

# 第三天

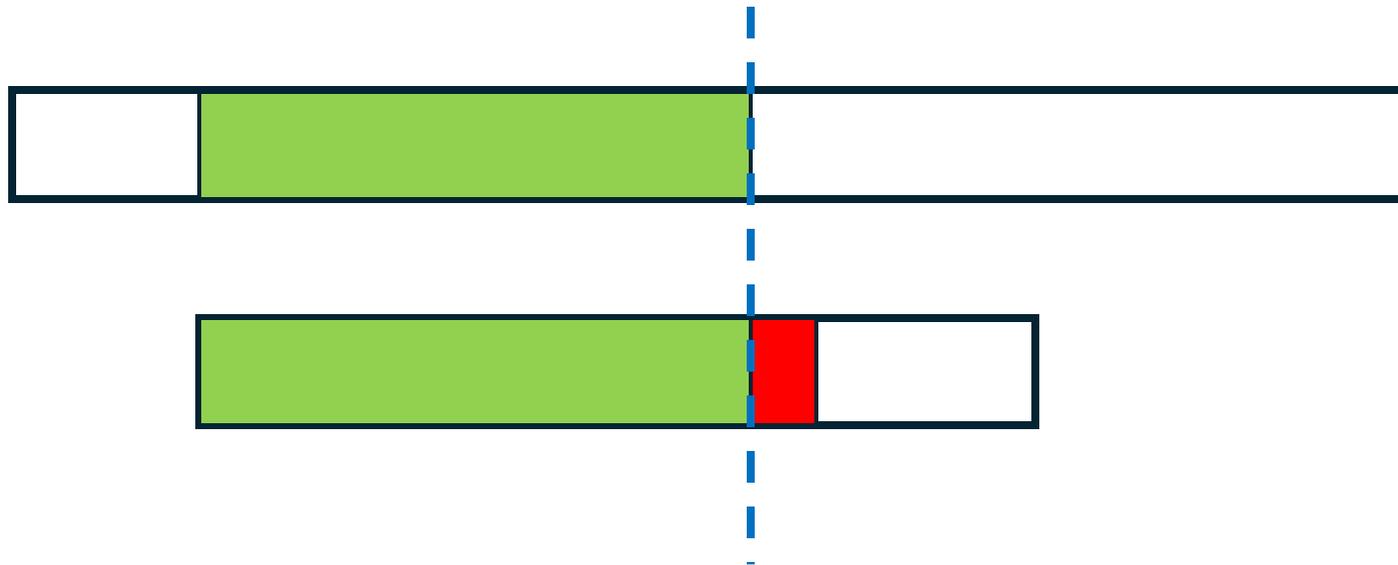
## KMP与Z函数

胡船长

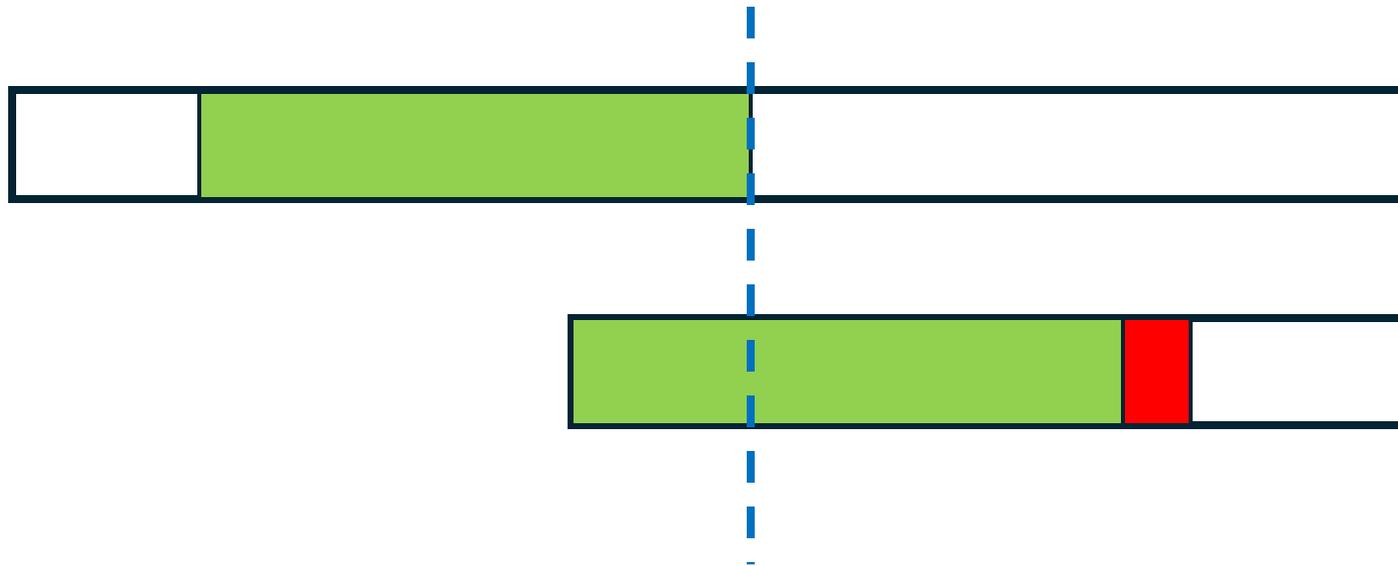
初航我带你，远航靠自己

# KMP算法

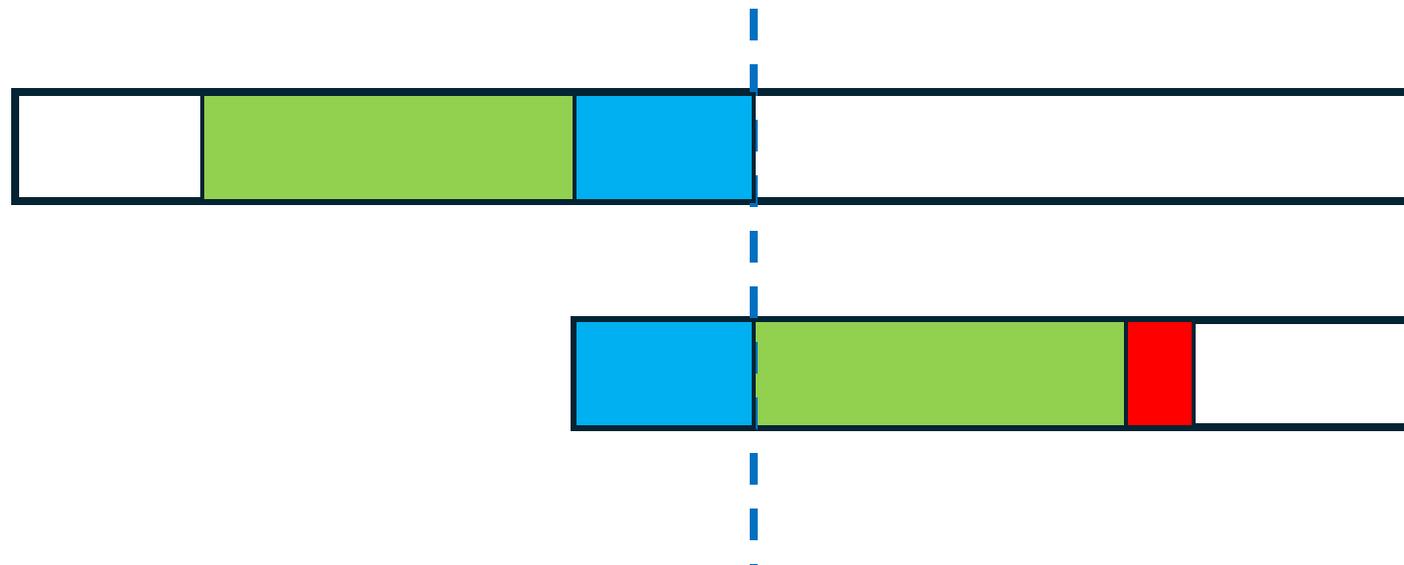
# KMP 算法



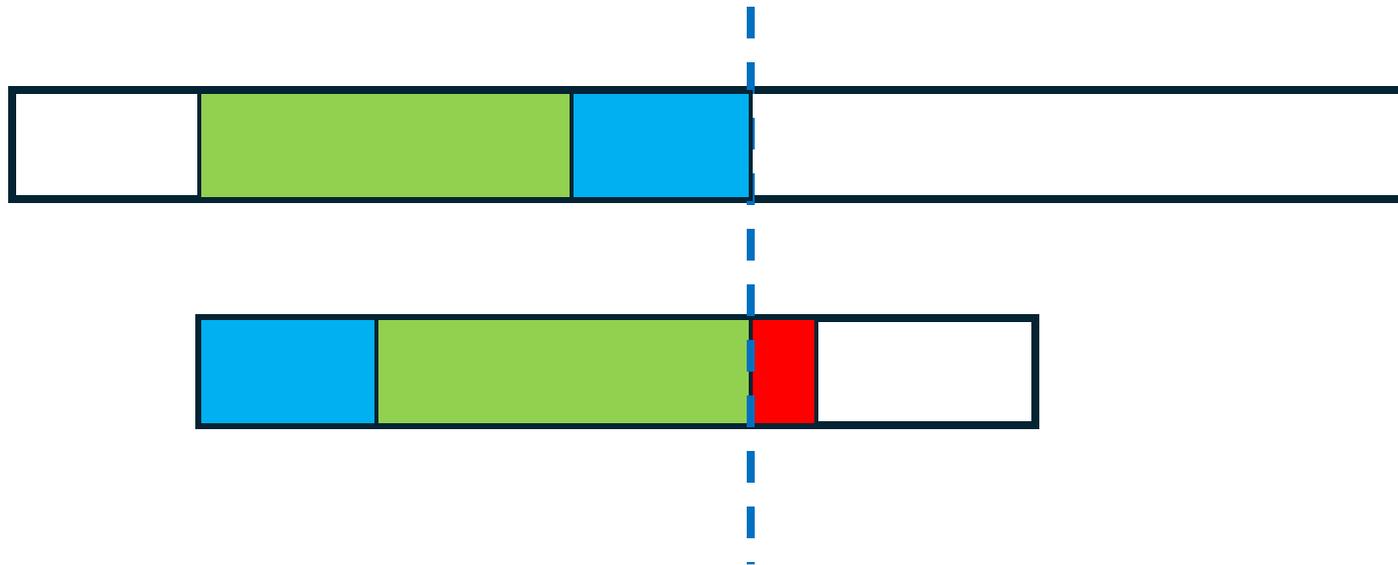
# KMP 算法



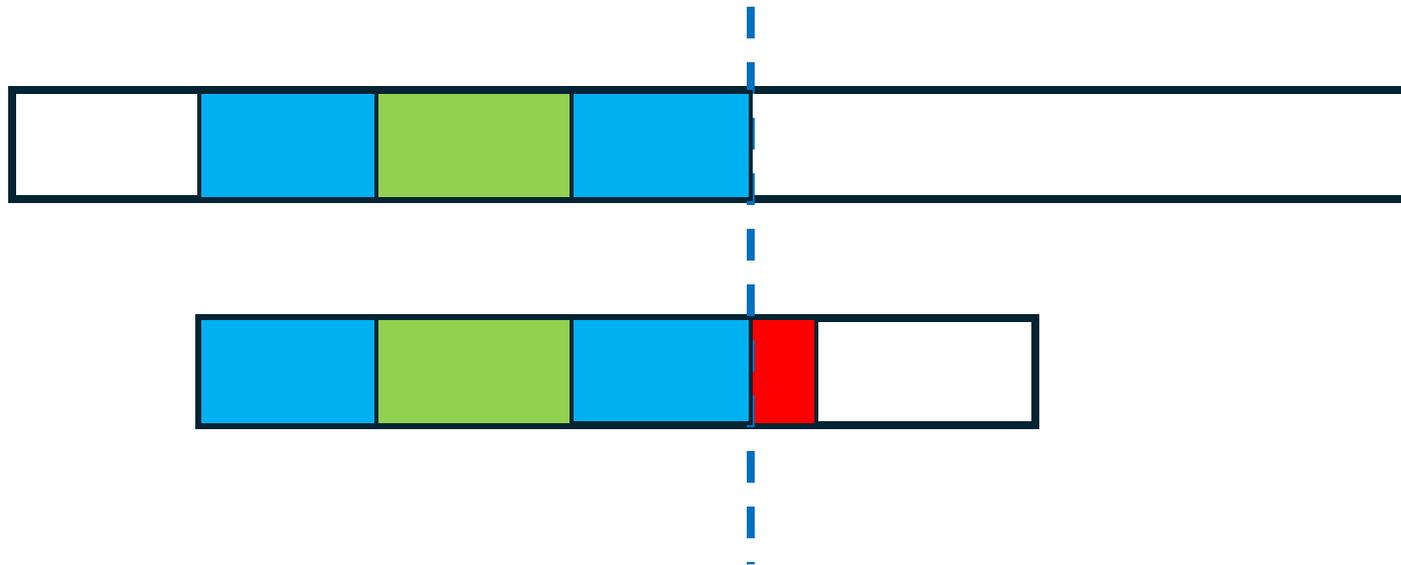
# KMP 算法



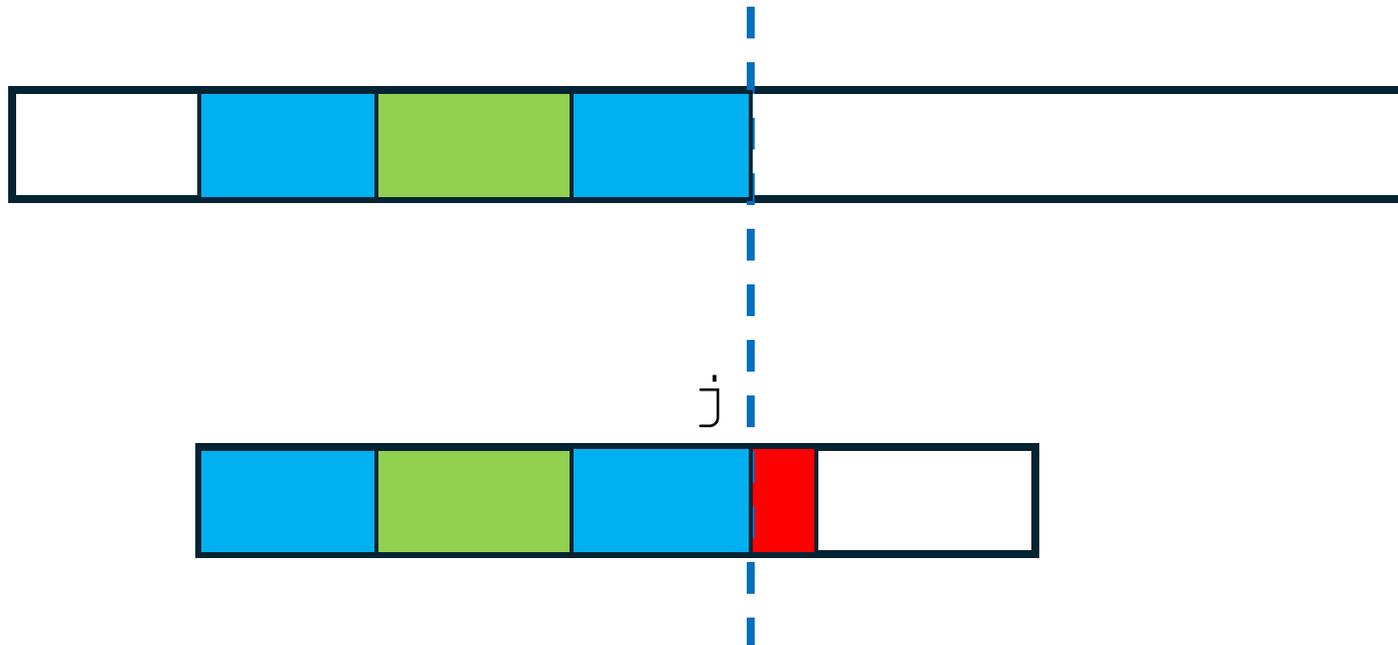
# KMP 算法



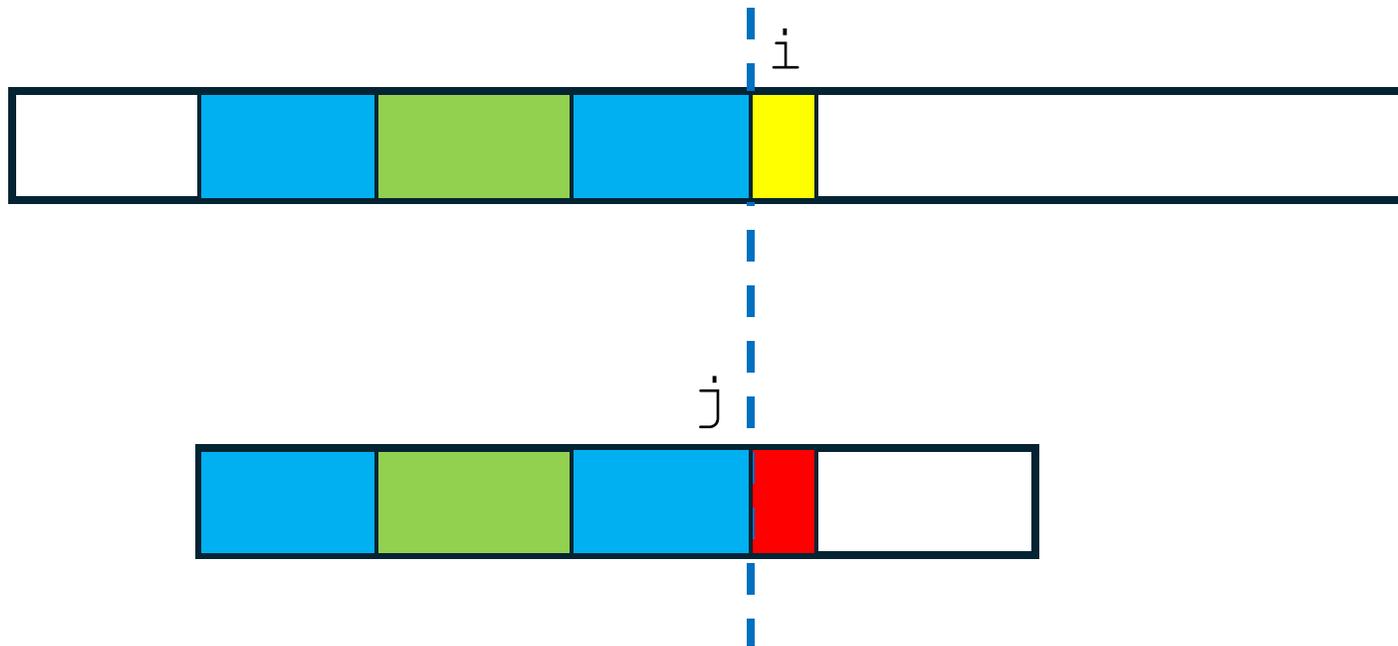
# KMP 算法



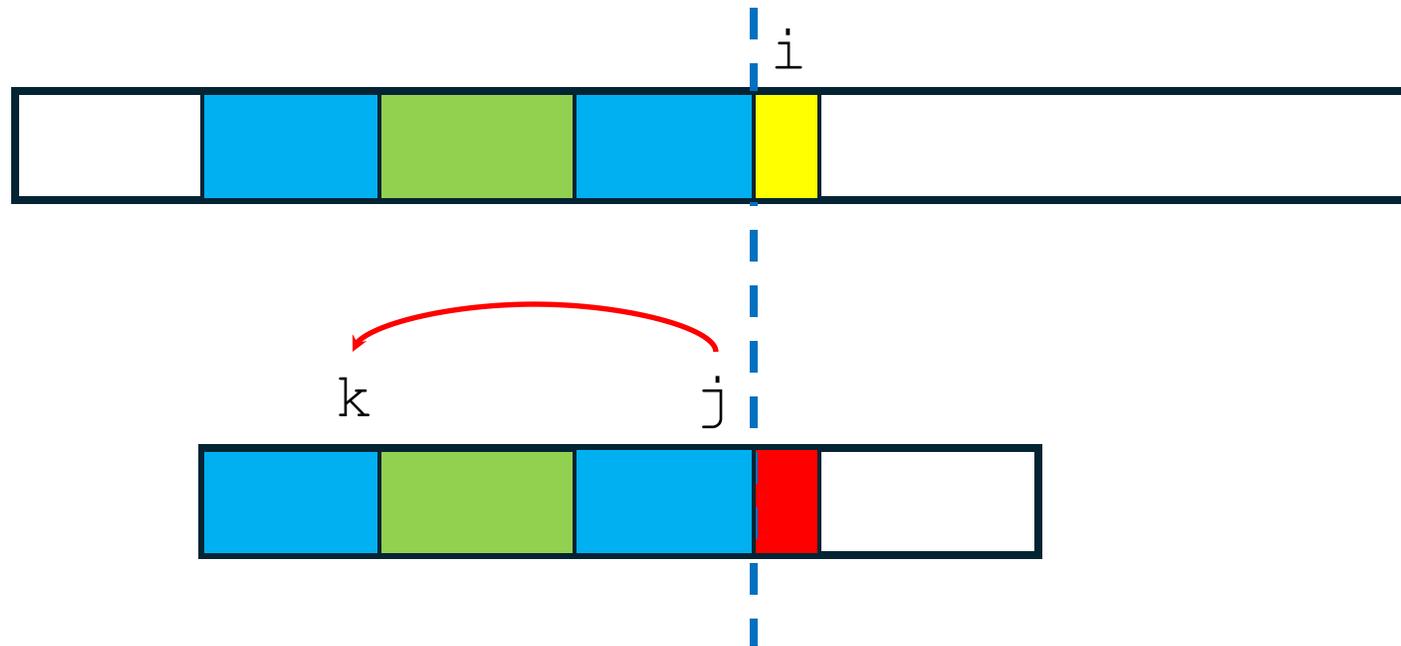
# KMP 算法



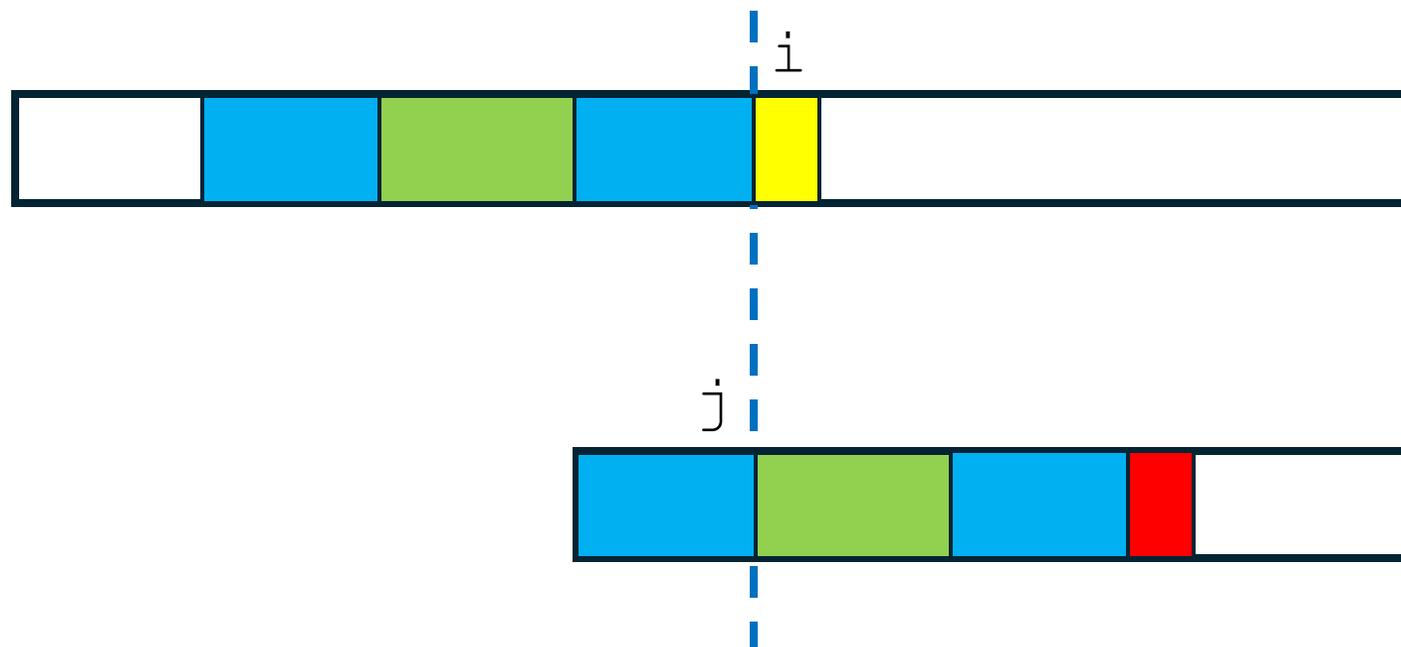
# KMP 算法



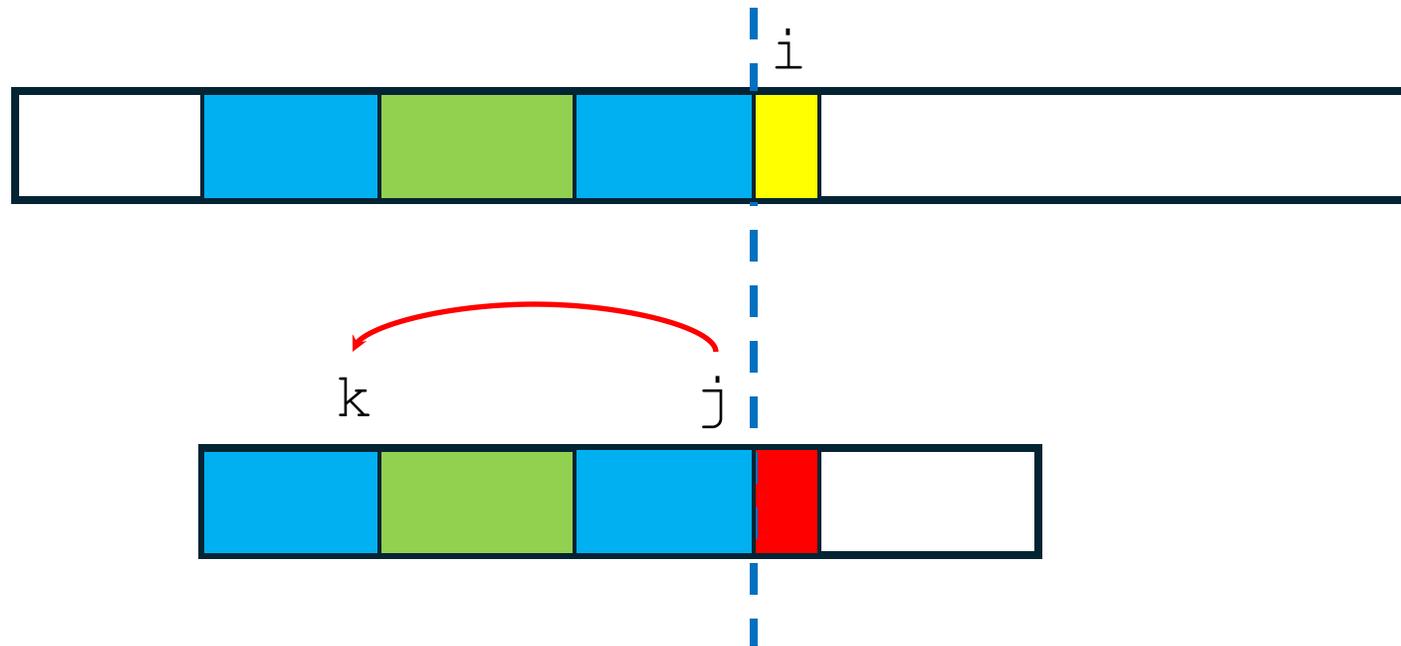
# KMP 算法



# KMP 算法

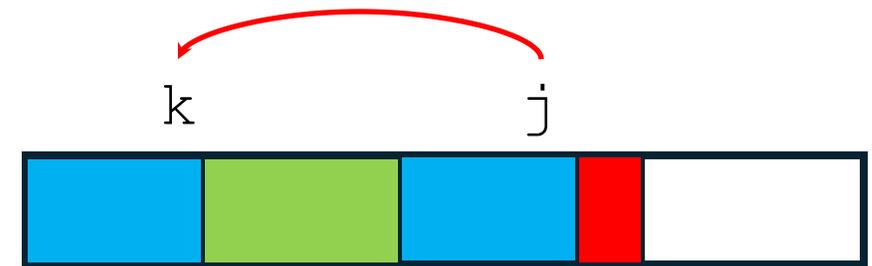


# KMP 算法



# KMP 算法-next数组

1. 最长前缀信息存储在 next 数组中,  $\text{next}[j] = k$
2.  $\text{next}[j]$  的值可以通过之前的 next 值得求得



# KMP 算法-next数组

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



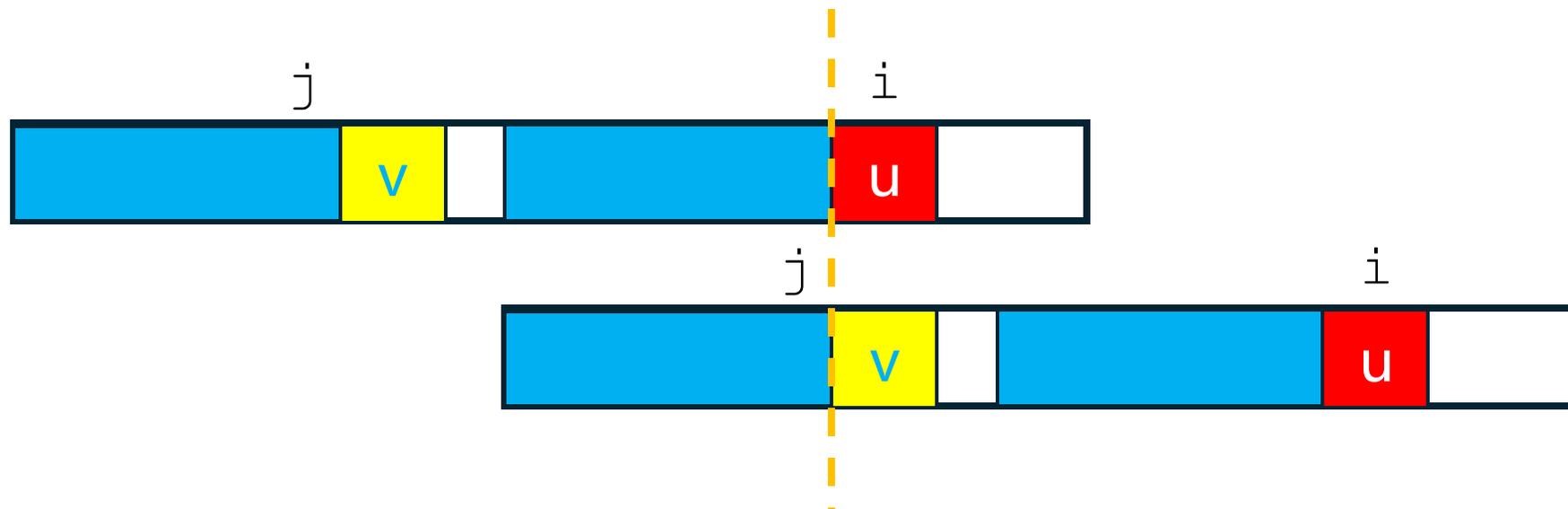
# KMP 算法-next数组

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



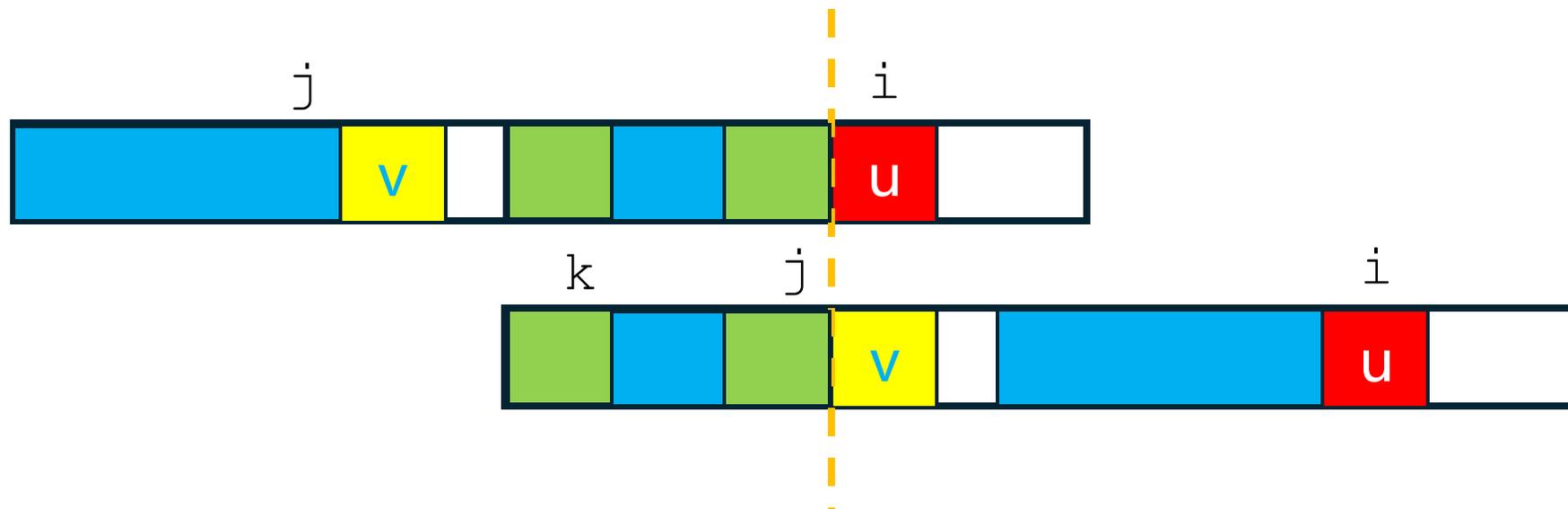
# KMP 算法-next数组

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



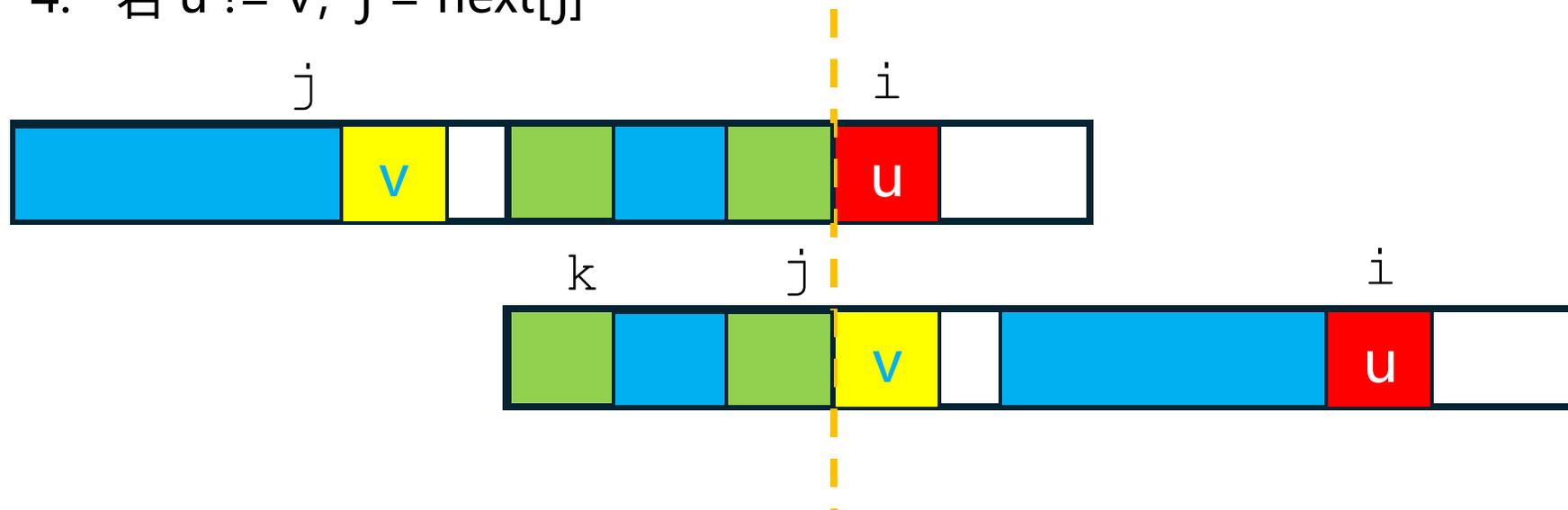
# KMP 算法-next数组

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



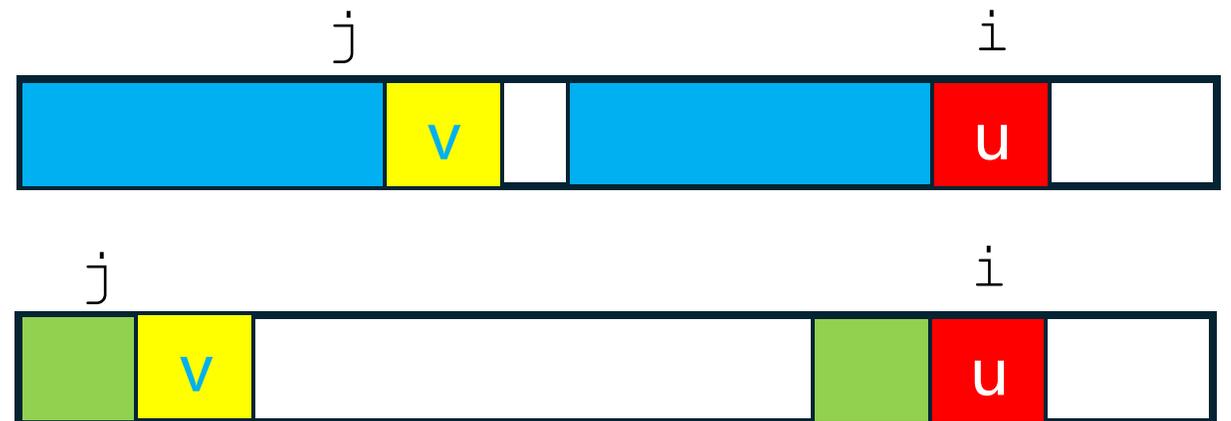
# KMP 算法-next数组

1. 最长前缀信息存储在 next 数组中,  $\text{next}[j] = k$
2.  $\text{next}[i]$  的值可以通过之前的 next 值得求得
3. 若  $u == v$ ,  $\text{next}[i] = j+1$
4. 若  $u \neq v$ ,  $j = \text{next}[j]$



# KMP 算法-next数组

1. 最长前缀信息存储在 next 数组中,  $\text{next}[j] = k$
2.  $\text{next}[i]$  的值可以通过之前的 next 值得求得
3. 若  $u == v$ ,  $\text{next}[i] = j+1$
4. 若  $u \neq v$ ,  $j = \text{next}[j]$



# KMP 算法-next数组

a	e	c	a	e	d
---	---	---	---	---	---

next 数组

--	--	--	--	--	--

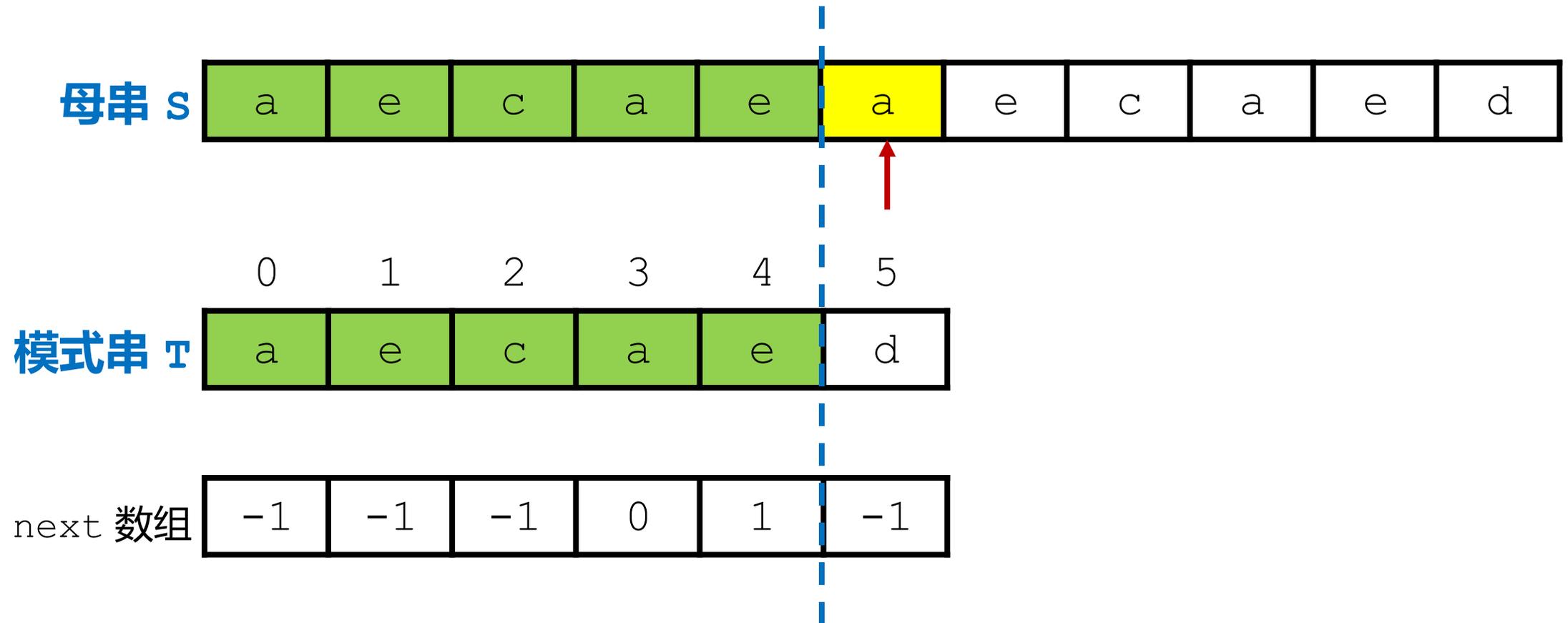
# KMP 算法-next数组

a	e	c	a	e	d
---	---	---	---	---	---

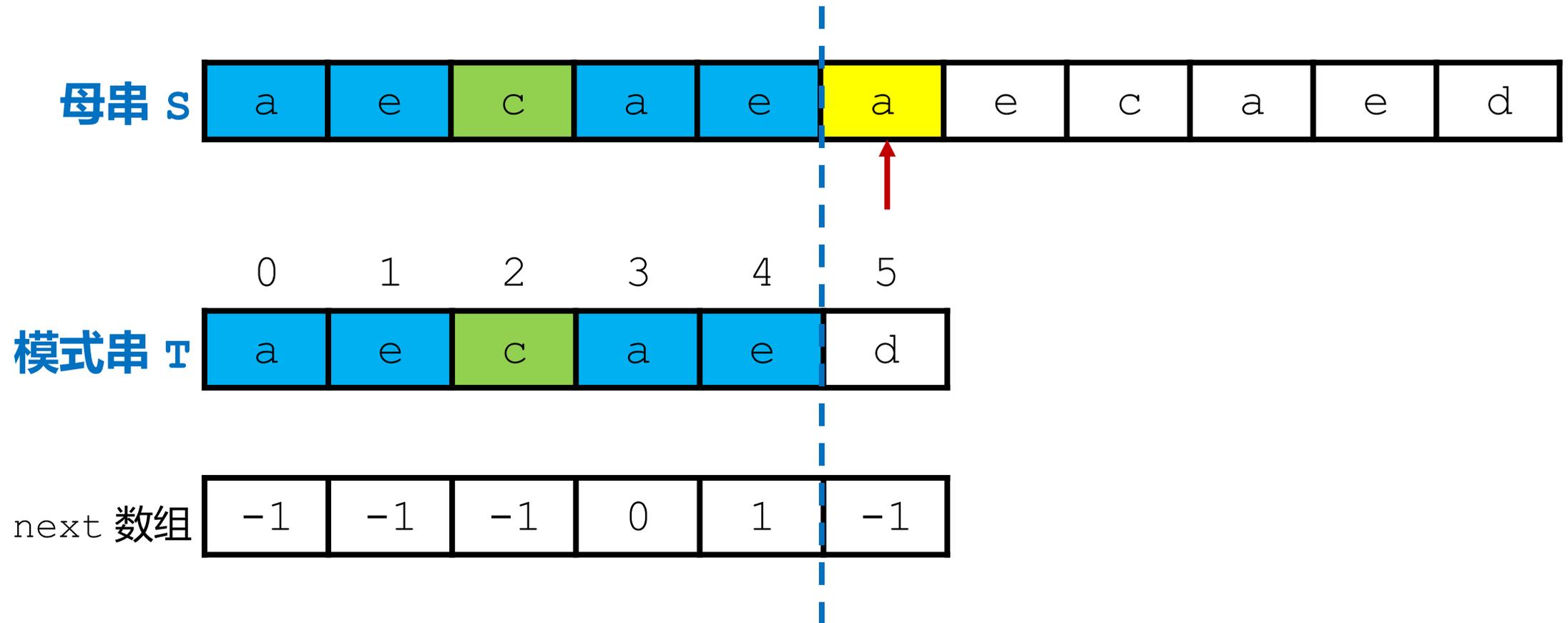
next 数组

-1	-1	-1	0	1	-1
----	----	----	---	---	----

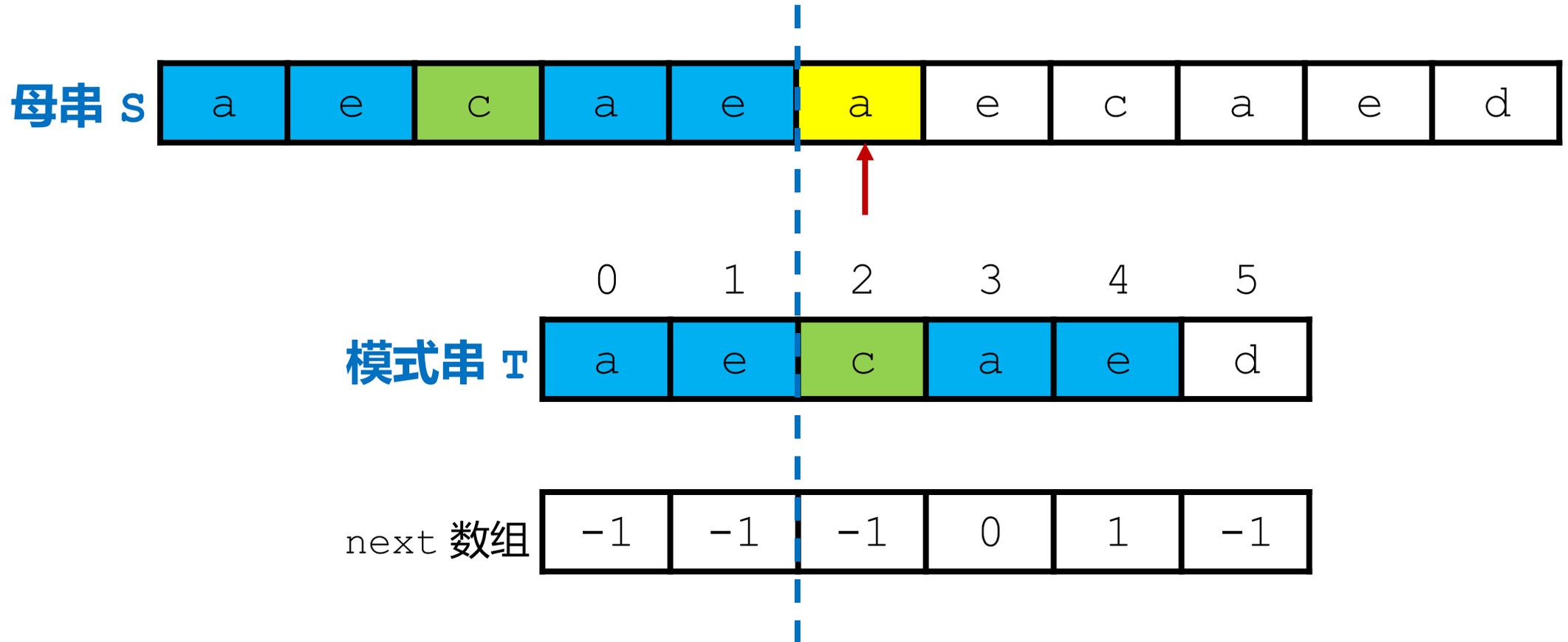
# KMP 算法-next数组



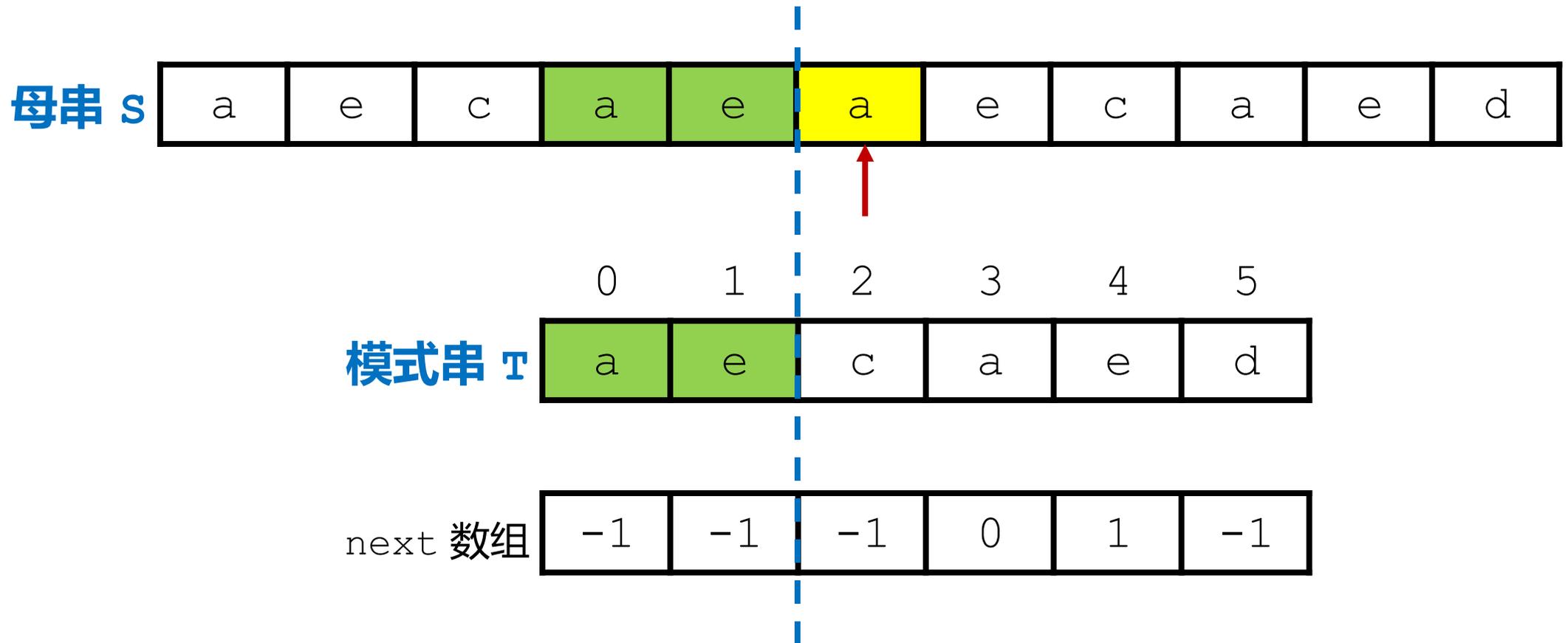
# KMP 算法-next数组



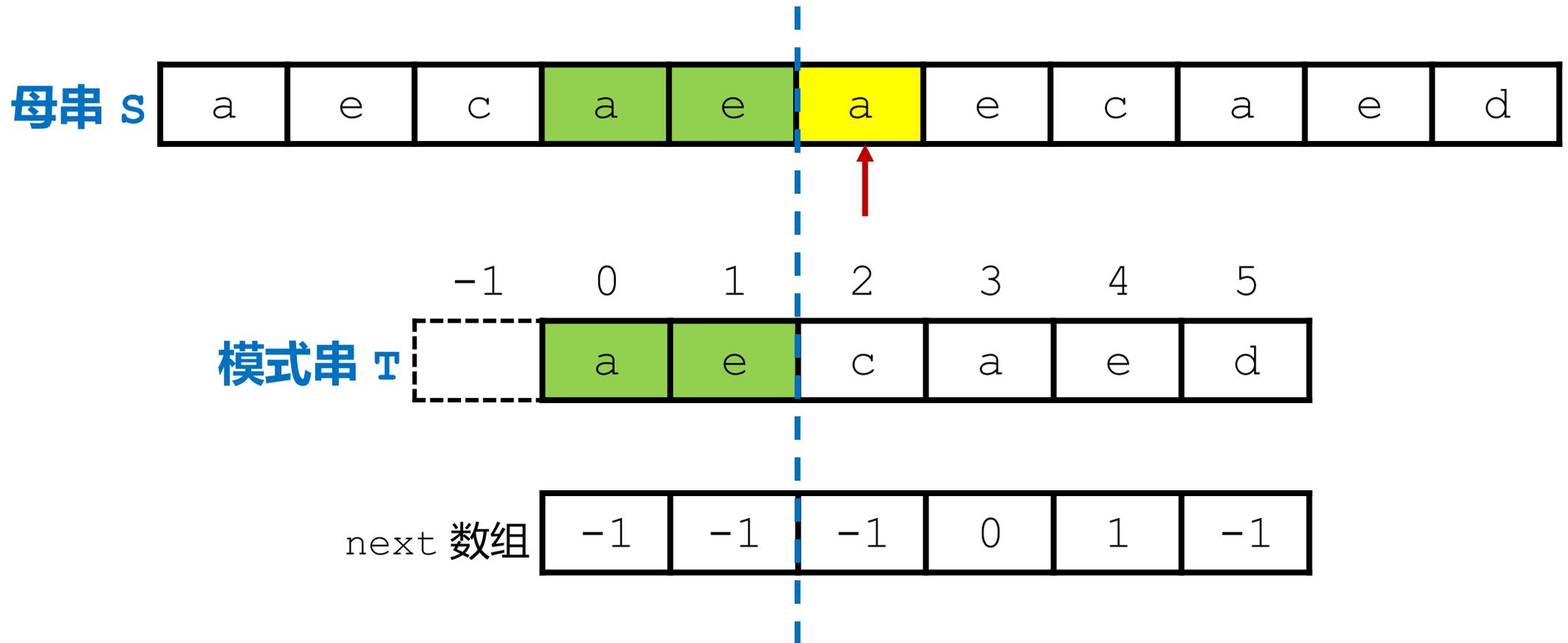
# KMP 算法-next数组



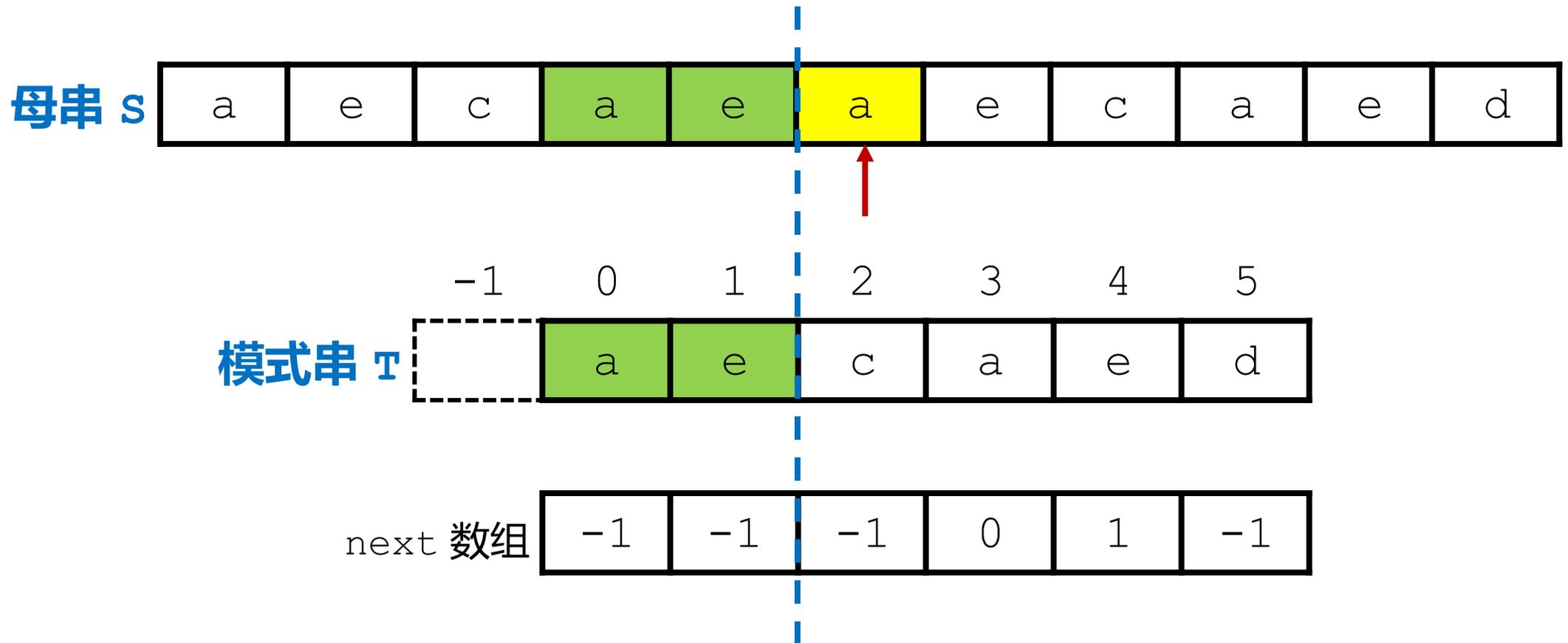
# KMP 算法-next数组



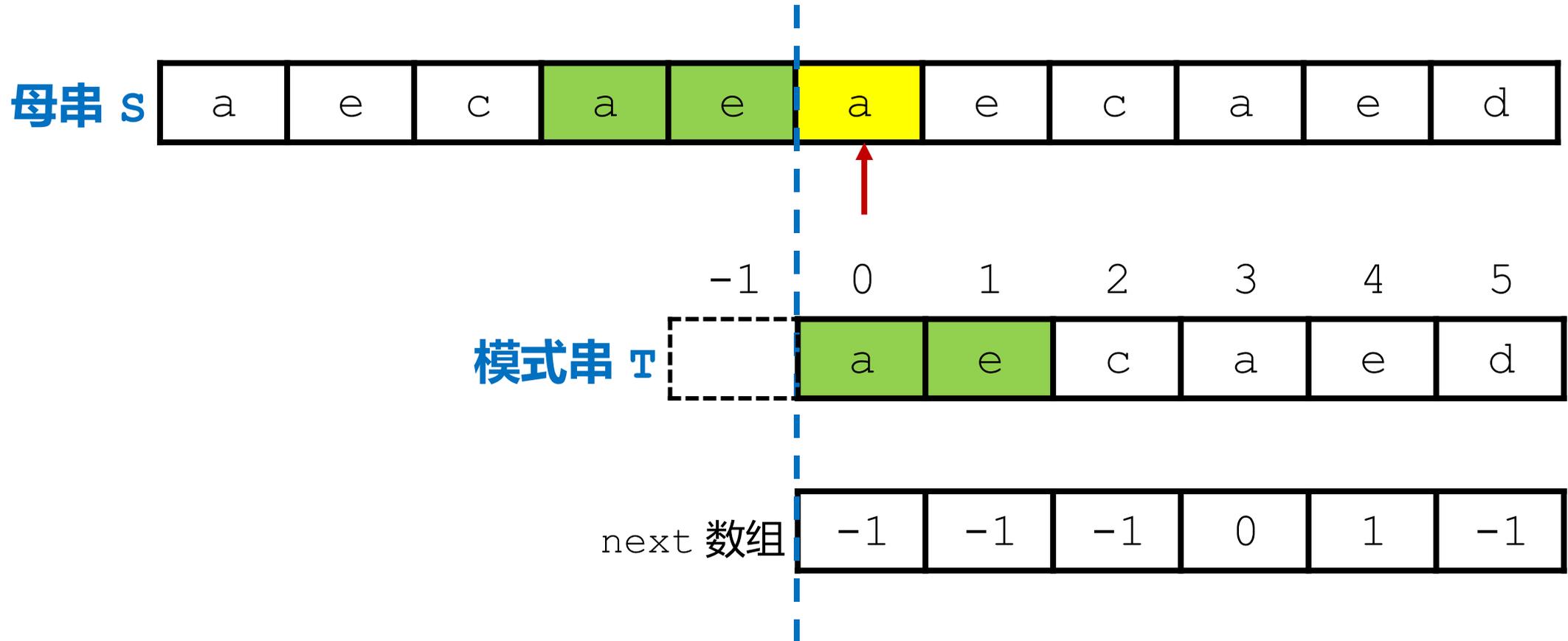
# KMP 算法-next数组



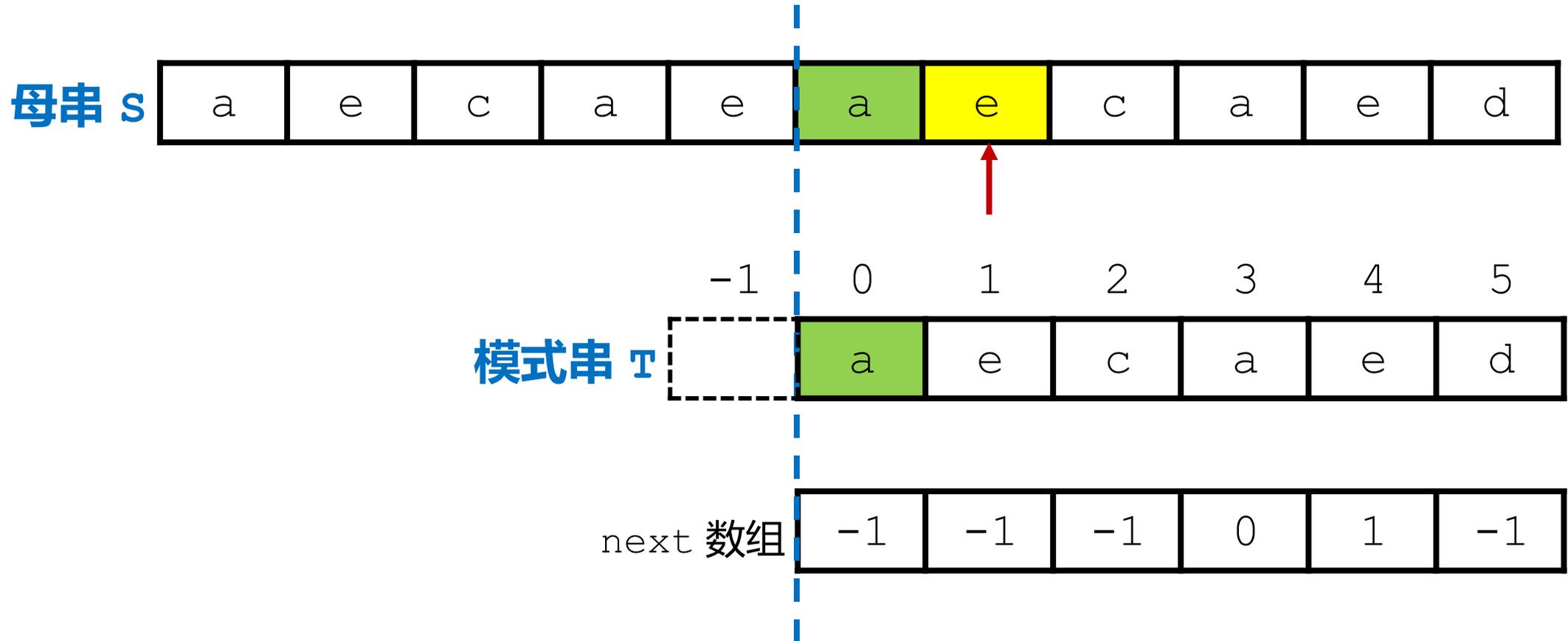
# KMP 算法-next数组



# KMP 算法-next数组



# KMP 算法-next数组



# KMP 算法-next数组

a	e	a	d	a	e	a	d	a	e	a	e
---	---	---	---	---	---	---	---	---	---	---	---

next 数组

--	--	--	--	--	--	--	--	--	--	--	--

# KMP 算法-next数组

a	e	a	d	a	e	a	d	a	e	a	e
---	---	---	---	---	---	---	---	---	---	---	---

next 数组

-1	-1	0	-1	0	1	2	3	4	5	6	1
----	----	---	----	---	---	---	---	---	---	---	---

# Z函数 (扩展KMP)

# 什么是 Z 函数

字符串与其每一个后缀的最长公共前缀 (LCP) 长度

S		a	b	a	c	a	b
Z		6	0	1	0	2	0

# 如何求 Z 函数

字符串与其每一个后缀的最长公共前缀 (LCP) 长度

S | a b a c a b  
Z | 6  
↑  
等于字符串长度

# 如何求 Z 函數

字符串与其每一个后缀的最长公共前缀 (LCP) 长度

S		a	b	a	c	a	b
Z		6					

↑

其他位置暴力求解

# 如何求 Z 函数

字符串与其每一个后缀的最长公共前缀 (LCP) 长度

S		a	b	a	c	a	b
Z		6	0				



其他位置暴力求解

# 如何求 Z 函数

字符串与其每一个后缀的最长公共前缀 (LCP) 长度

S		a	b	a	c	a	b
Z		6	0	1			



其他位置暴力求解

# 如何求 Z 函数

字符串与其每一个后缀的最长公共前缀 (LCP) 长度

S		a	b	a	c	a	b
Z		6	0	1	0		



其他位置暴力求解

# 如何求 Z 函数

字符串与其每一个后缀的最长公共前缀 (LCP) 长度

S		a	b	a	c	a	b
Z		6	0	1	0	2	



其他位置暴力求解

# 如何求 Z 函数

字符串与其每一个后缀的最长公共前缀 (LCP) 长度

S		a	b	a	c	a	b
Z		6	0	1	0	2	0

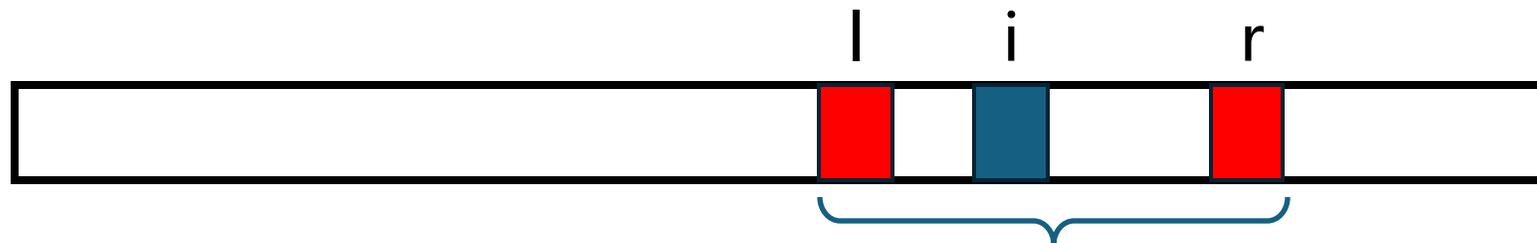
# 对暴力求解进行优化

字符串与其每一个后缀的最长公共前缀 (LCP) 长度



# 对暴力求解进行优化

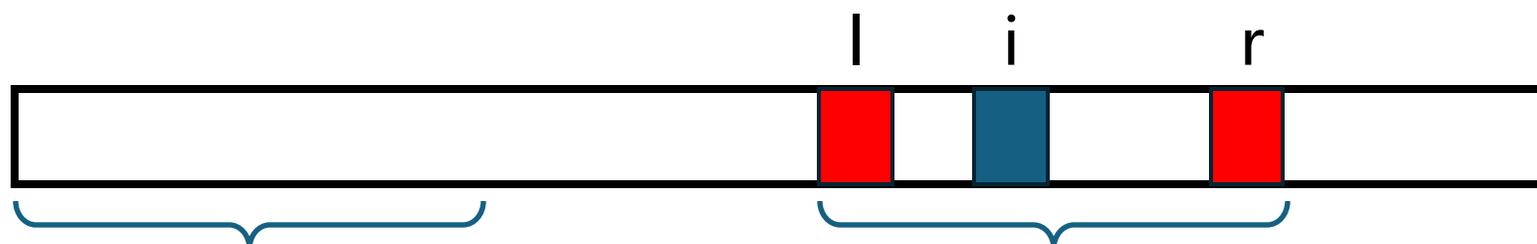
字符串与其每一个后缀的最长公共前缀 (LCP) 长度



从  $l$  位置进行前缀匹配的右端点  $r$  最远

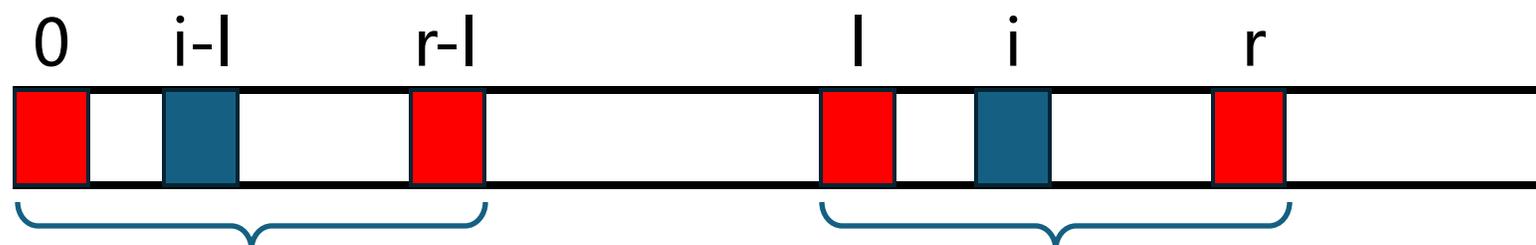
# 对暴力求解进行优化

字符串与其每一个后缀的最长公共前缀 (LCP) 长度



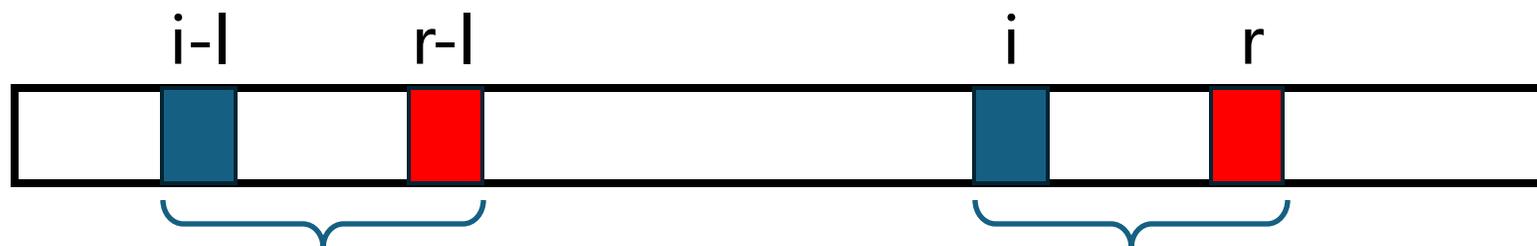
# 对暴力求解进行优化

字符串与其每一个后缀的最长公共前缀 (LCP) 长度



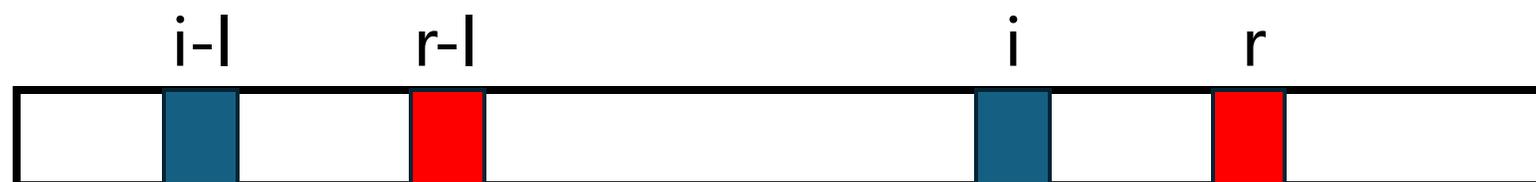
# 对暴力求解进行优化

字符串与其每一个后缀的最长公共前缀 (LCP) 长度



# 对暴力求解进行优化

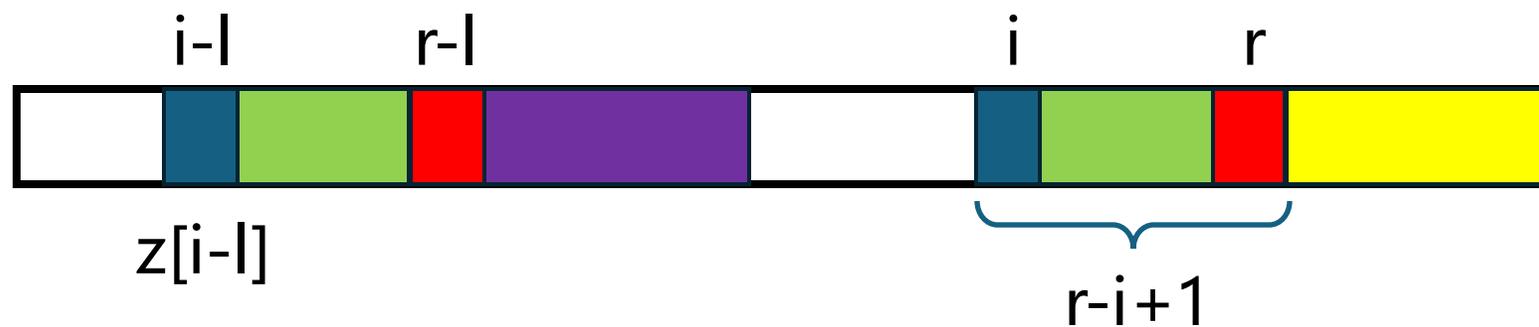
字符串与其每一个后缀的最长公共前缀 (LCP) 长度



$z[i]$  最小值为多少?

# 对暴力求解进行优化

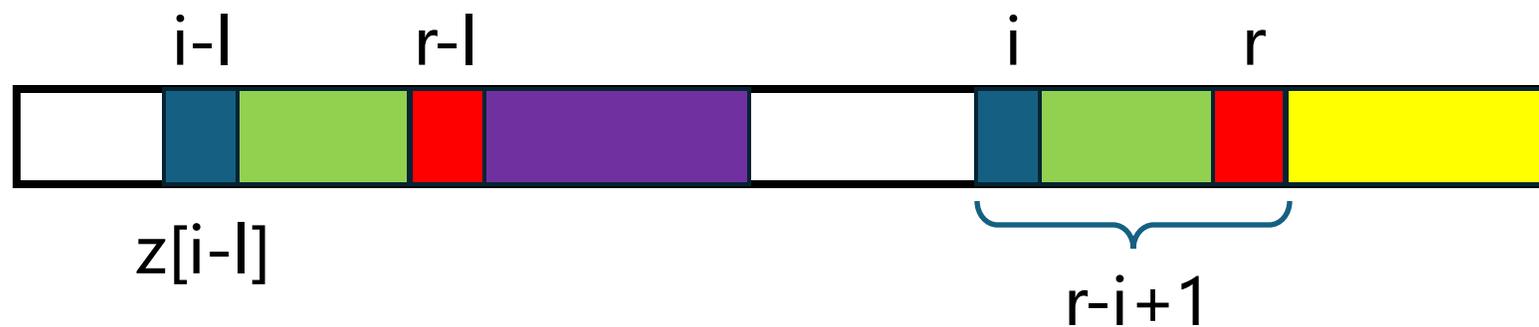
字符串与其每一个后缀的最长公共前缀 (LCP) 长度



$z[i]$  最小值为多少?

# 对暴力求解进行优化

字符串与其每一个后缀的最长公共前缀 (LCP) 长度



$z[i]$  不会小于  $\min(z[i-1], r-i+1)$  中的较小值

# 题目列表：KMP与Z函数

1. LC796. 旋转字符串
2. LC214. 最短回文串
3. LC2800. 包含三个字符串的最短字符串
4. LC1764. 通过连接另一个数组的子数组得到一个数组
5. P2375 [NOI2014] 动物园
6. P3193 [HNOI2008] GT考试
7. P10581 [蓝桥杯 2024 国 A] 重复的串
8. LC28. 找出字符串中第一个匹配项的下标
9. LC2223. 构造字符串的总得分和
10. LC3474. 字典序最小的生成字符串