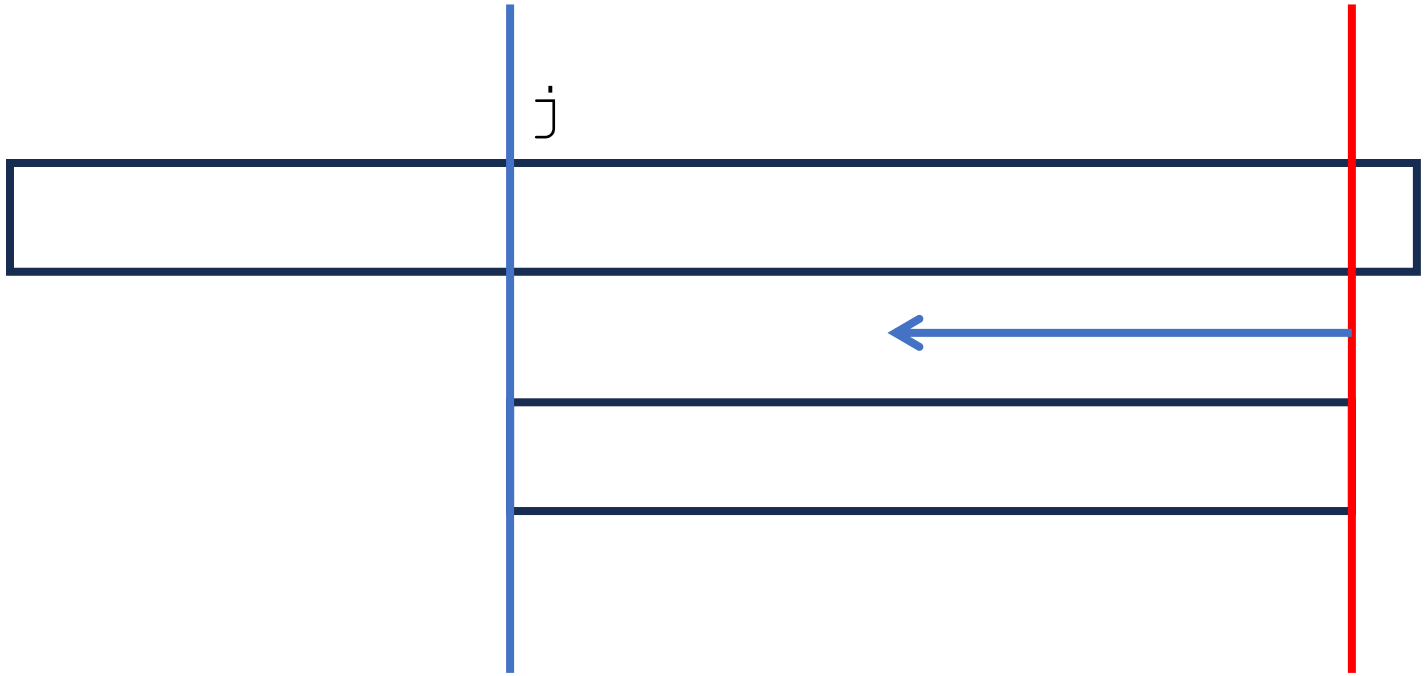
Boyer Moore 算法-坏字符规则



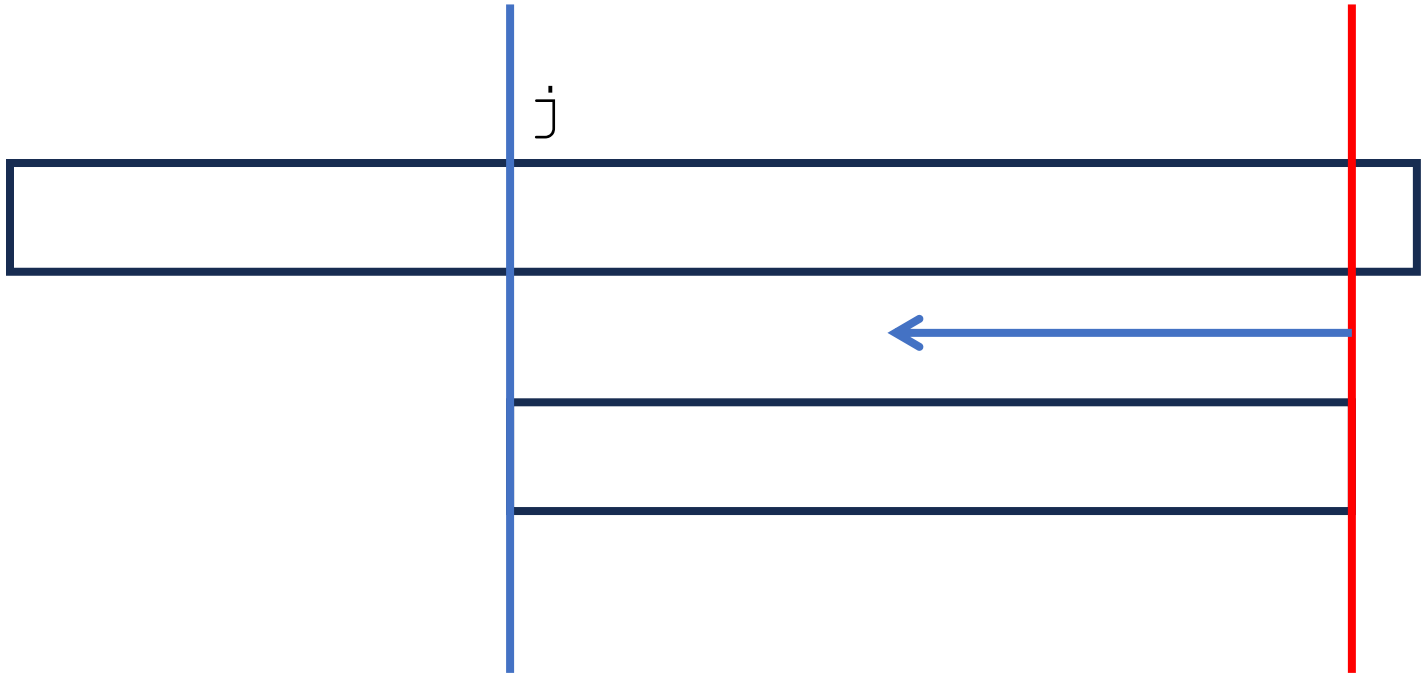
Boyer Moore 算法-坏字符规则



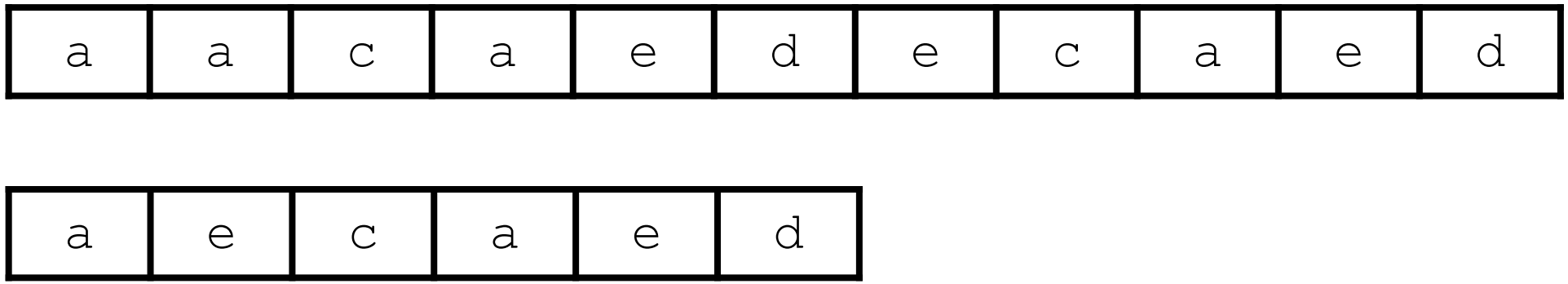
Boyer Moore 算法-坏字符规则



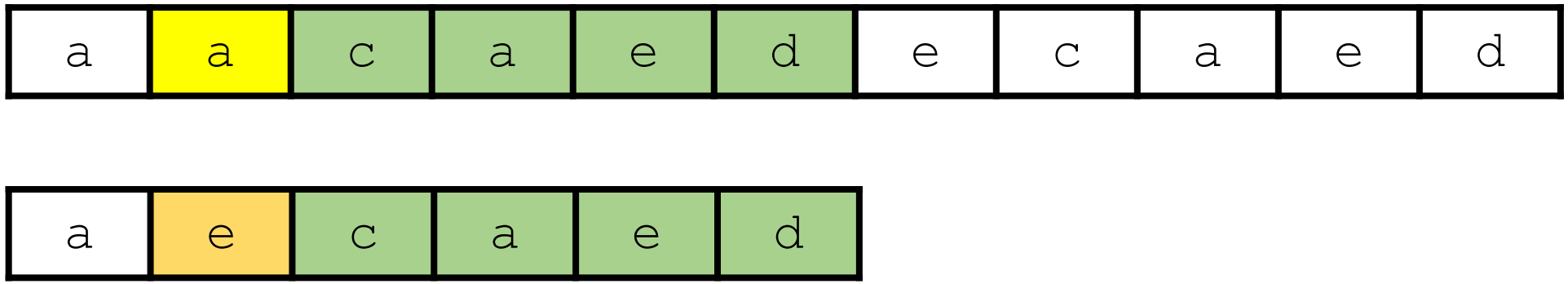
坏字符规则-回退BUG



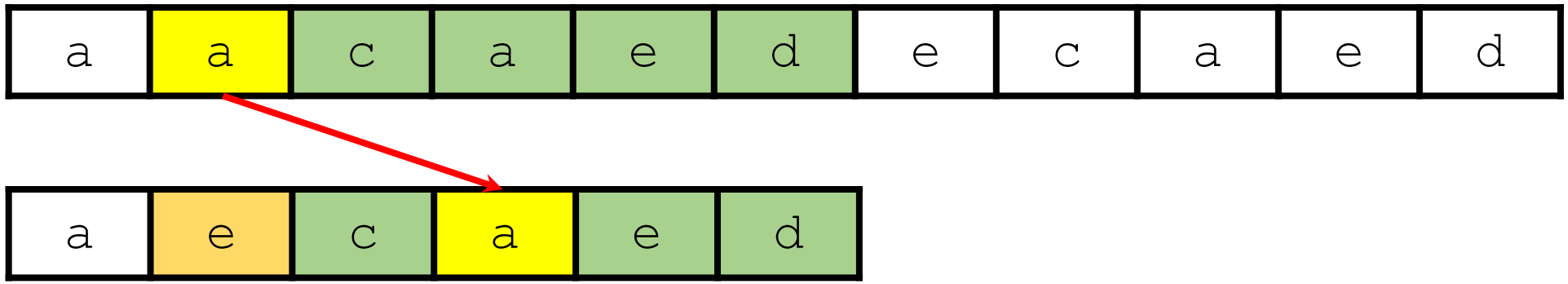
坏字符规则-回退BUG



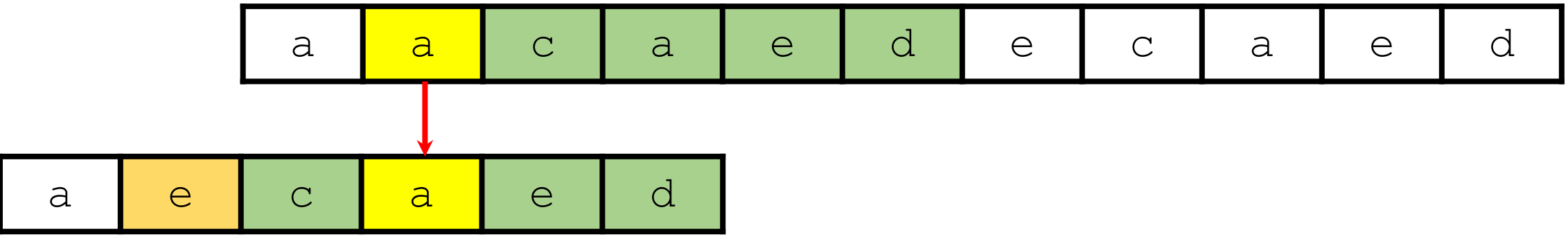
坏字符规则-回退BUG



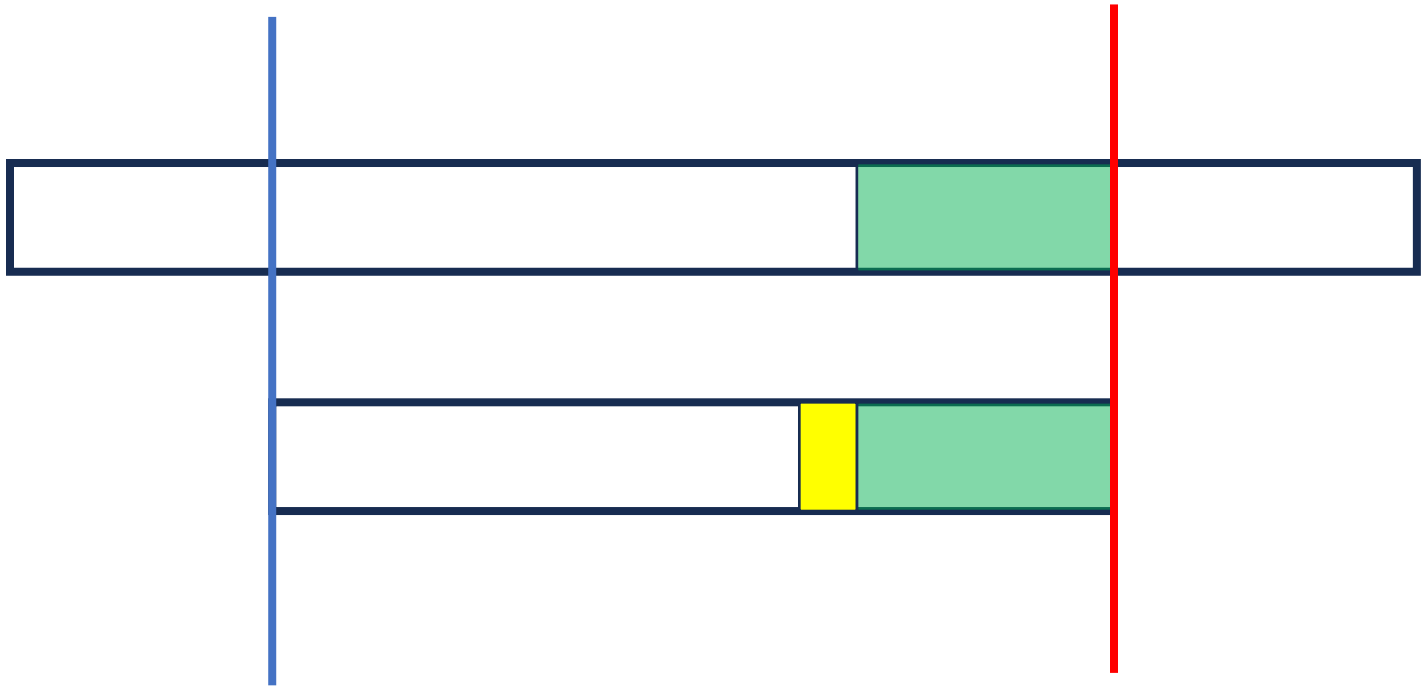
坏字符规则-回退BUG



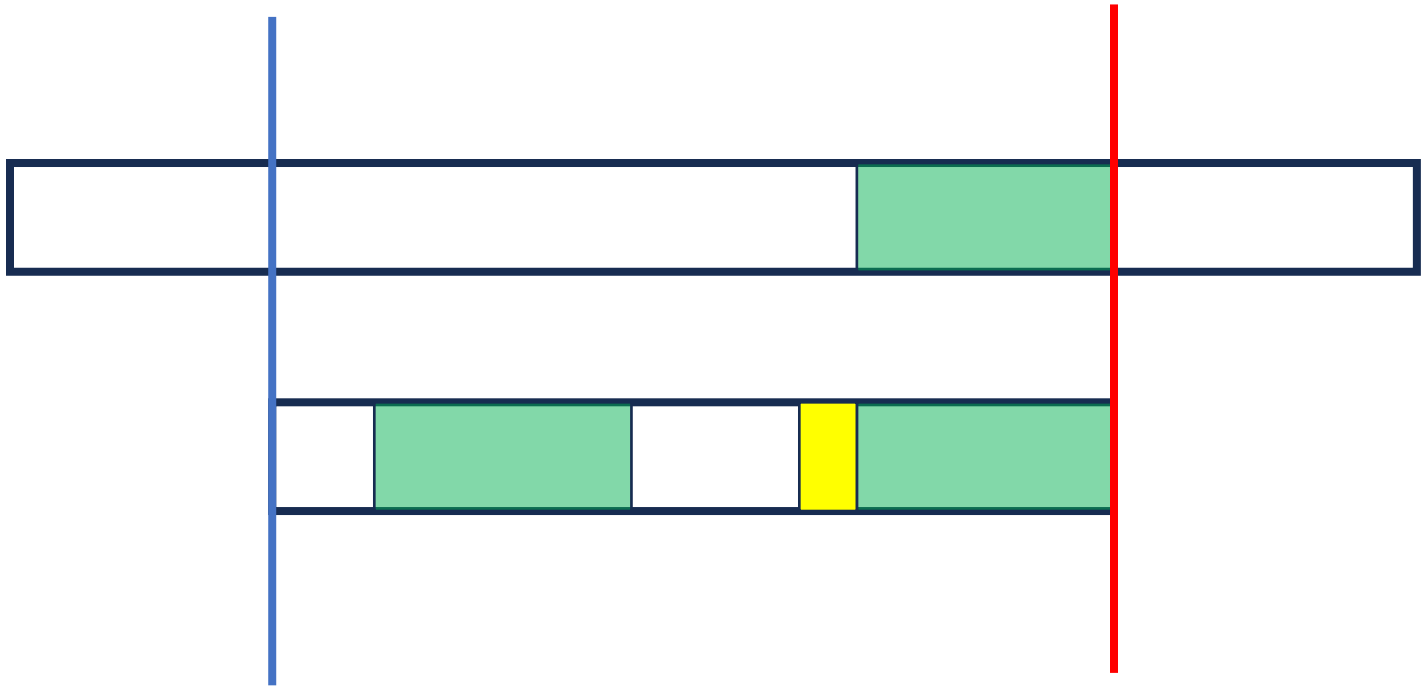
坏字符规则-回退BUG



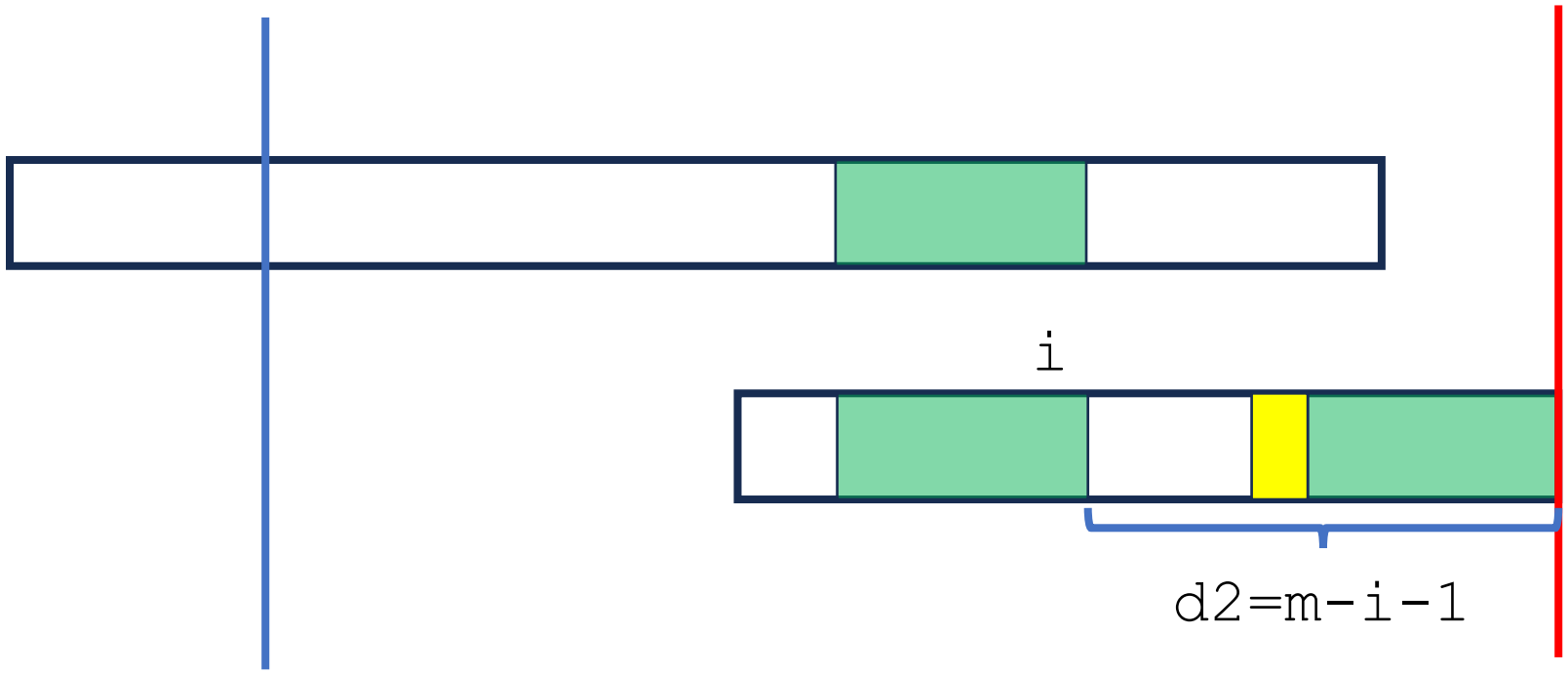
Boyer Moore 算法-好后缀规则



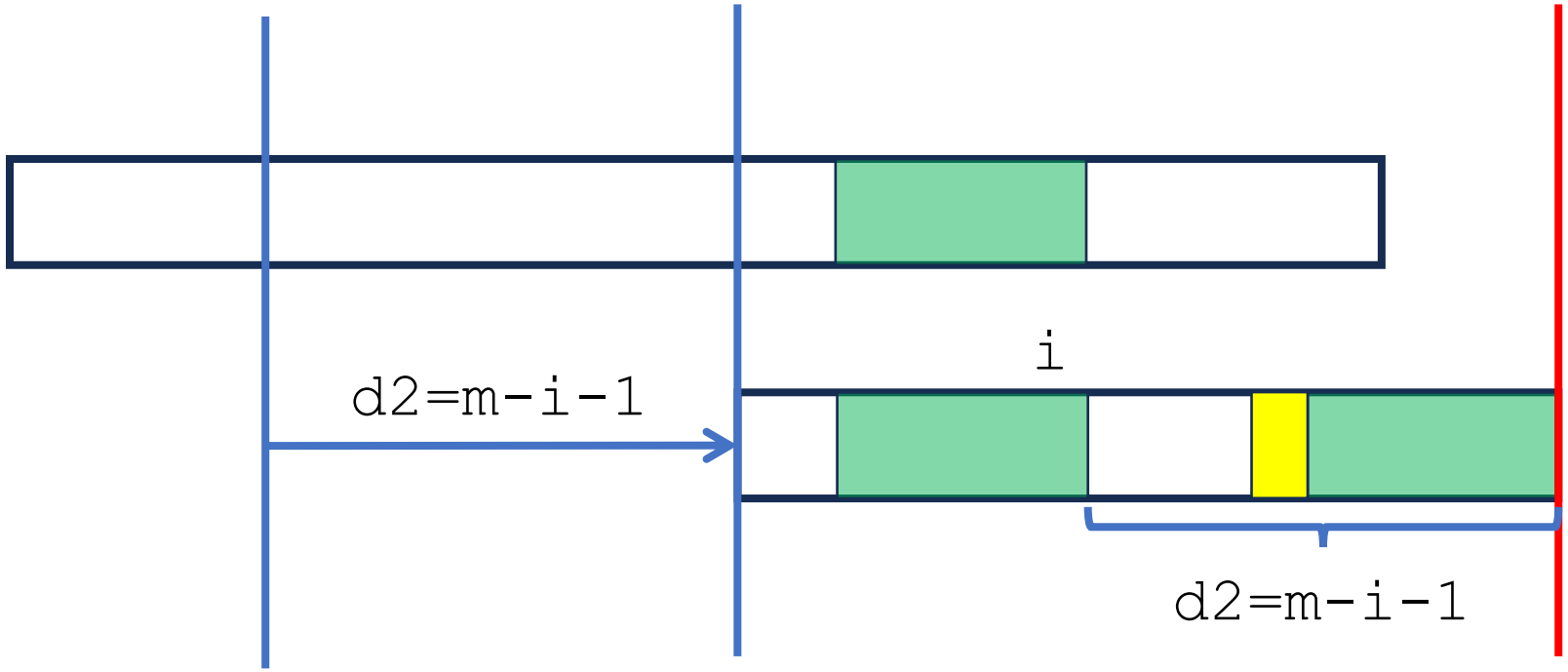
Boyer Moore 算法-好后缀规则



Boyer Moore 算法-好后缀规则



Boyer Moore 算法-好后缀规则

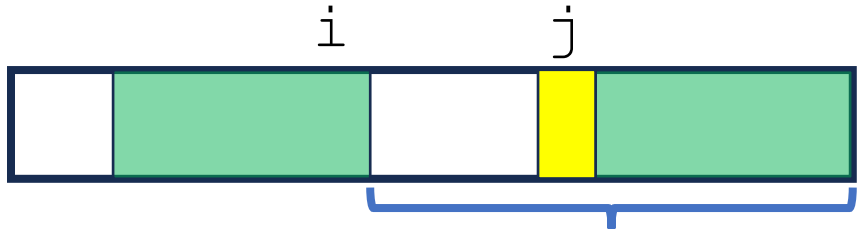


好后缀规则-情况1

情况1：

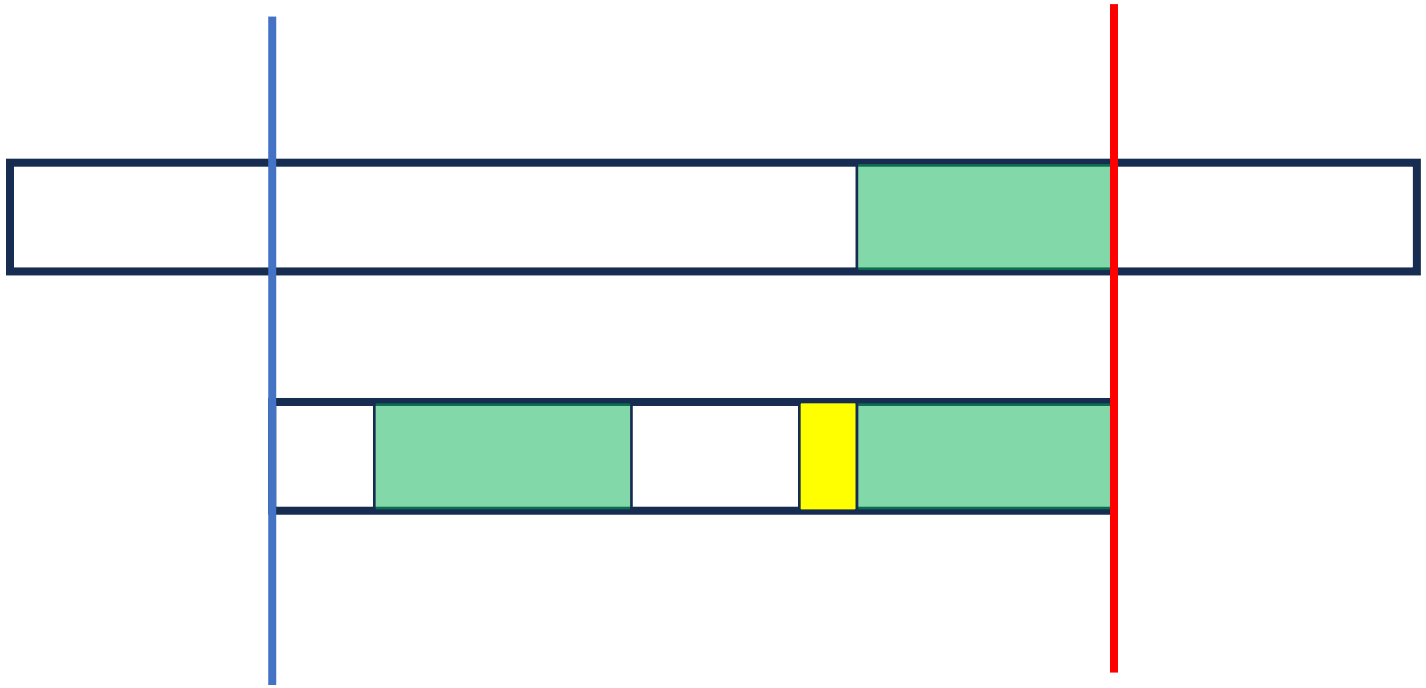
1. 在 j 位置前，能找到完整的后缀

2. 移动距离记录在 delta2[j] 中

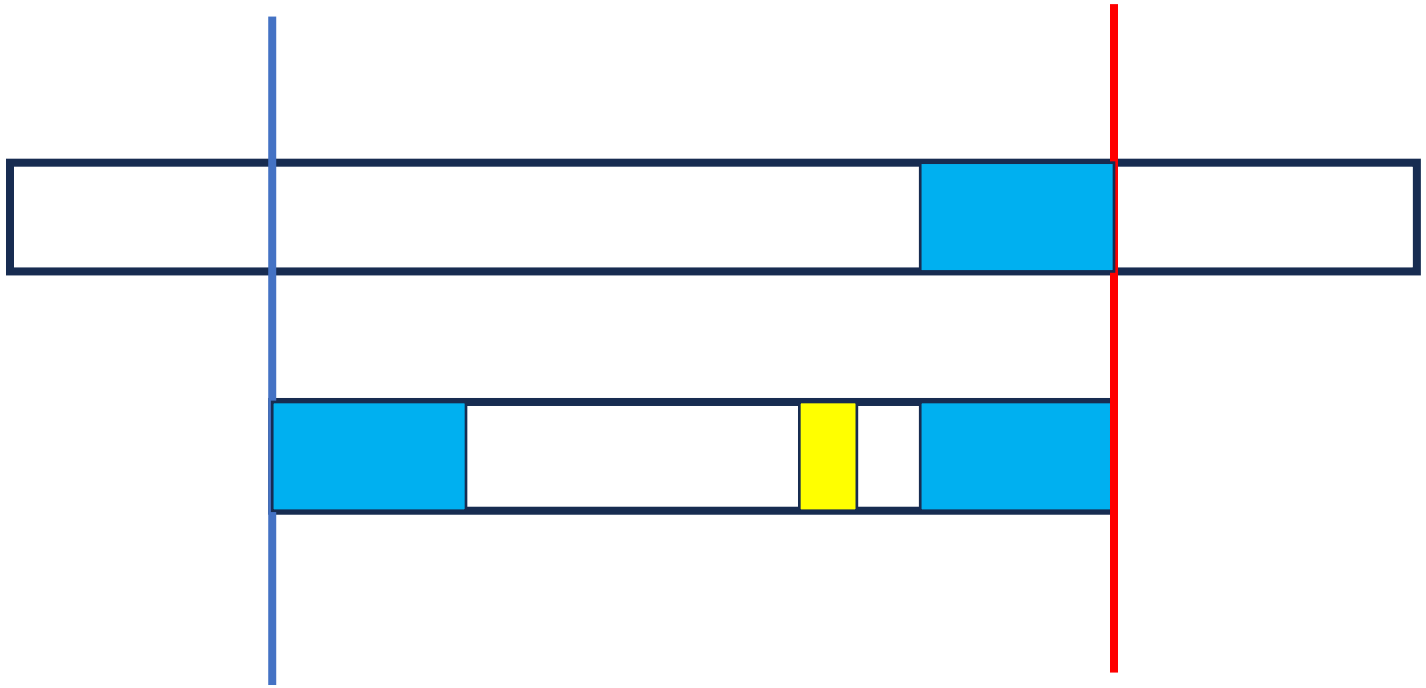


delta2[j]=d2=m-i-1

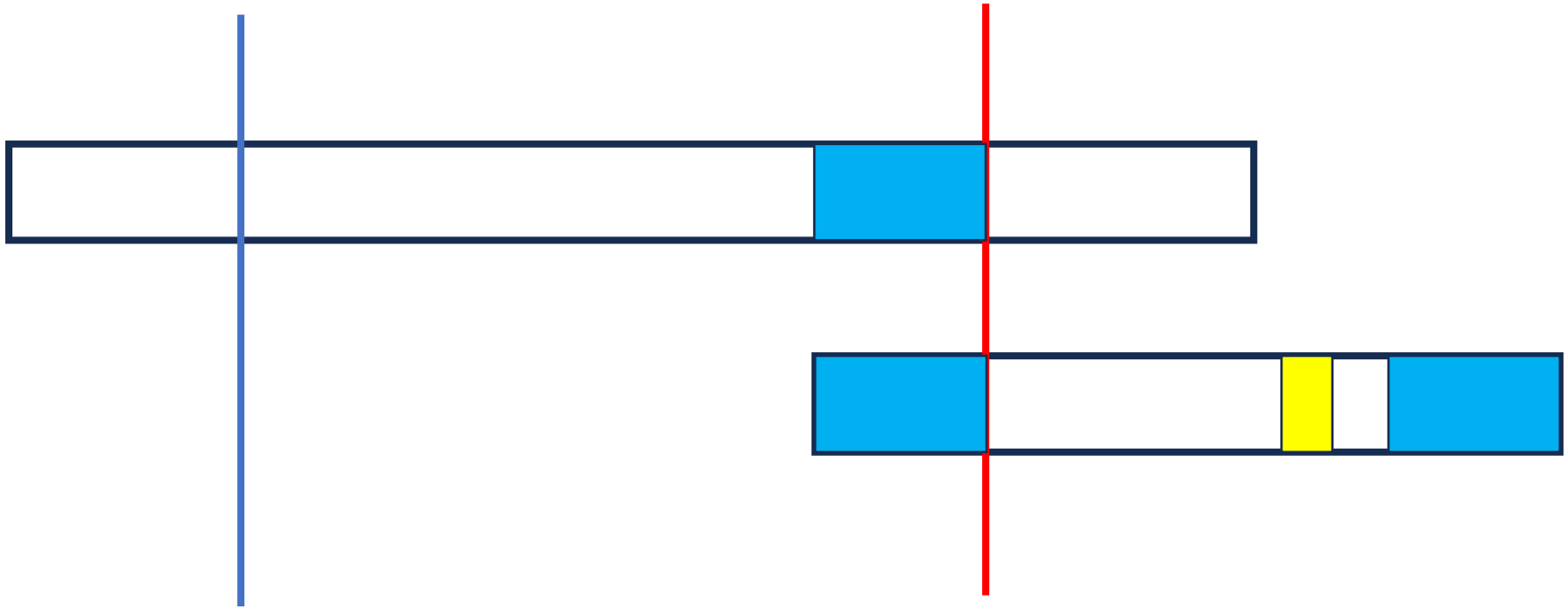
好后缀规则-情况2



好后缀规则-情况2

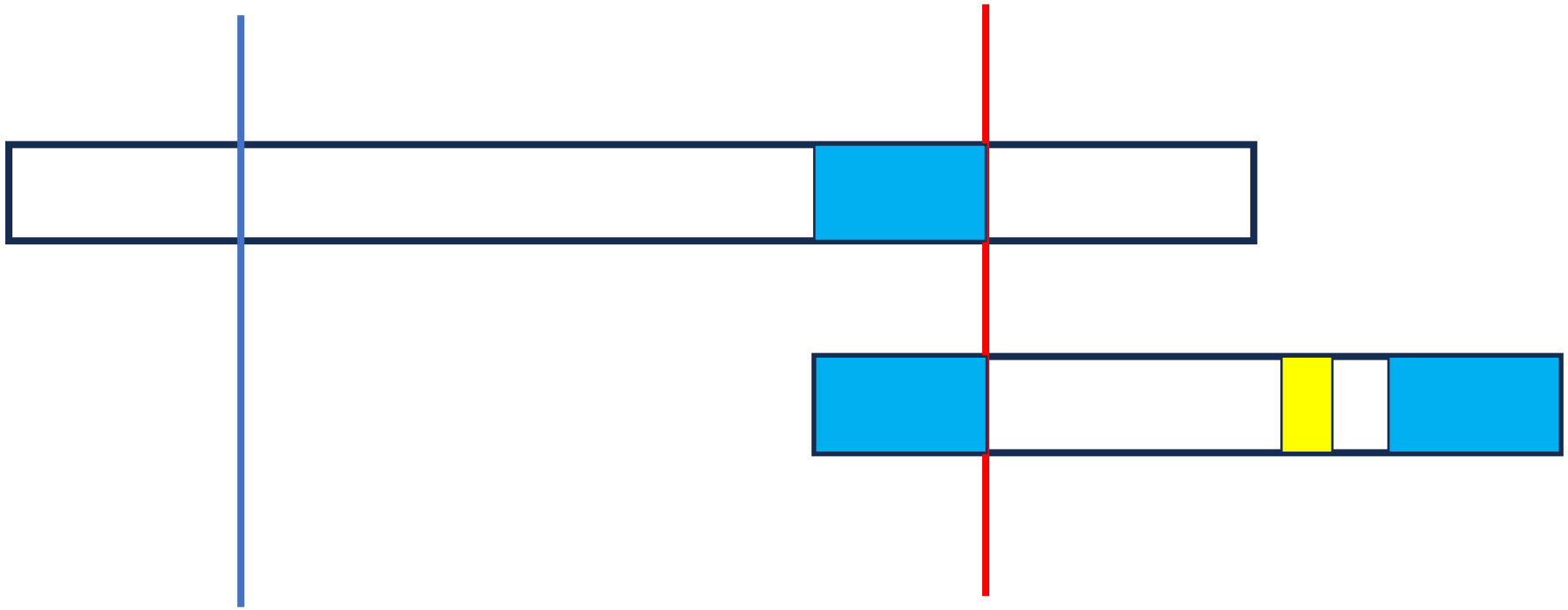


好后缀规则-情况2



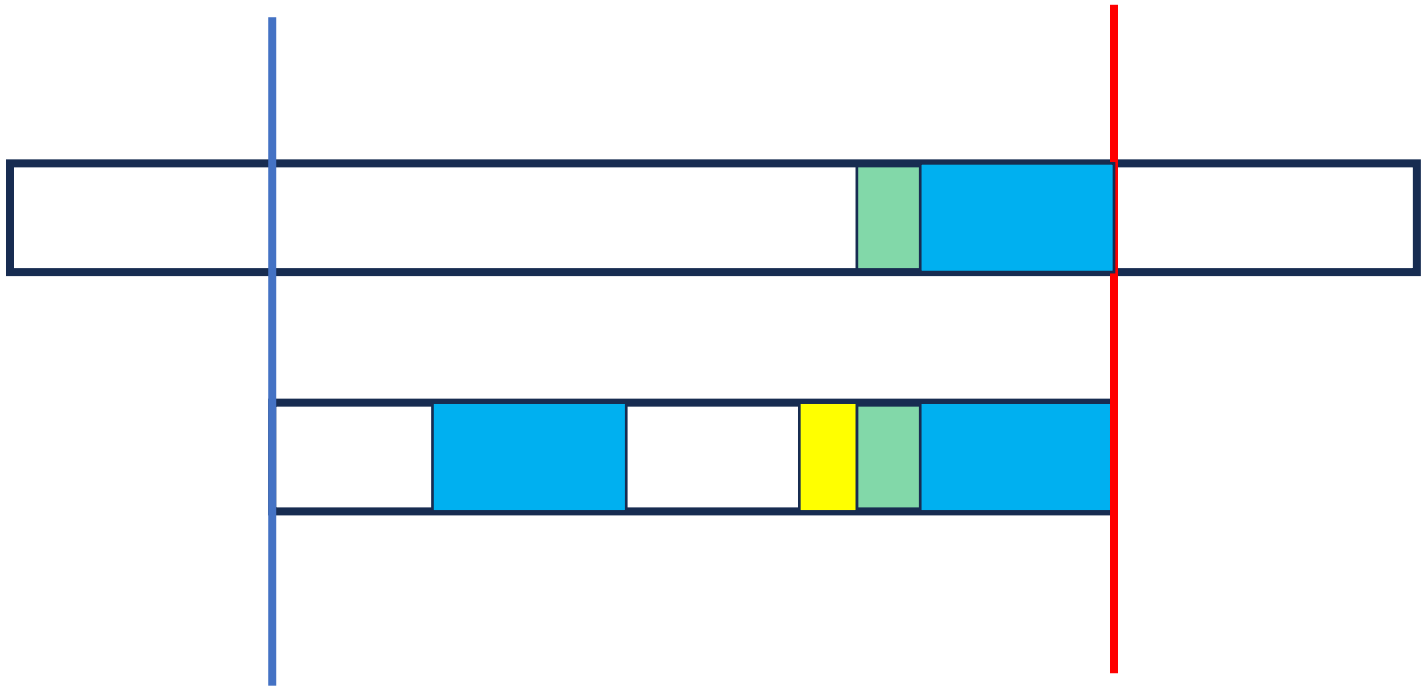
好后缀规则-情况2

问题：蓝色部分后缀的起始位置，一定是从模式串的开头匹配的么？



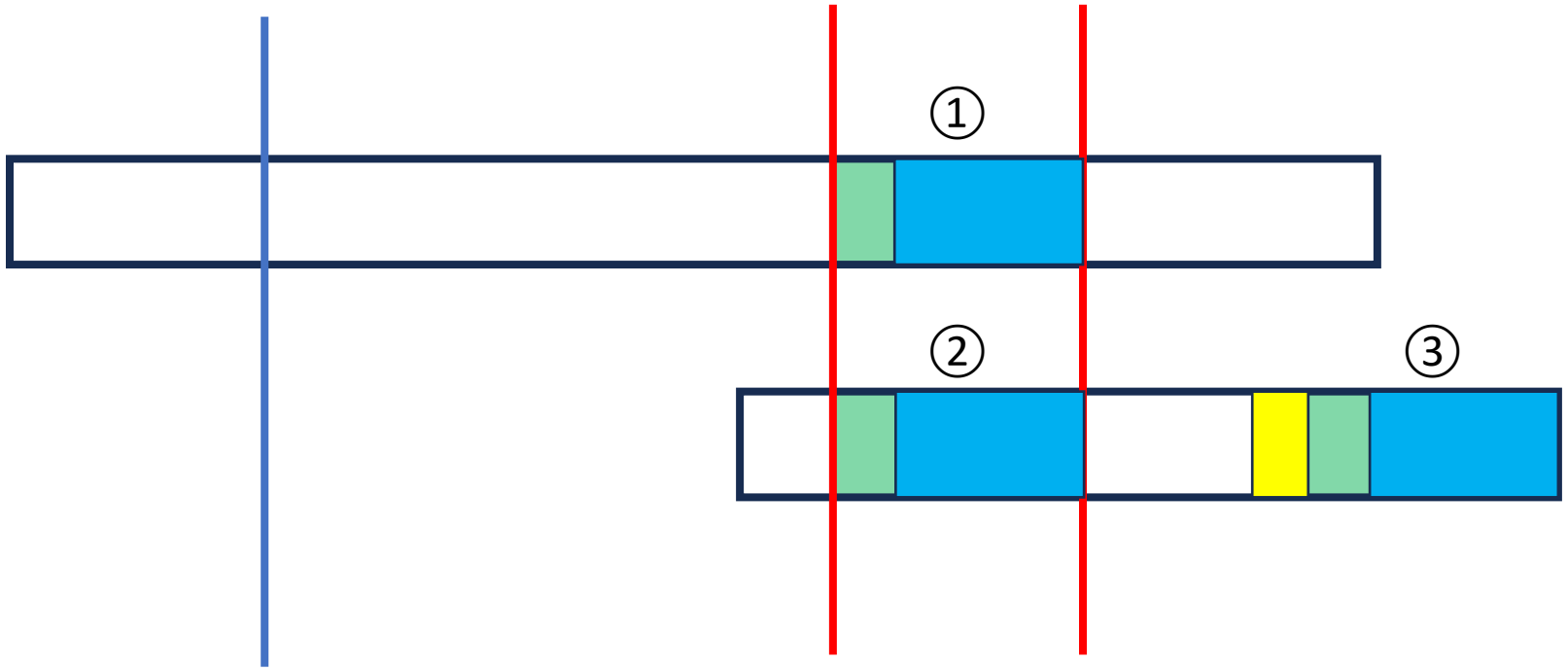
好后缀规则-情况2

问题：蓝色部分后缀的起始位置，一定是从模式串的开头匹配的么？



好后缀规则-情况2

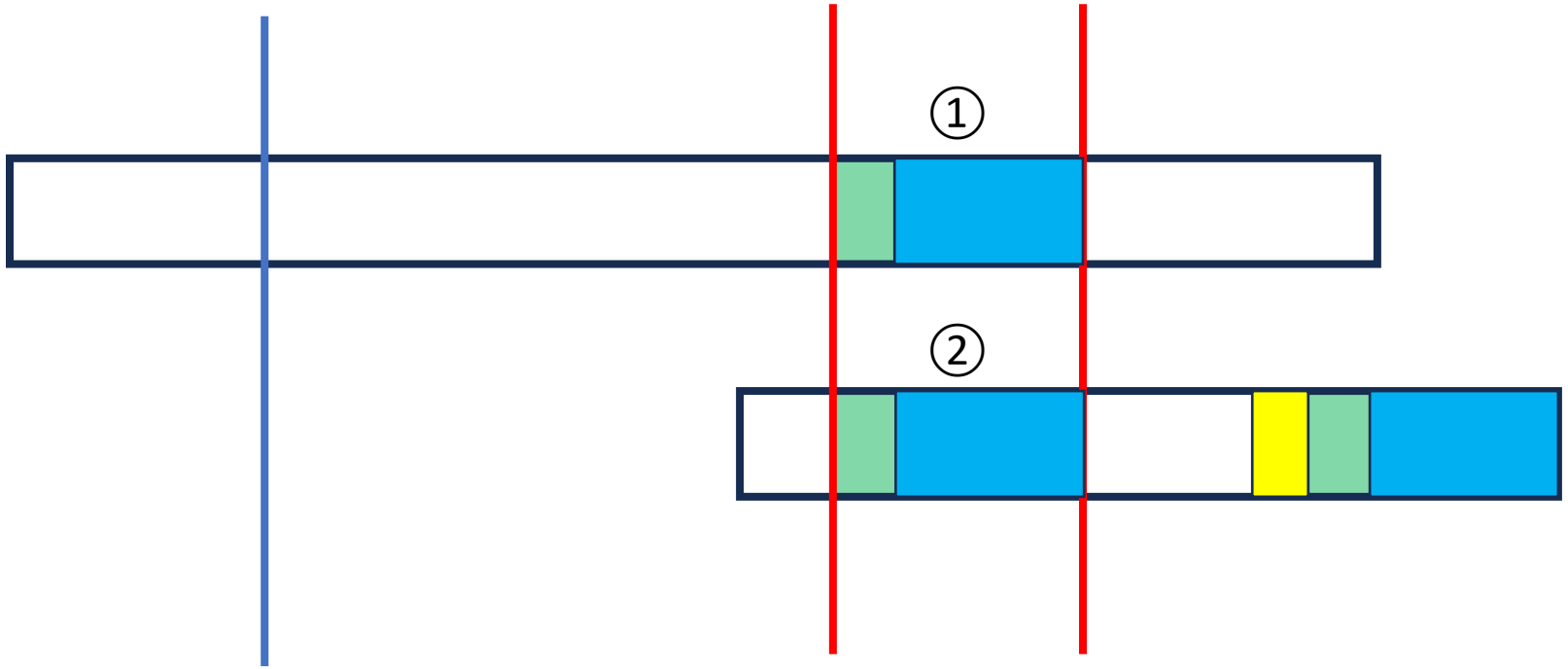
问题：蓝色部分后缀的起始位置，一定是从模式串的开头匹配的么？



好后缀规则-情况2

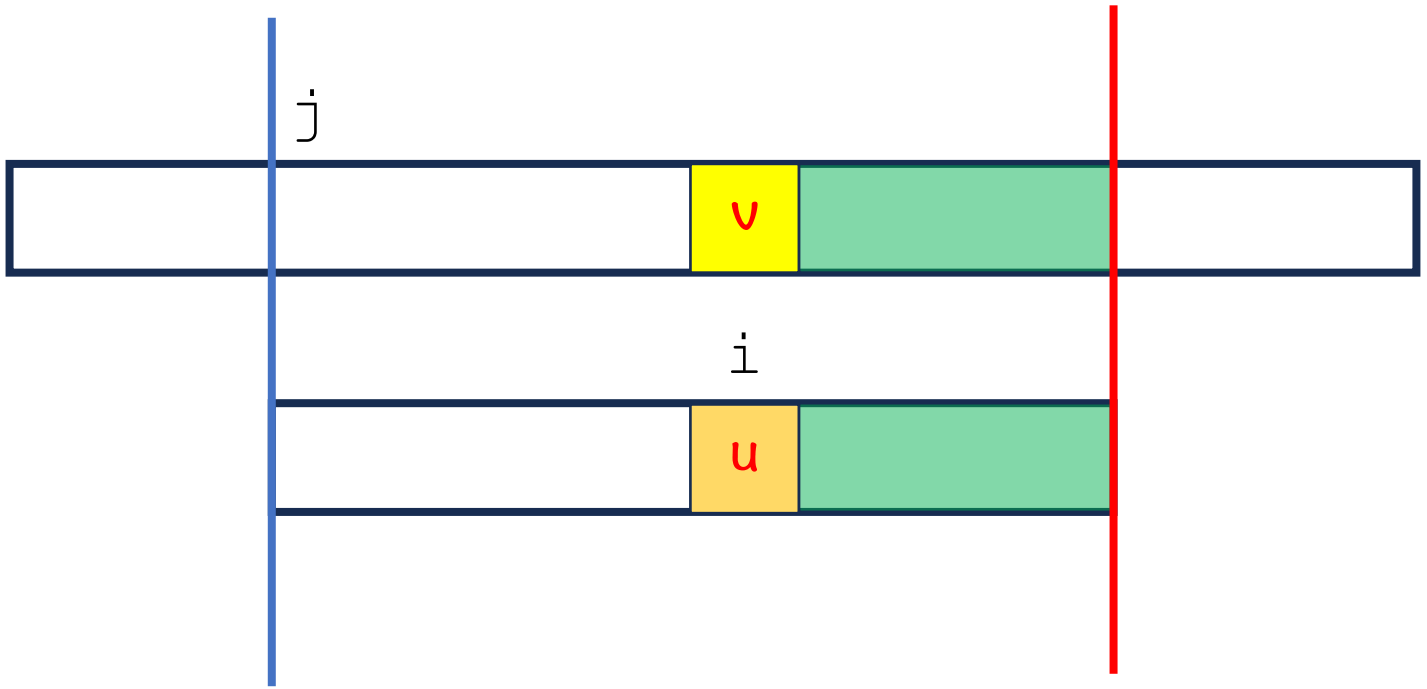
核心目标：移动以后，能匹配成功

1. 当初能找到完整的后缀匹配
2. 蓝色部分不是最长的后缀匹配



Boyer Moore 算法-总结

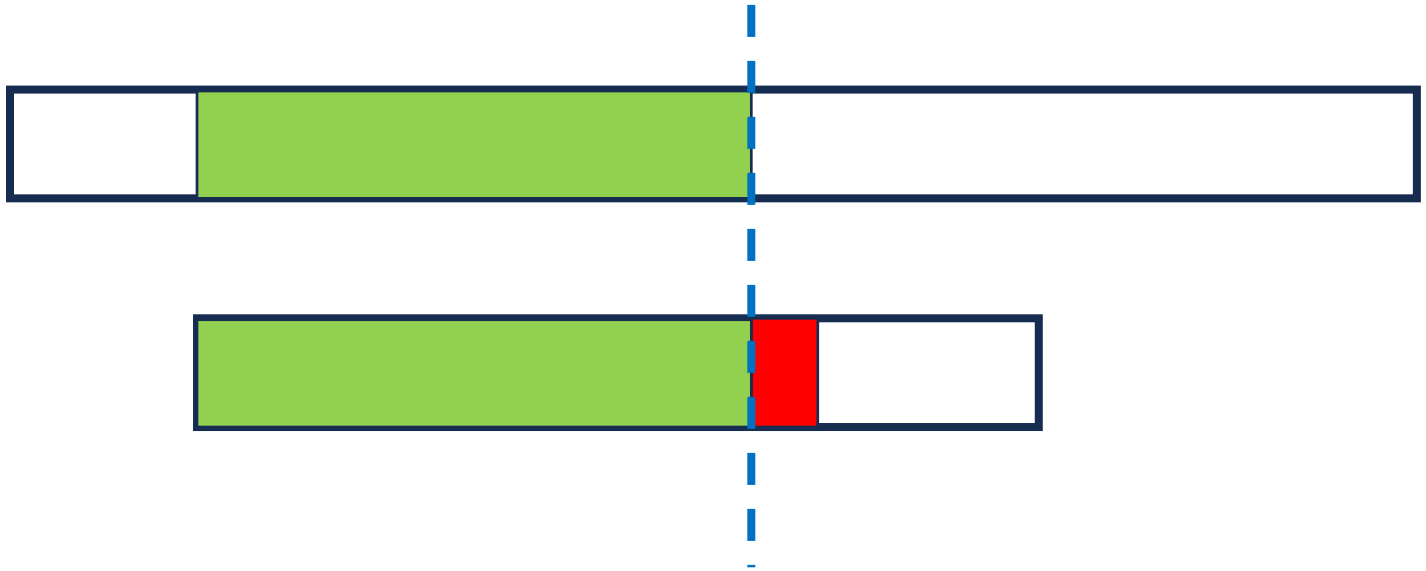
1. 通过字符 v，得到 delta1[v]
2. 通过位置 i，得到 delta2[i]
3. j += max(i-delta1[v], delta2[i])



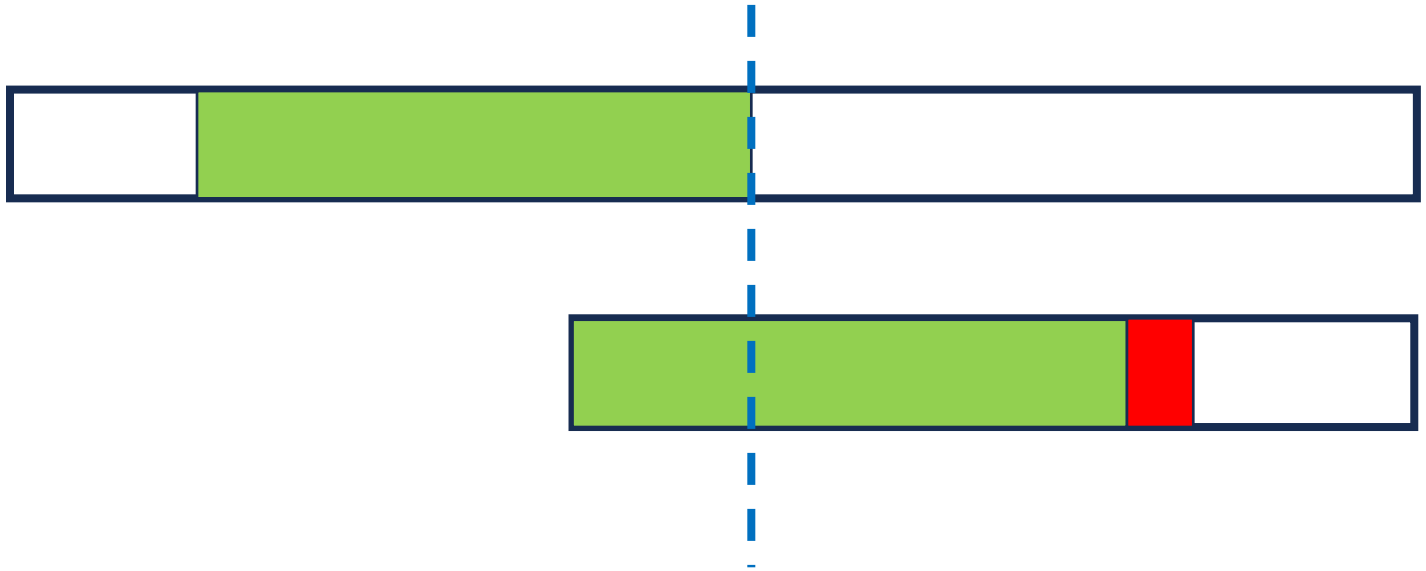
一、单模匹配问题

1. 易学易懂：Brute Force 算法
2. 高效方便：Sunday 算法
3. 经典回顾：Boyer Moore 算法
4. **变化多端：KMP 算法**

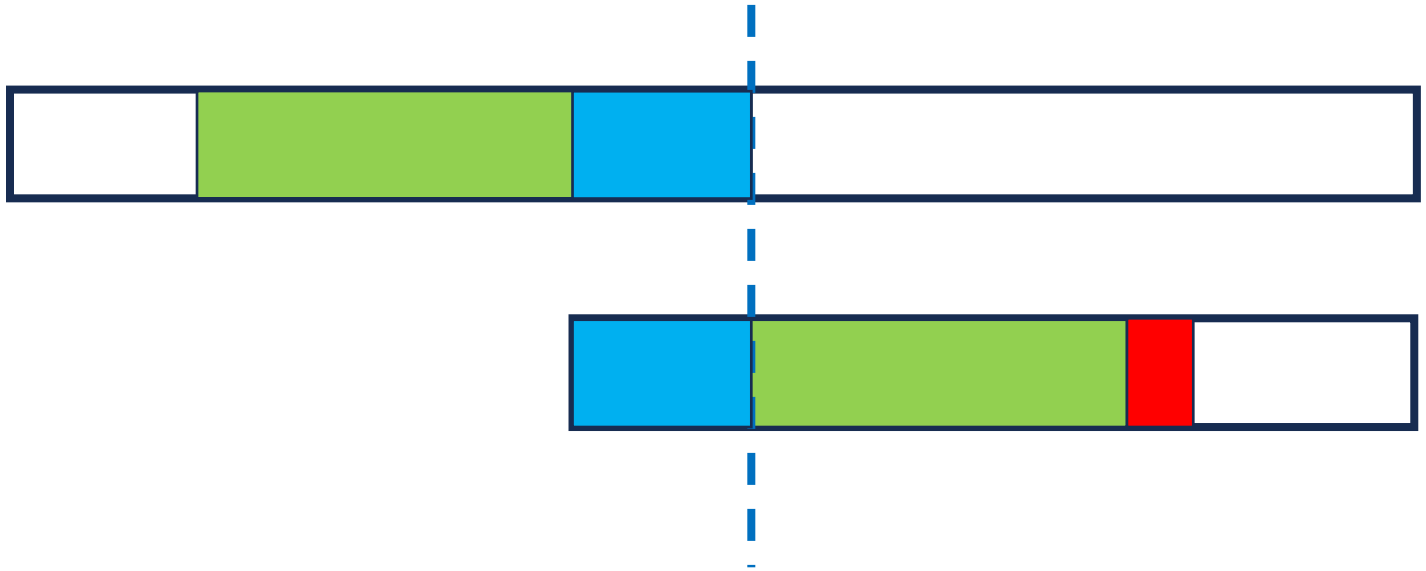
KMP 算法



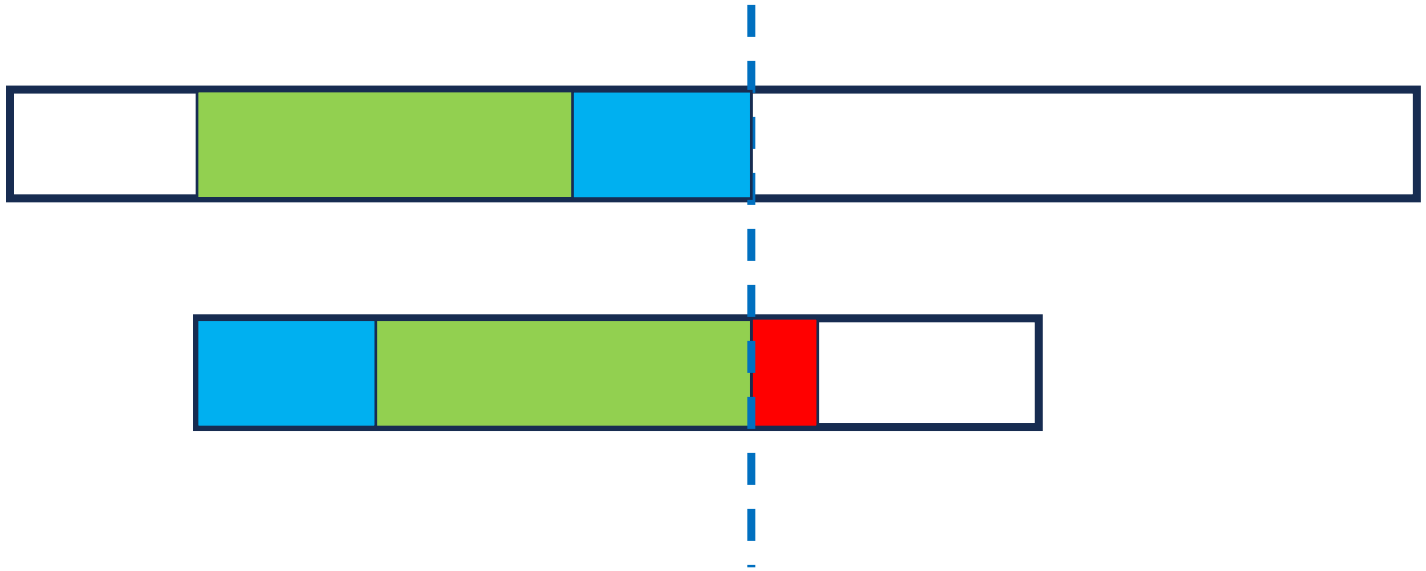
KMP 算法



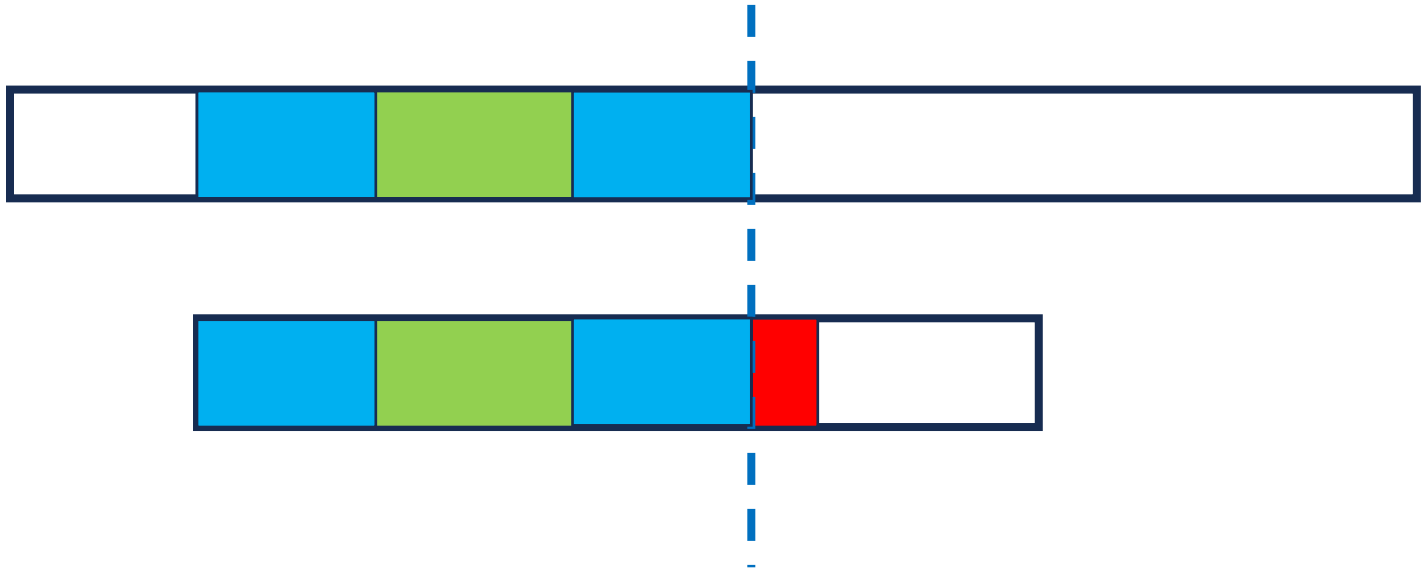
KMP 算法



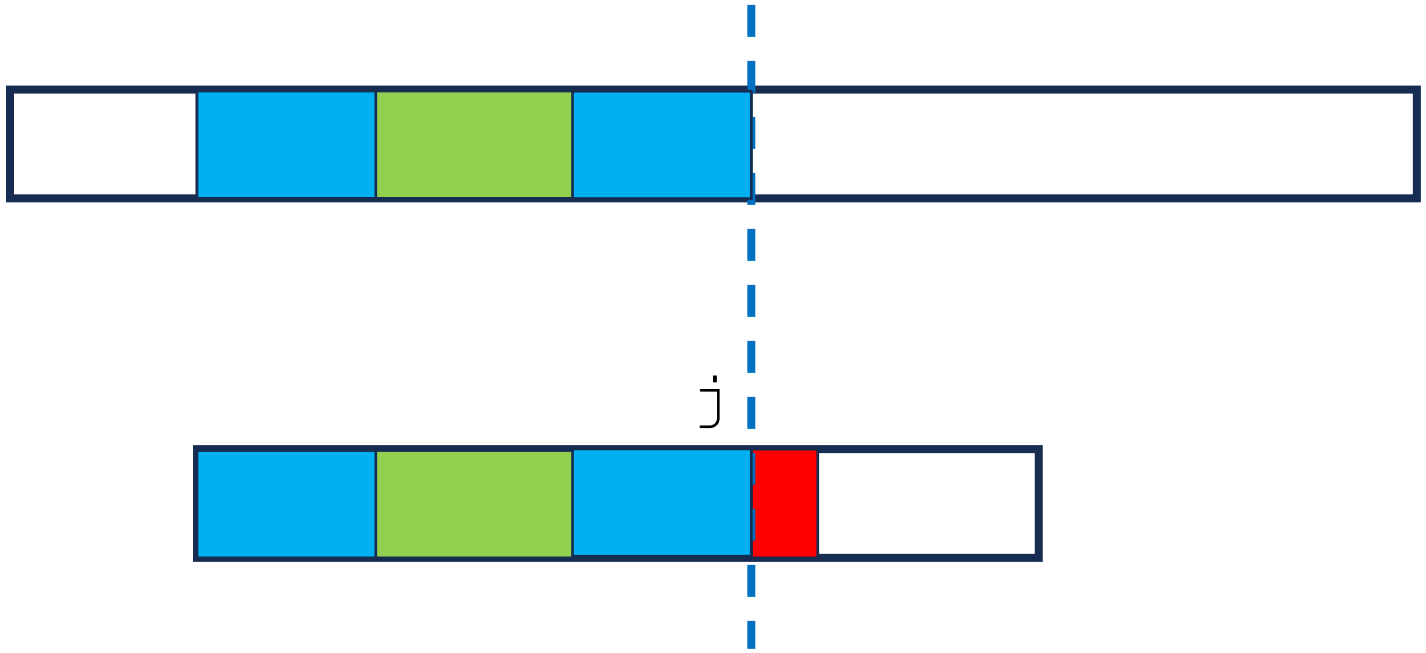
KMP 算法



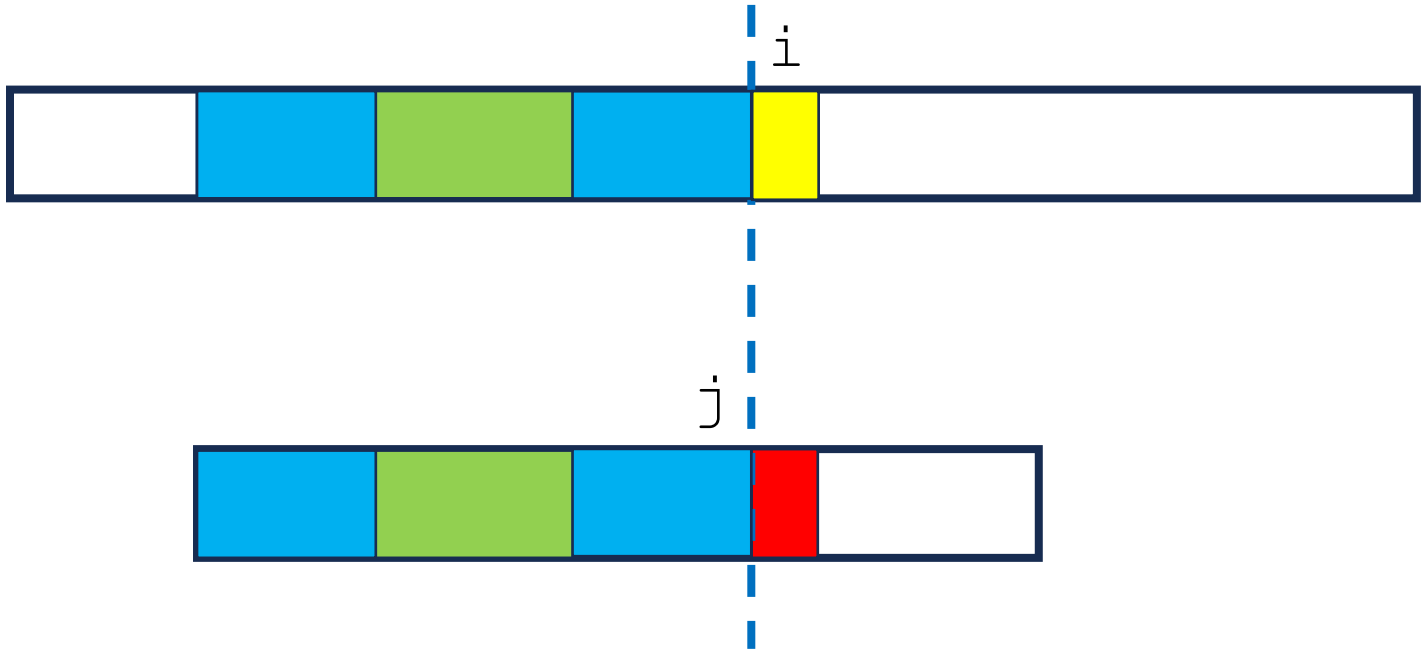
KMP 算法



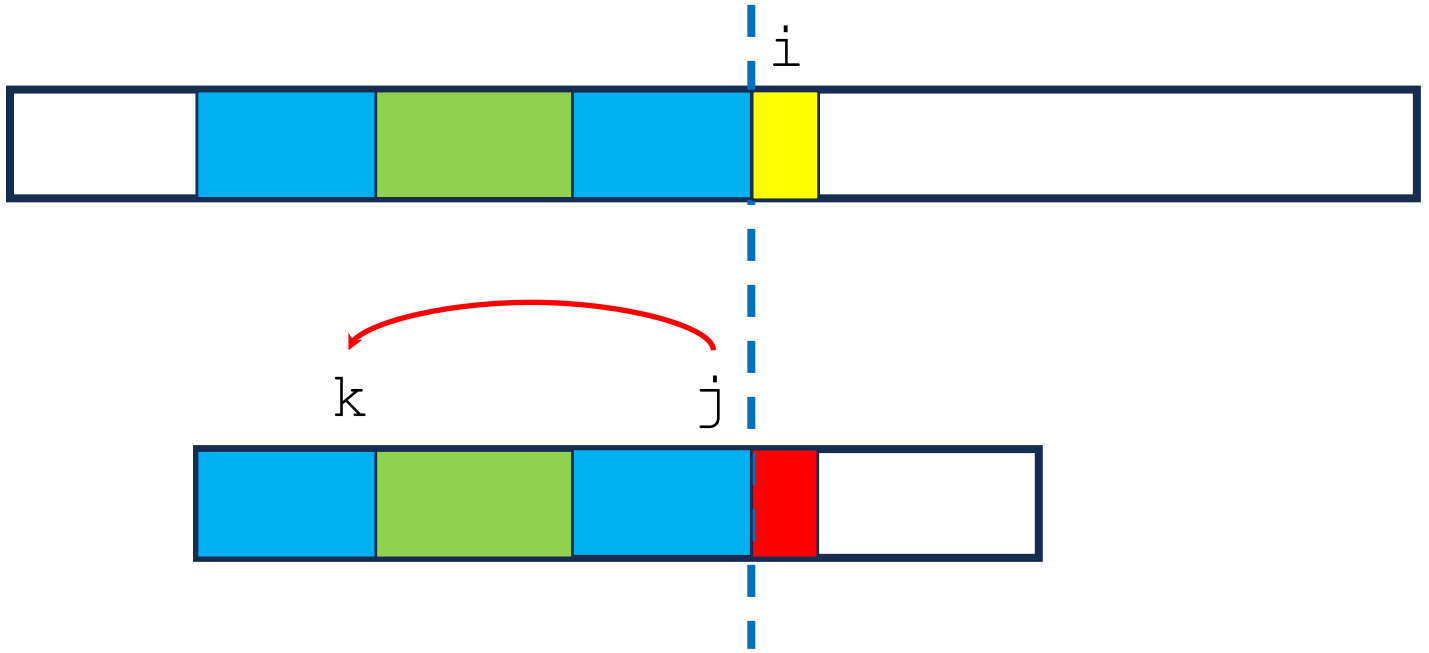
KMP 算法



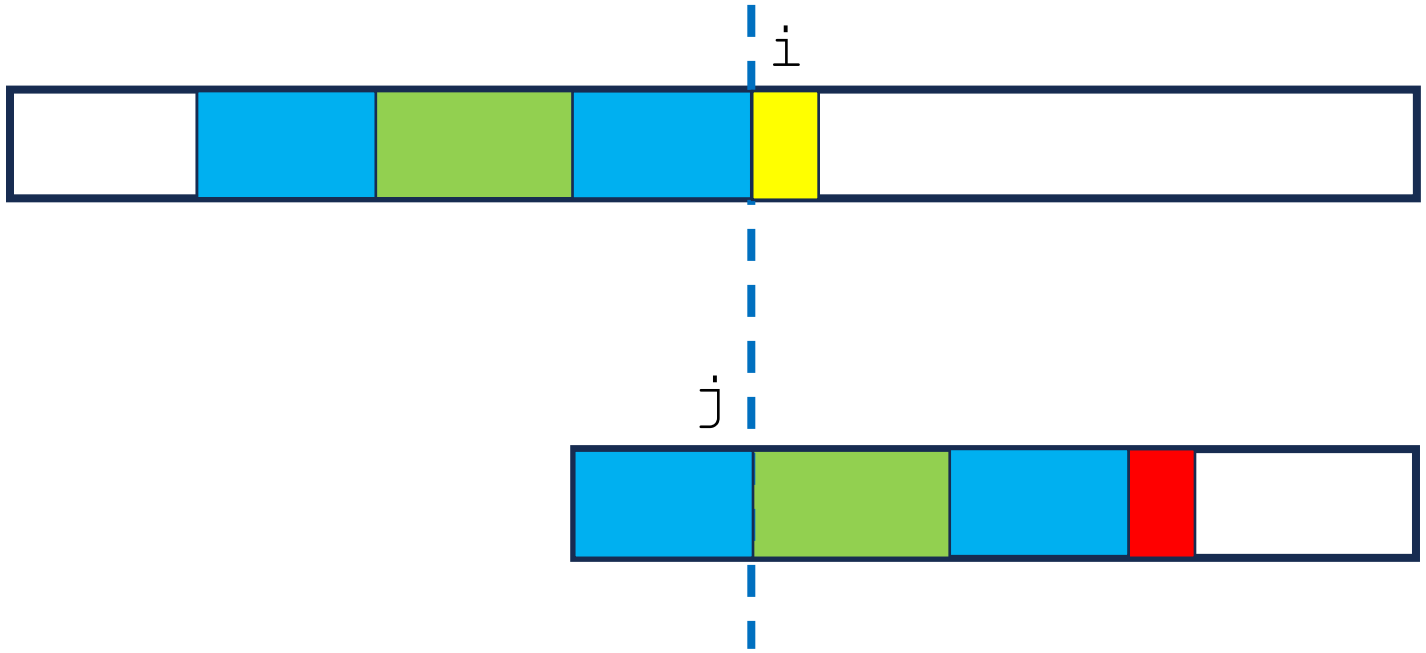
KMP 算法



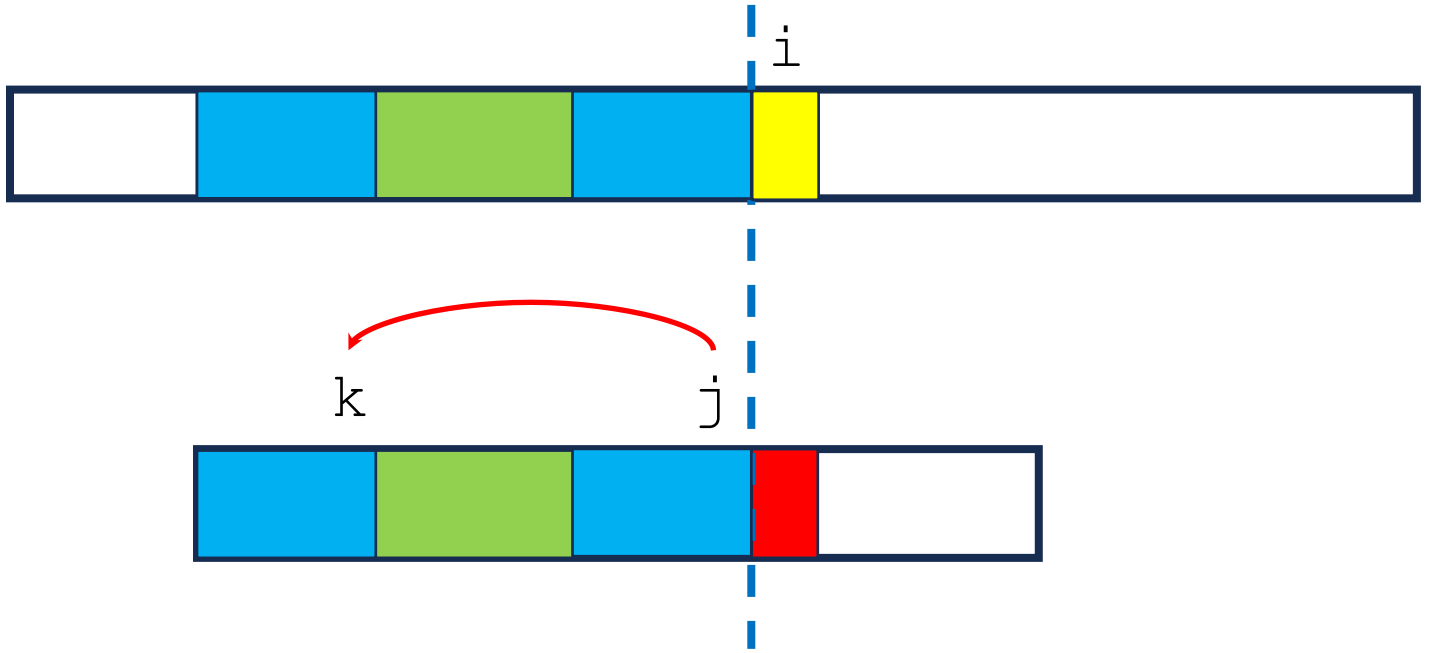
KMP 算法



KMP 算法

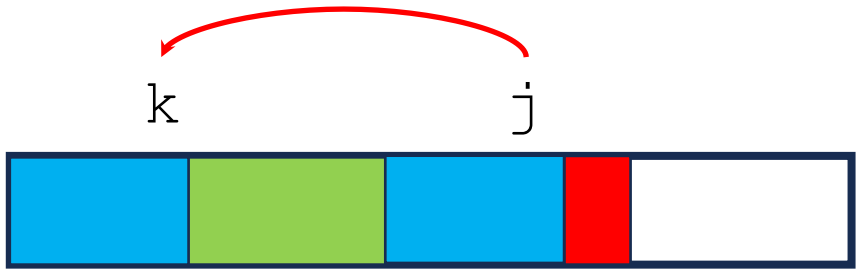


KMP 算法



KMP 算法-next数组

1. 最长前缀信息存储在 next 数组中，next[j] = k
2. next[j] 的值可以通过之前的 next 值求得



KMP 算法-next数组

1. 最长前缀信息存储在 next 数组中，next[j] = k

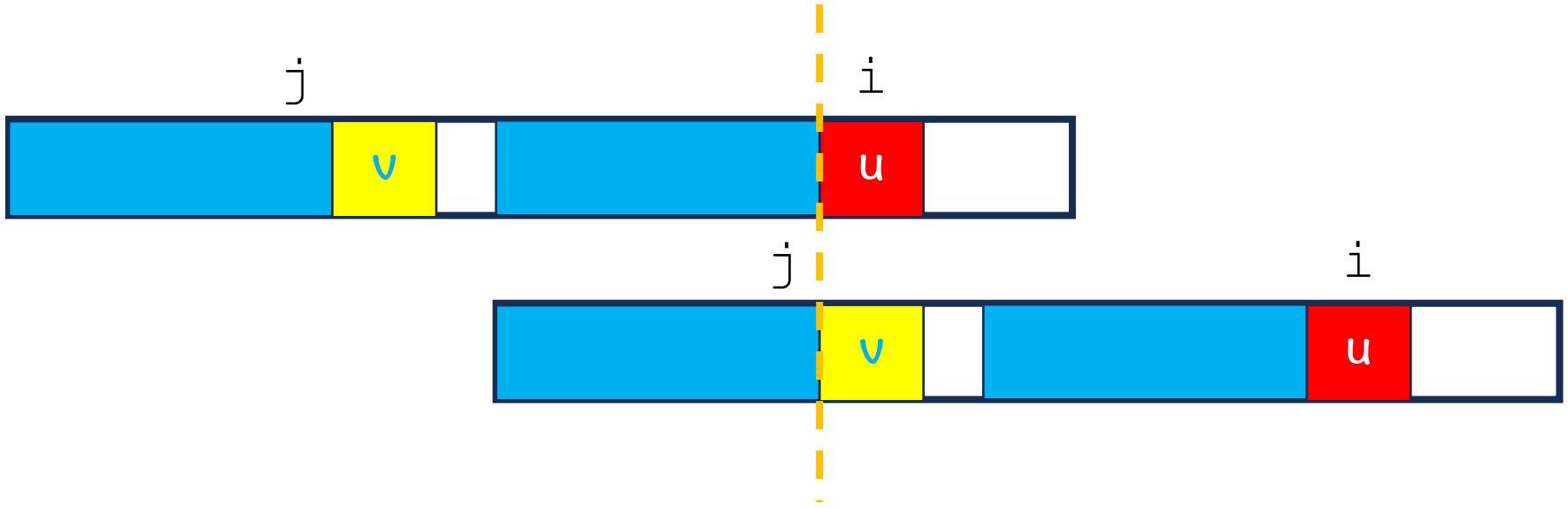
2. next[i] 的值可以通过之前的 next 值求得

KMP 算法-next数组

1. 最长前缀信息存储在 next 数组中，next[j] = k
2. next[i] 的值可以通过之前的 next 值求得
3. 若 u == v，next[i] = j+1

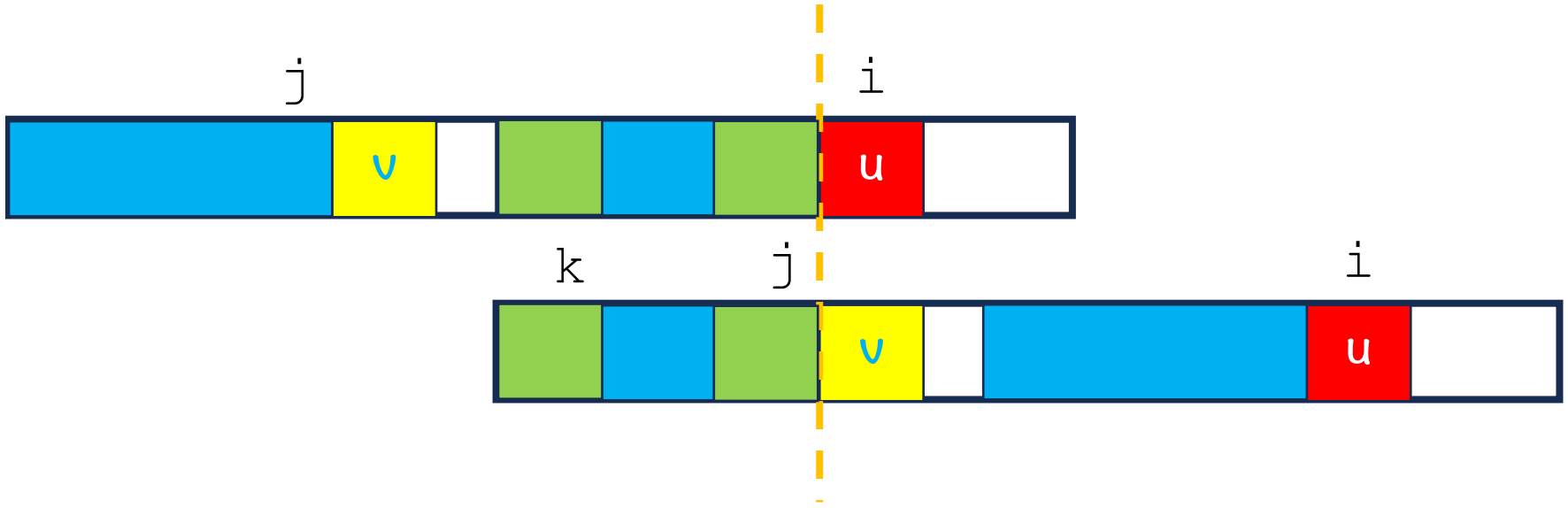
KMP 算法-next数组

1. 最长前缀信息存储在 next 数组中，next[j] = k
2. next[i] 的值可以通过之前的 next 值求得
3. 若 u == v，next[i] = j+1



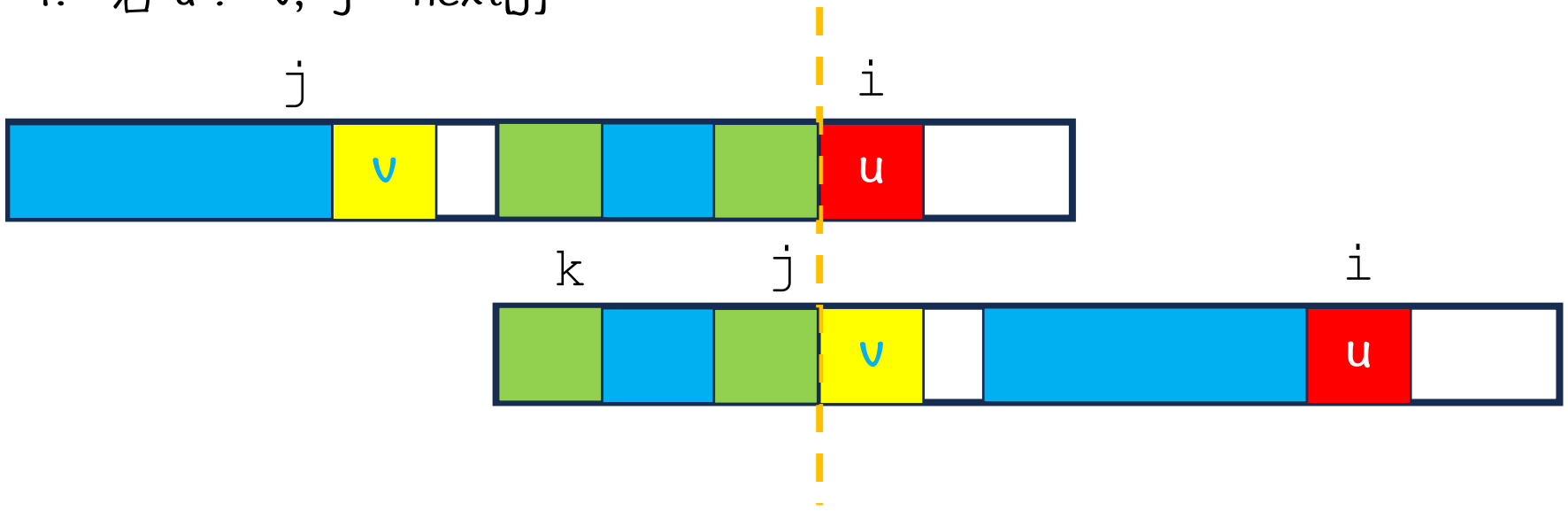
KMP 算法-next数组

1. 最长前缀信息存储在 next 数组中，next[j] = k
2. next[i] 的值可以通过之前的 next 值求得
3. 若 u == v，next[i] = j+1



KMP 算法-next数组

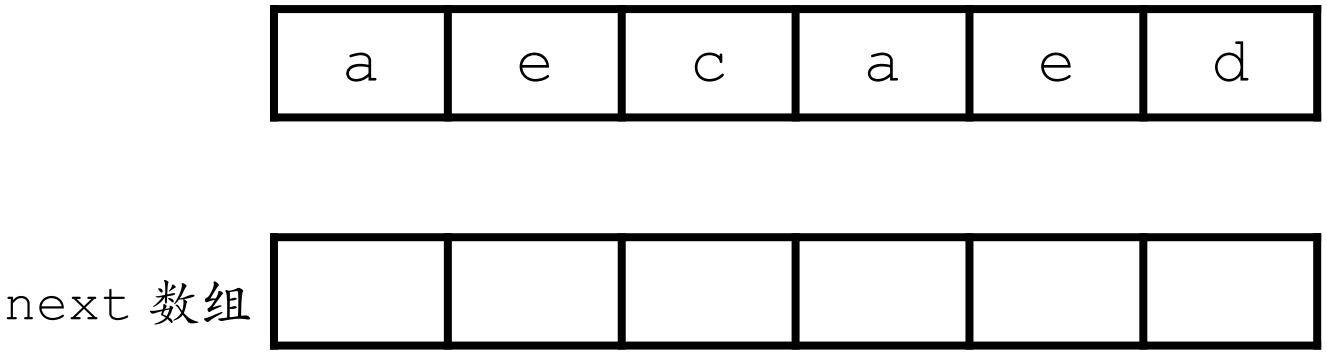
1. 最长前缀信息存储在 next 数组中，next[j] = k
2. next[i] 的值可以通过之前的 next 值求得
3. 若 u == v，next[i] = j+1
4. 若 u != v，j = next[j]



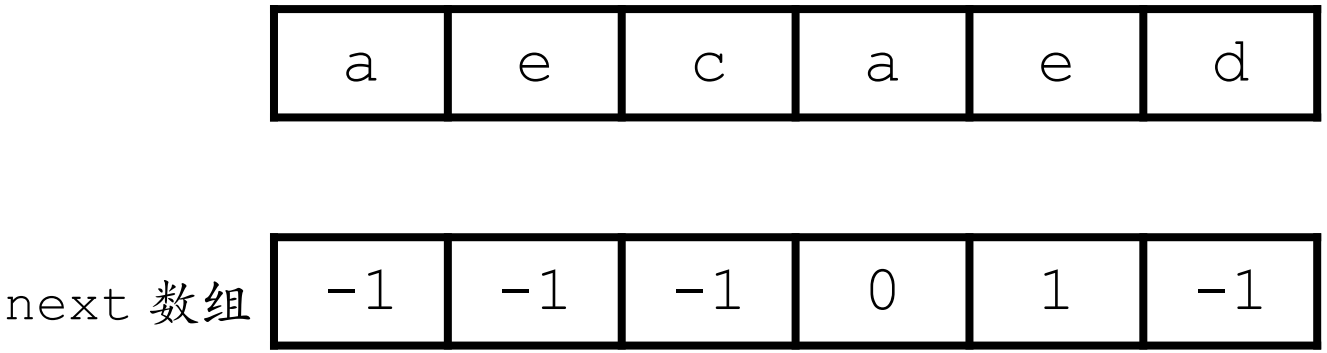
KMP 算法-next数组

1. 最长前缀信息存储在 next 数组中，next[j] = k
2. next[i] 的值可以通过之前的 next 值求得
3. 若 u == v，next[i] = j+1
4. 若 u != v，j = next[j]

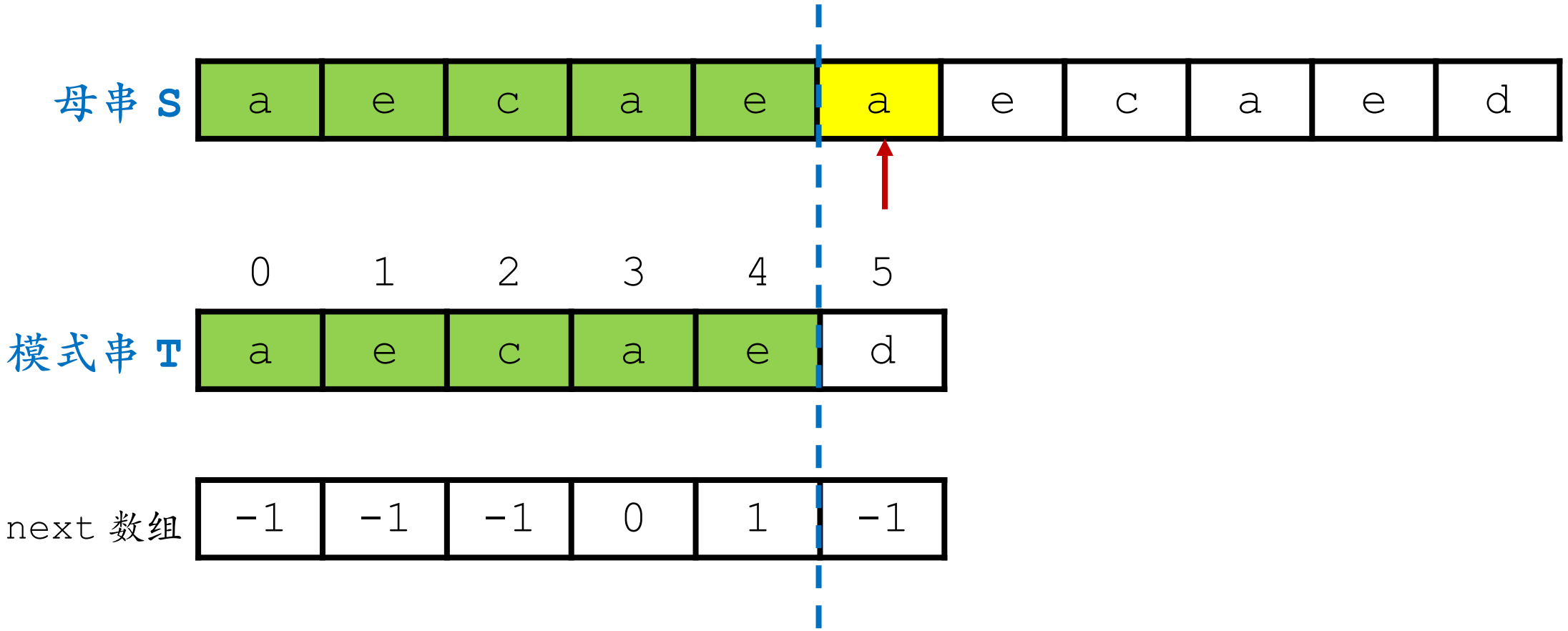
KMP 算法-next数组



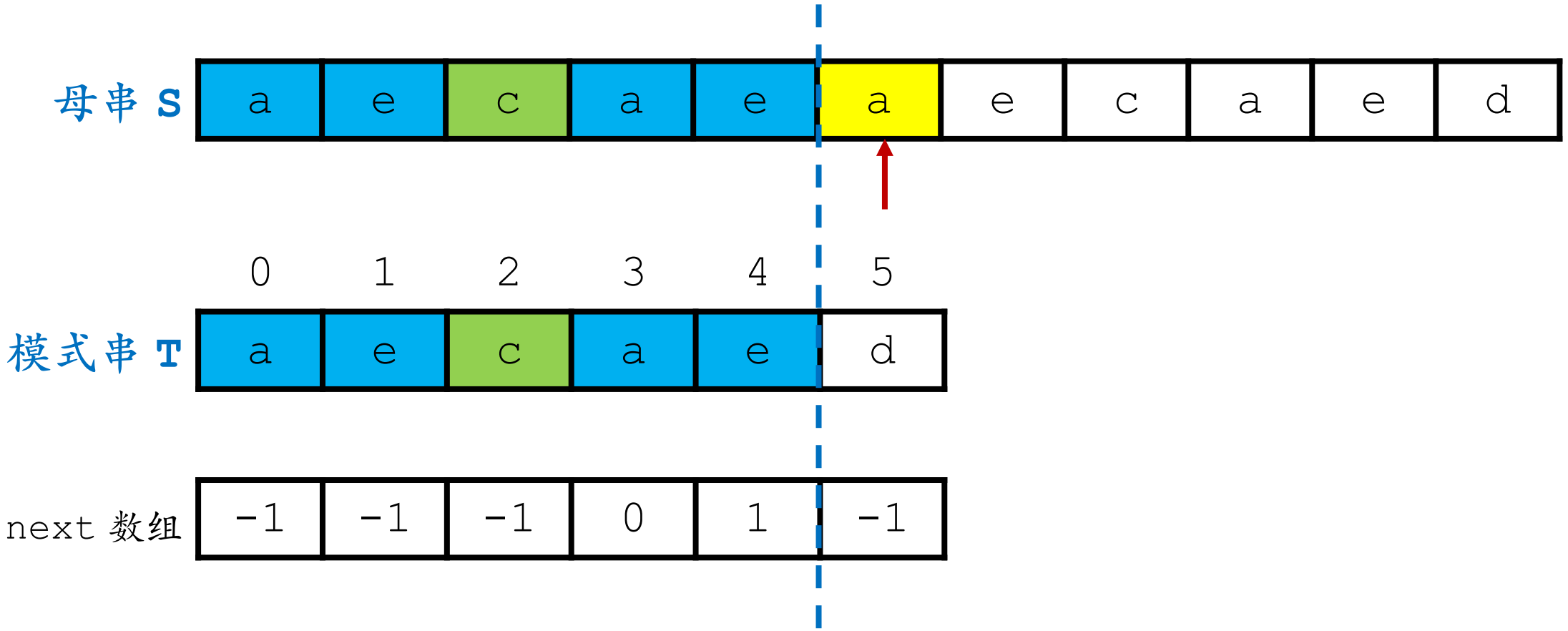
KMP 算法-next数组



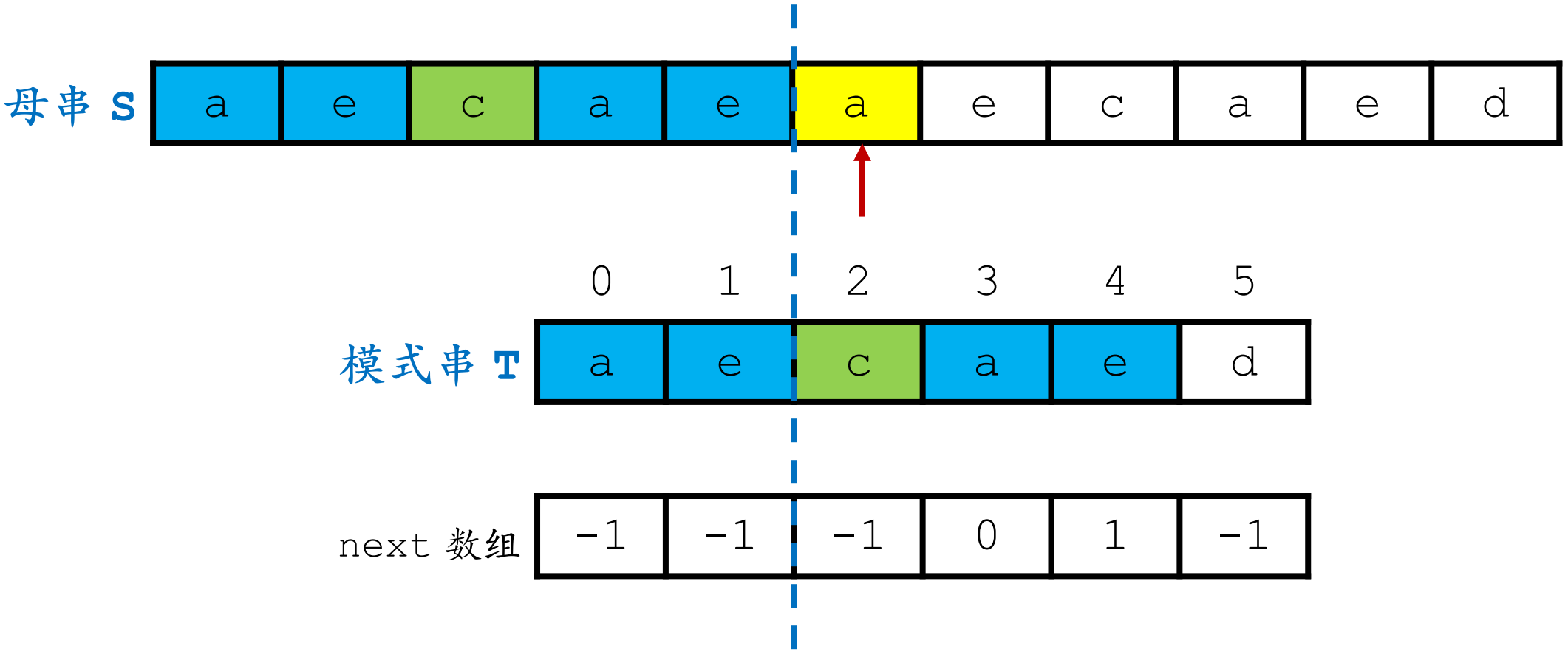
KMP 算法-next数组



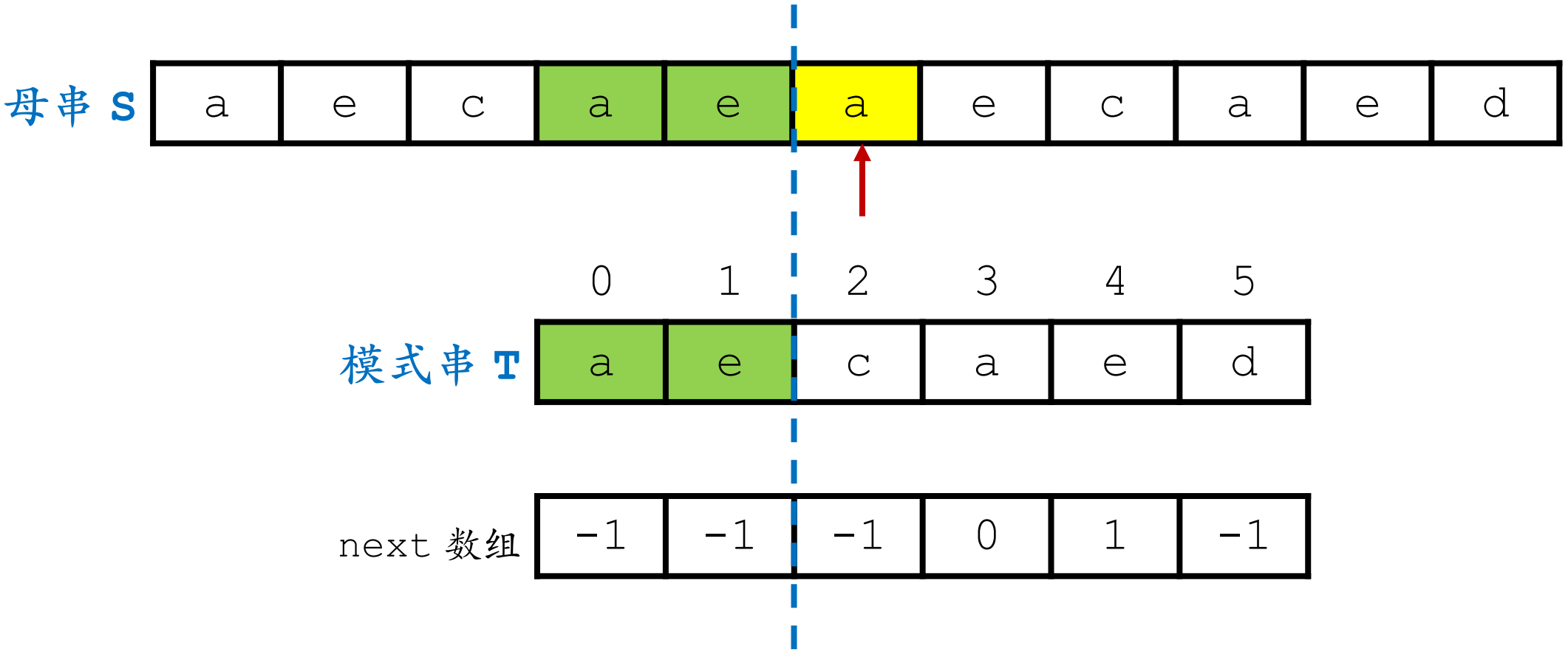
KMP 算法-next数组



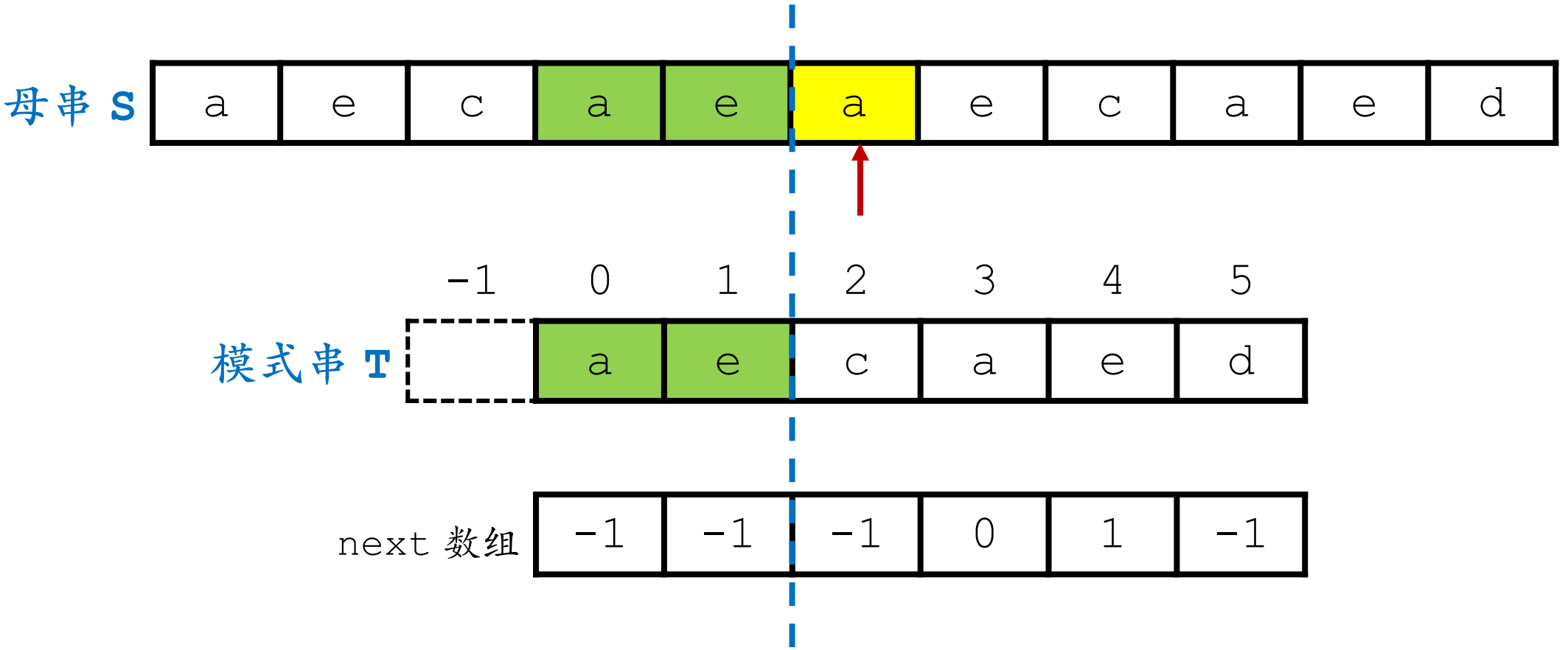
KMP 算法-next数组



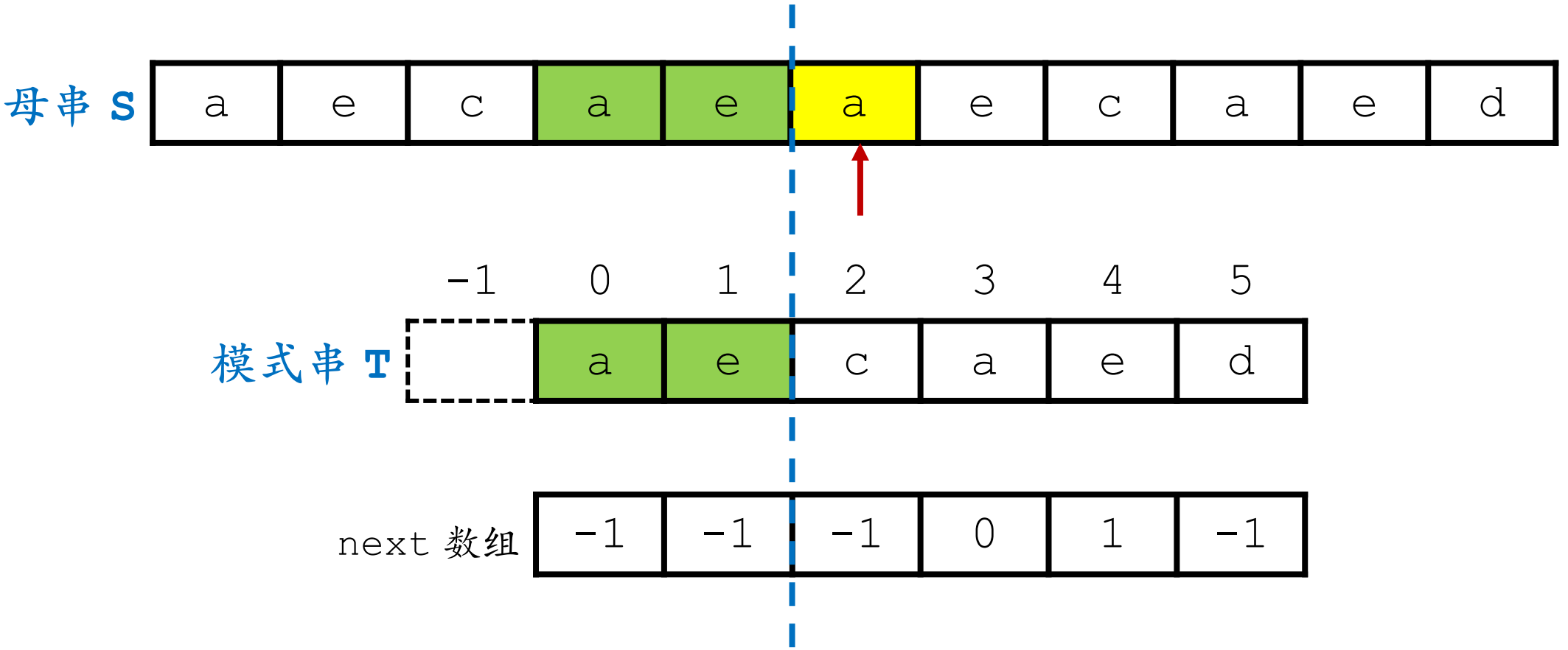
KMP 算法-next数组



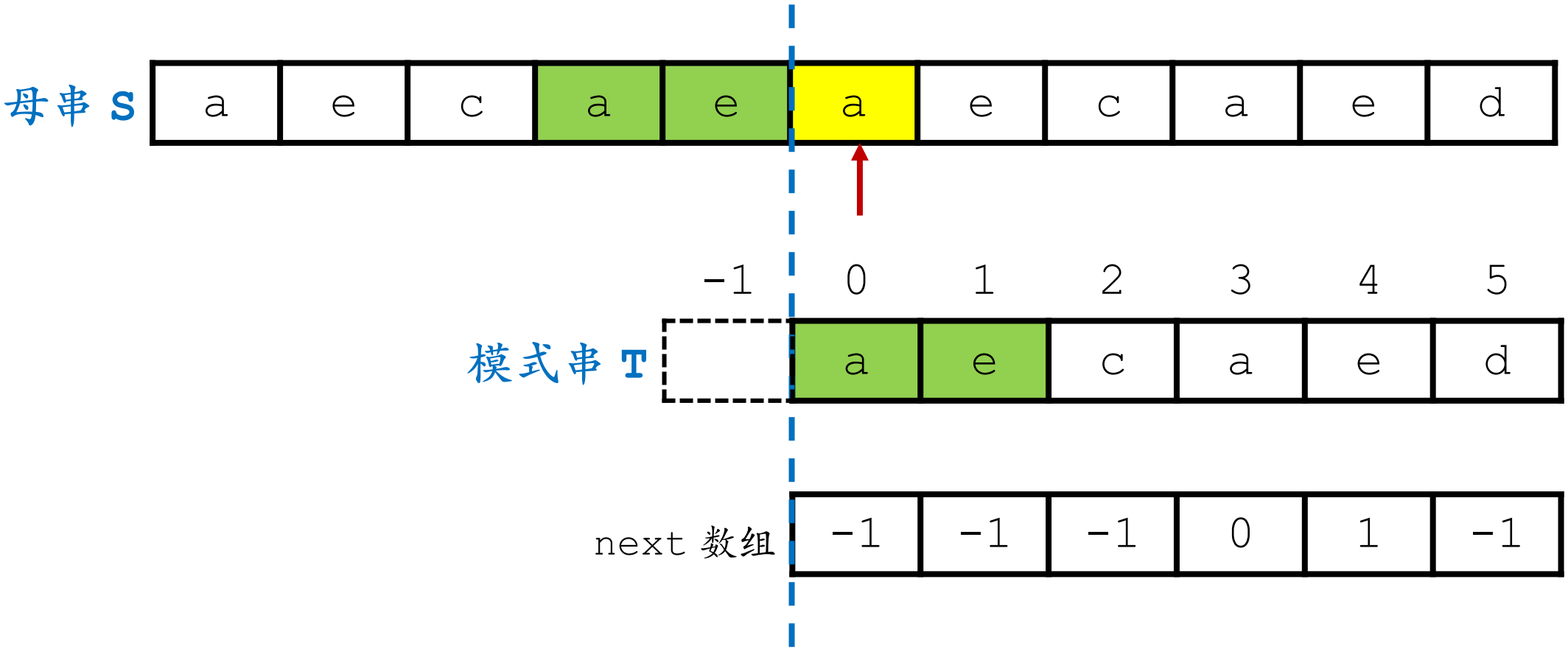
KMP 算法-next数组



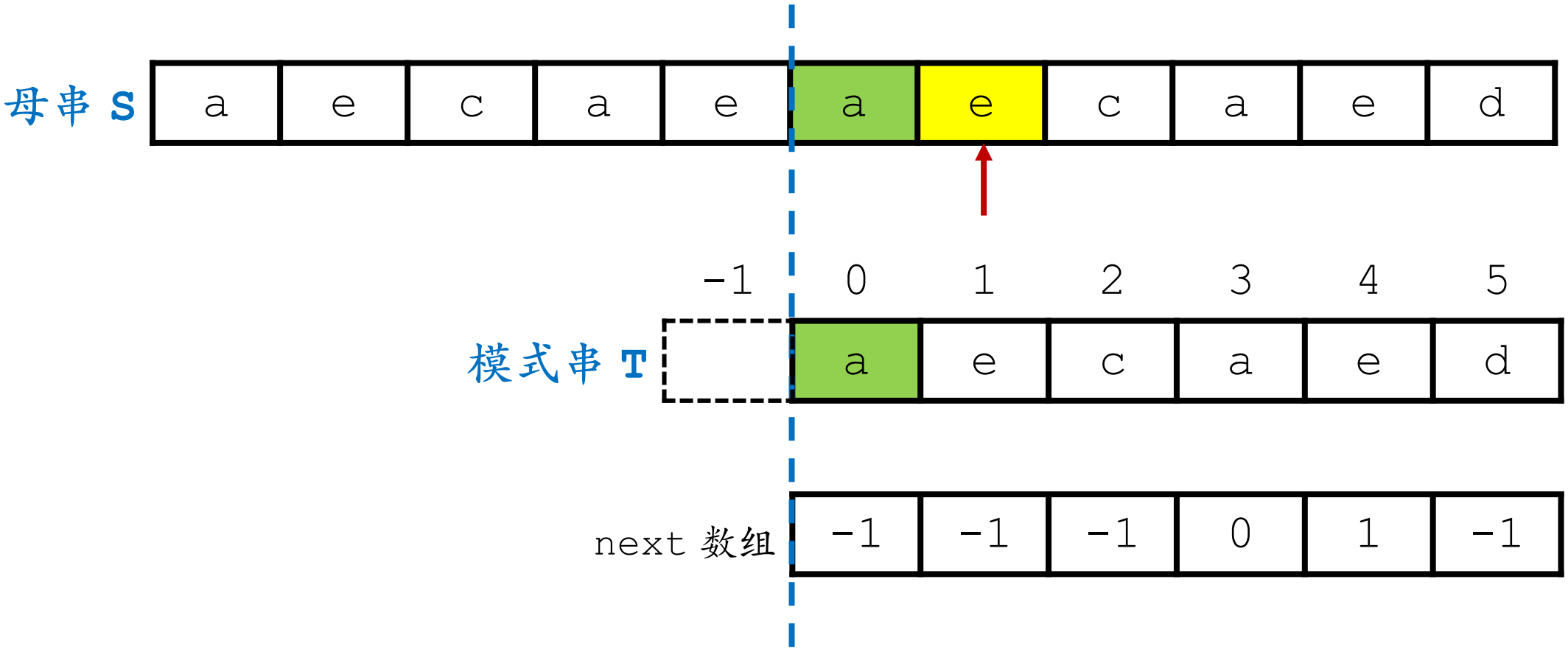
KMP 算法-next数组



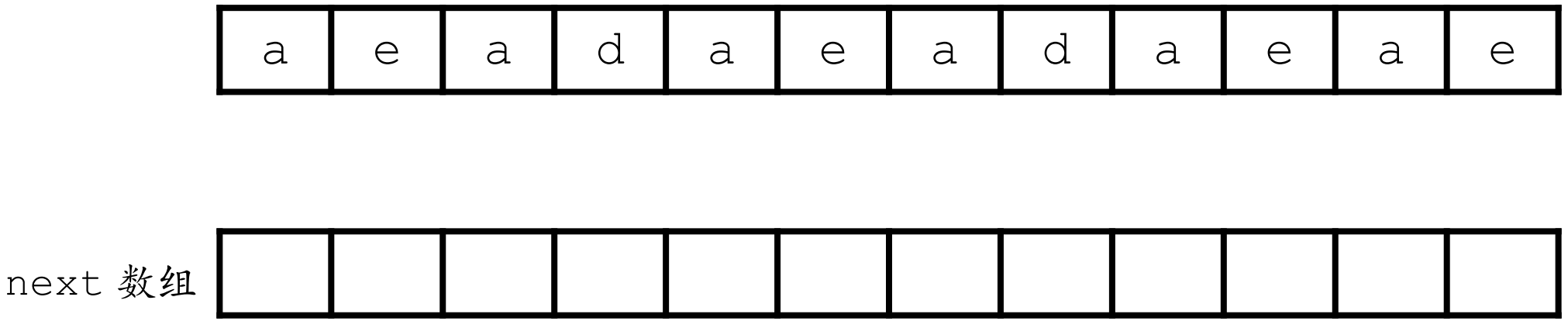
KMP 算法-next数组



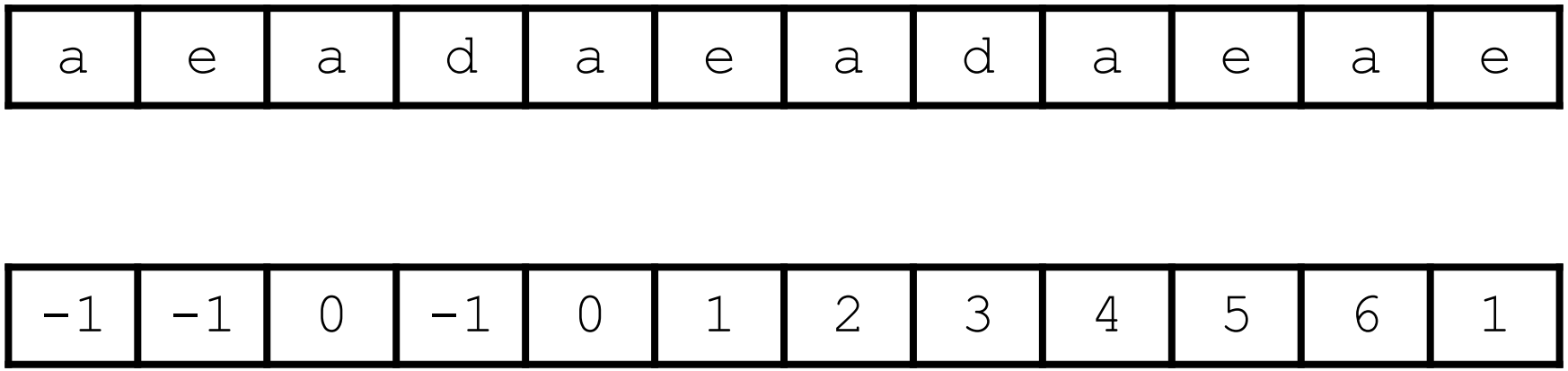
KMP 算法-next数组

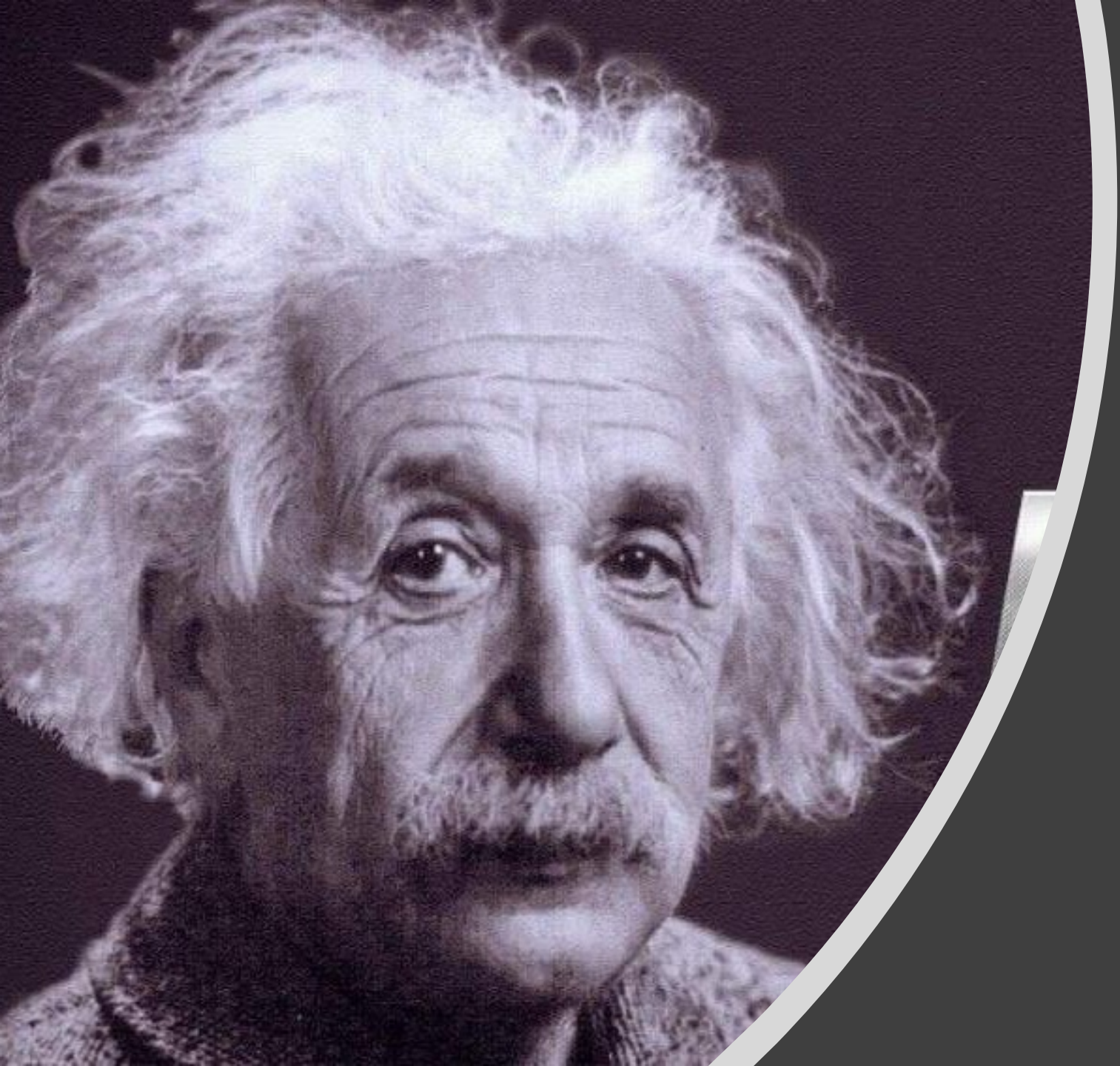


KMP 算法-next数组



KMP 算法-next数组





为什么
会出一样的题目？

《船说：算法与数据结构》
第15章-字符串匹配