求最大重复子串

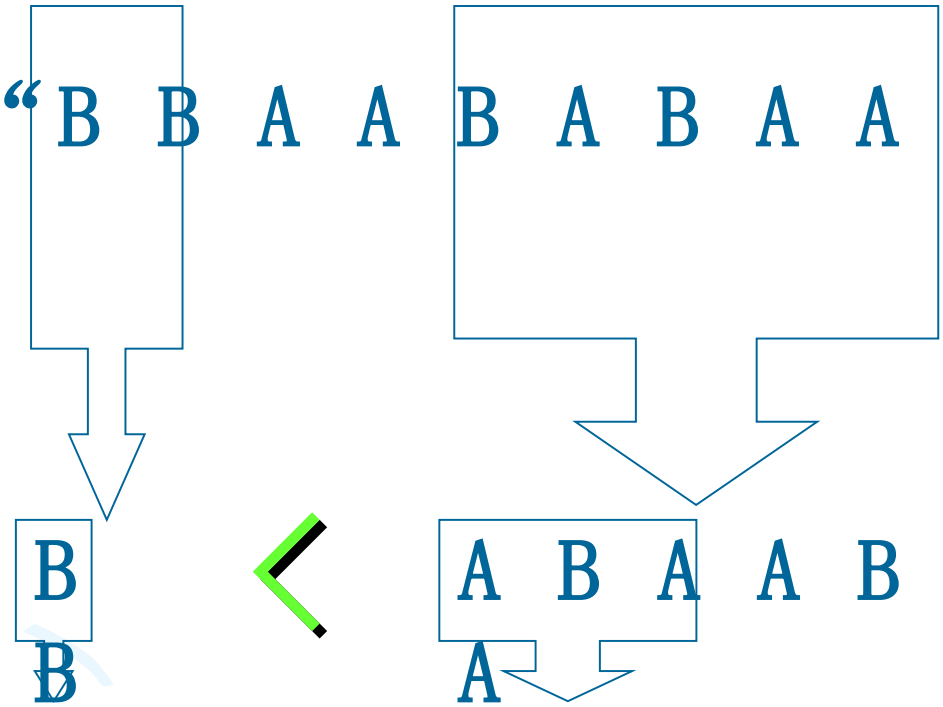
江苏金陵中学　林希德

题目

字符串W由大写字母组成，W中包含一些连续出现**两次**的**相同**子串，称之为重复子串。**重复子串**的大小决定于循环节的长度。

举例

W = “B B A A B A B A A B A B B”



循环节长度为1　　循环节长度为3

题目

字符串W由大写字母组成，W中包含一些连续出现两次的相同子串，称之为重复子串。重复子串的大小决定于循环节的长度。请你求出最大重复子串的循环节长度。

数据规模

$n = |w| <= 100000$

$O(n^2)$ 

$O(n\lg_2^n)$ 

$O(n)$ 

两个辅助算法

后缀树 O(n)

KMP 模式匹配 O(n+m)

为方便表达，使用

W(u，v)

表示开始于位置 u 结束于位置 v 的 W 的子串

问题的转化

定义S是循环周期为L的**最优子串**，仅当S满足：

1、S中的字符以L为周期循环出现

$S_i = S_{i+L}$ （u <= i <= v - L）

2、|S| >= 2L，即S至少包括两个完整循环节。

3、S不能向左扩展，

即 u = 1 或者 W(u-1,v) 不满足条件1

4、S不能向右扩展，

即 v = n 或者 W(u,v+1) 不满足条件1

最大重复子串必然被某个最优子串包含！！

求出所有最优子串连同它们的周期

算法基本框架

1、找到S的一个完整循环节？

2、根据循环节将S分别向左、向右扩展到不

能扩展为止

3、判断扩展以后的S是否长度 >=2L

如果是，则认为找到了一个循环周期为L的最优子串S。

一、字符串分解

将 W 分解成 $W = U_1 + U_2 + U_3 + \cdots\cdots + U_m$ 的形式，其中 U_i 定义如下：

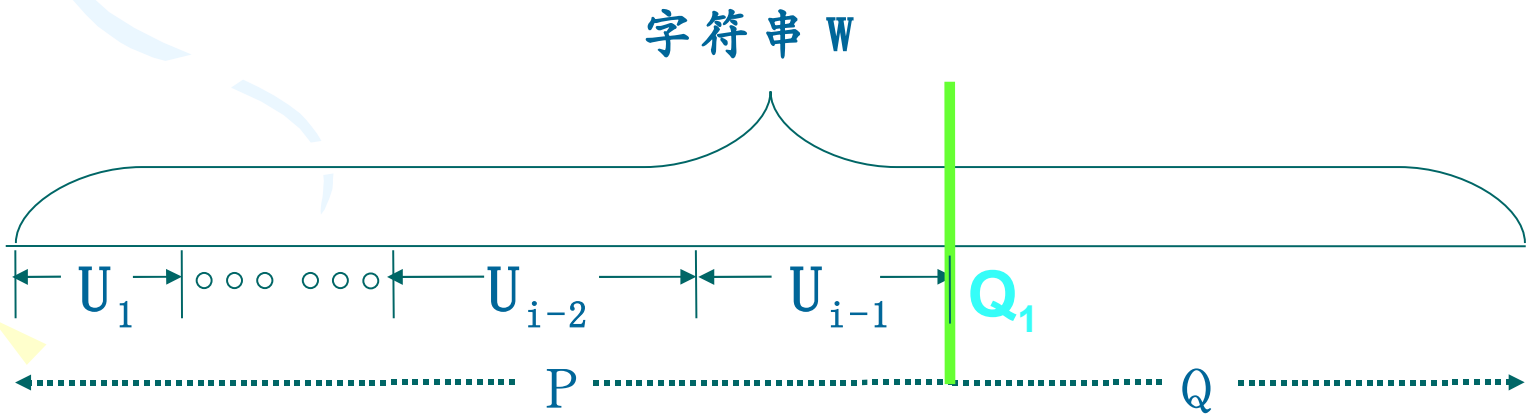
$P = U_1 + U_2 + \cdots\cdots + U_{i-1}$

如果字母 Q_1 从未在 P 中出现过，

那么 $U_i = Q_1$

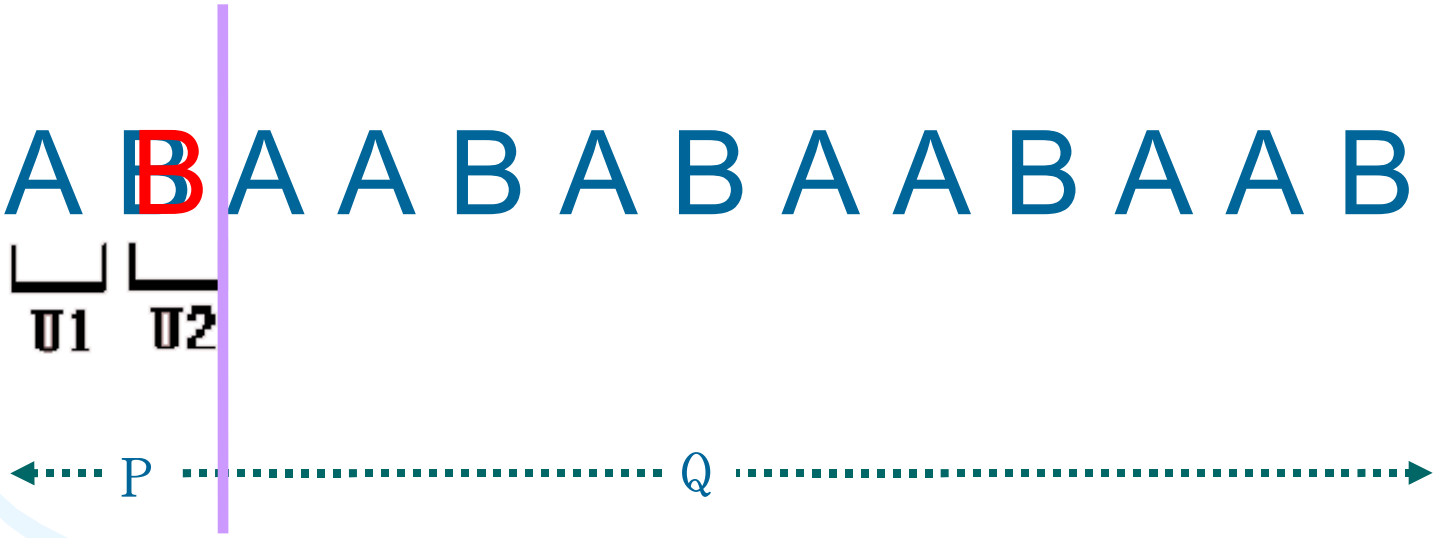
否则 U_i = P 中出现过的 Q 的最长前缀

只要字符串 x 的开始位置在 P 内，就认为 x 在 P 中出现过！



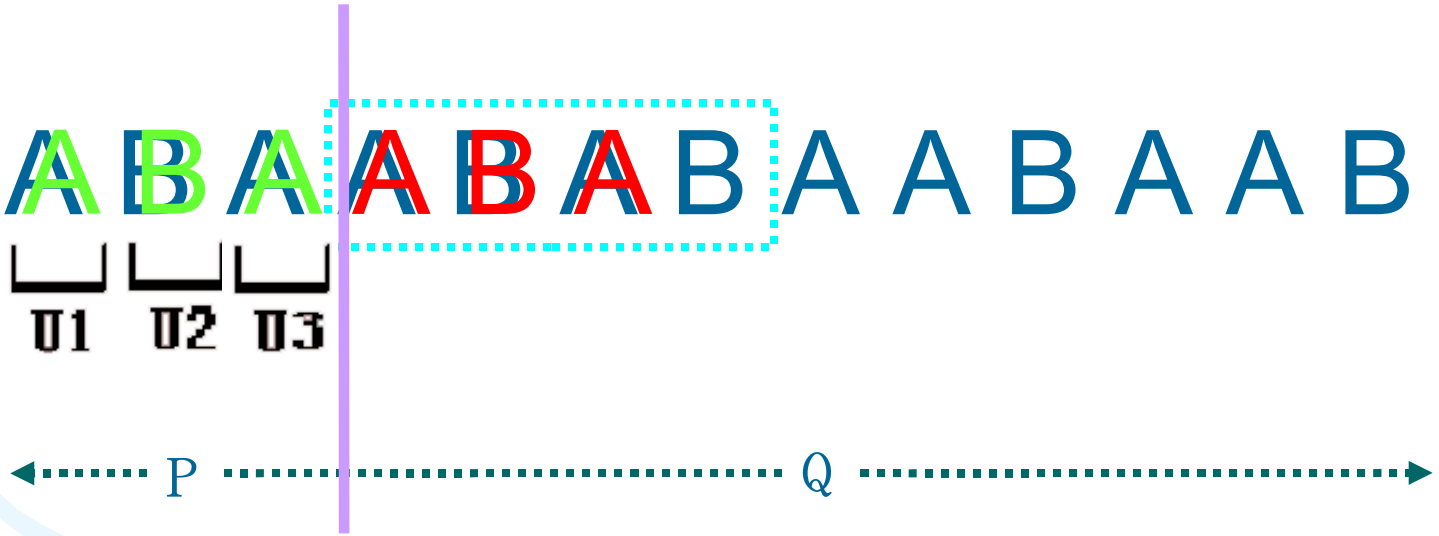
举例

举例

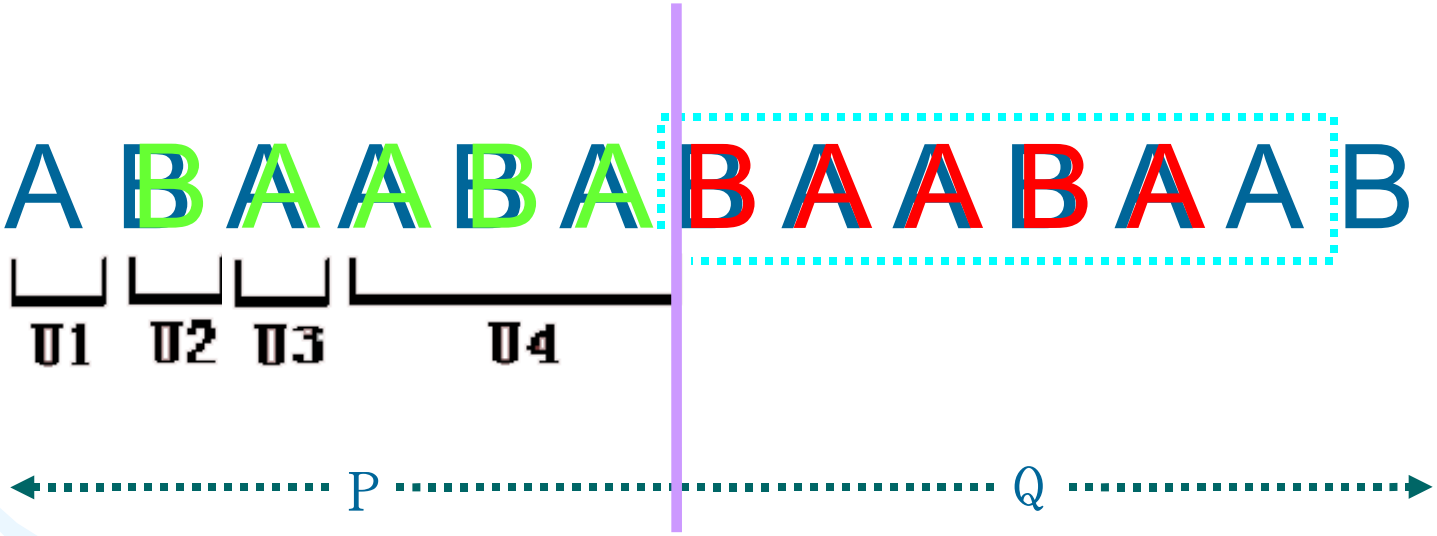


举例

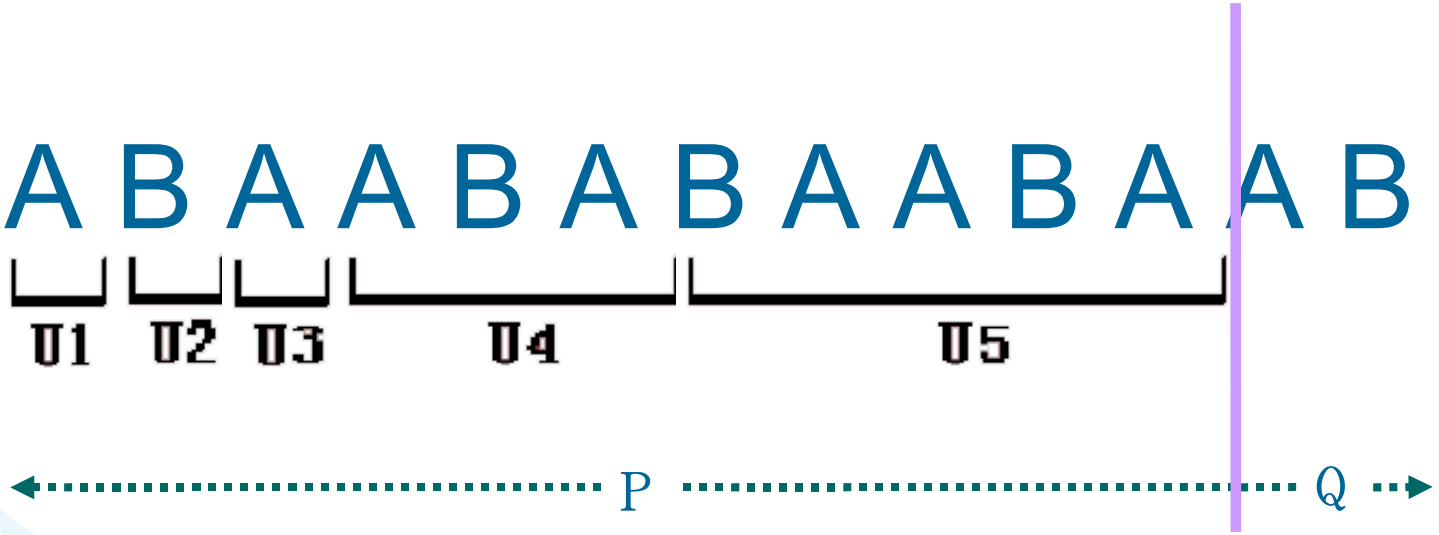
举例



举例



举例



举例

字符串分解过程借助“后缀树”算法实现

二、寻找完整循环节

怎样利用字符串分解的特殊定义找到最优子串 S 的一个完整循环节呢?

假设 S 的结束位置在固定片断 U_i 内

问题:

S 的开始位置在何处呢?

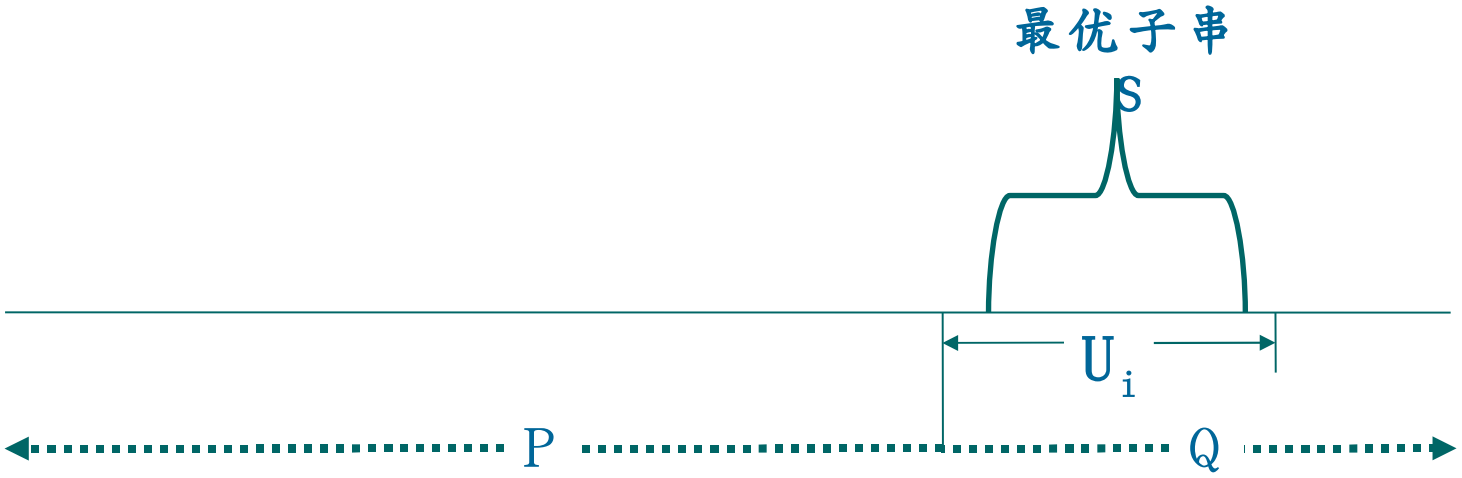
S 的循环节能有多长呢?

一定要记住:整数 i 是个已知常量!

解决方法:　　分类讨论。

S 的开始位置不能太迟

- S 的开始位置也在 U_i 内 .



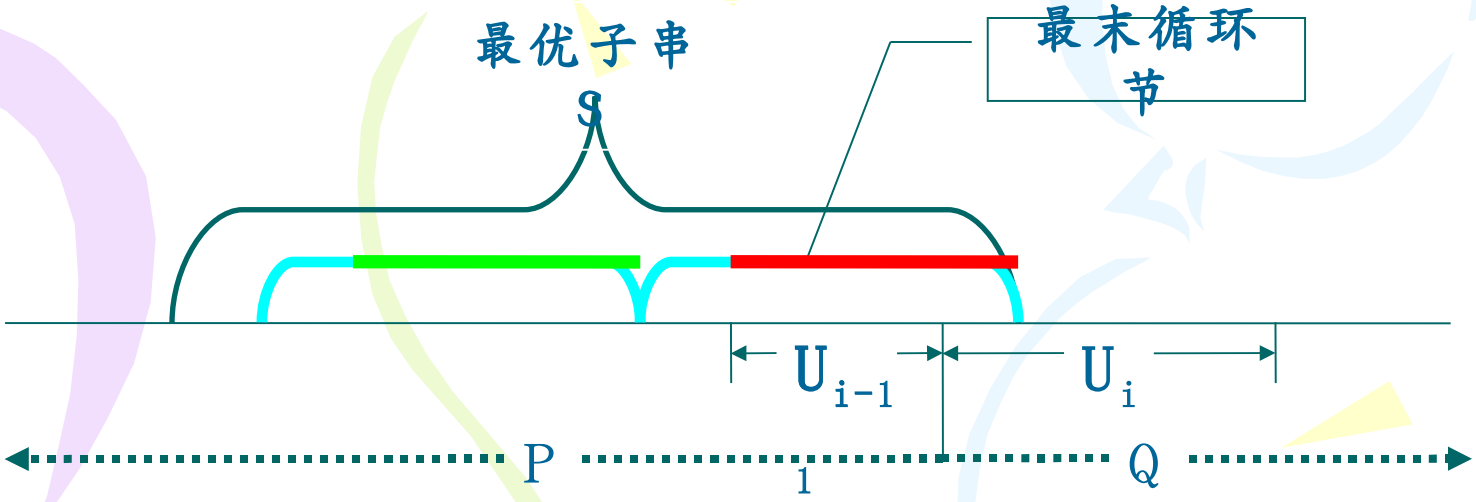
U_i 在 P 中某处出现过 ⇔ S 在 P 中某处出现过

为避免重复工作，此情况不予考虑！

这里用到了字符串分解的定义

S 的循环节不能太长

b. 最末循环节包含 U_{i-1}



红色和绿色线段标示了相同的子串

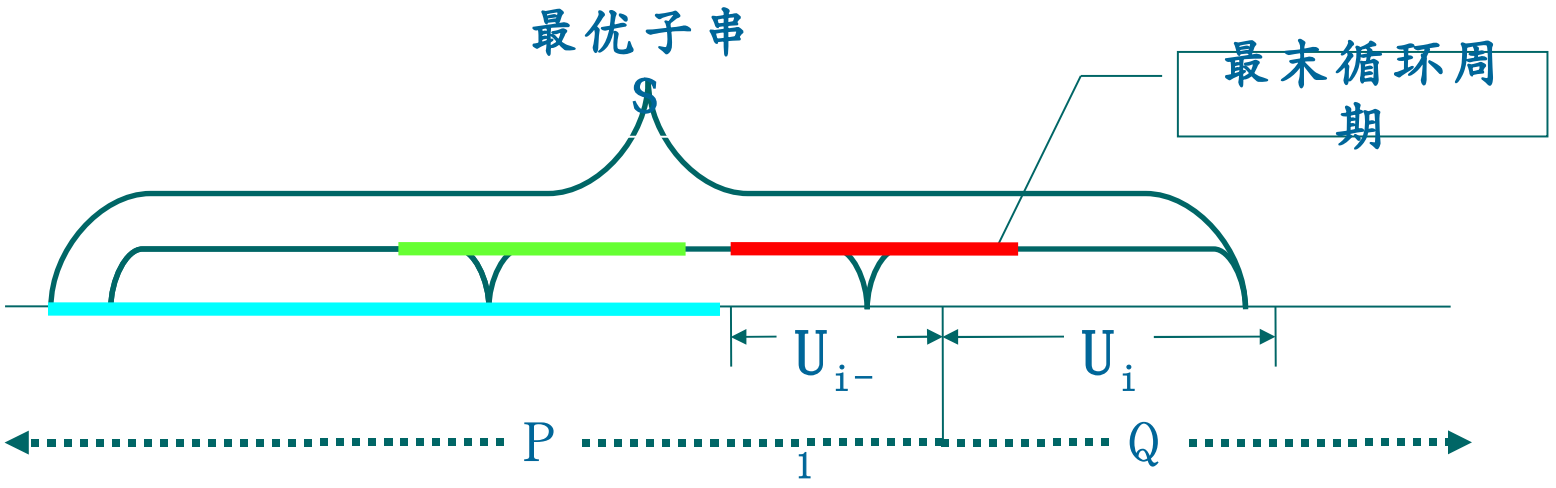
根据定义，$|U_{i-1}|$ >= 红色线段

矛盾，情况 b 不存在。

这里再次用到了字符串分解的定义

S 的开始位置不能太早

c. |S 位于 U_{i-1} 之前的子串| >= 循环周期 L



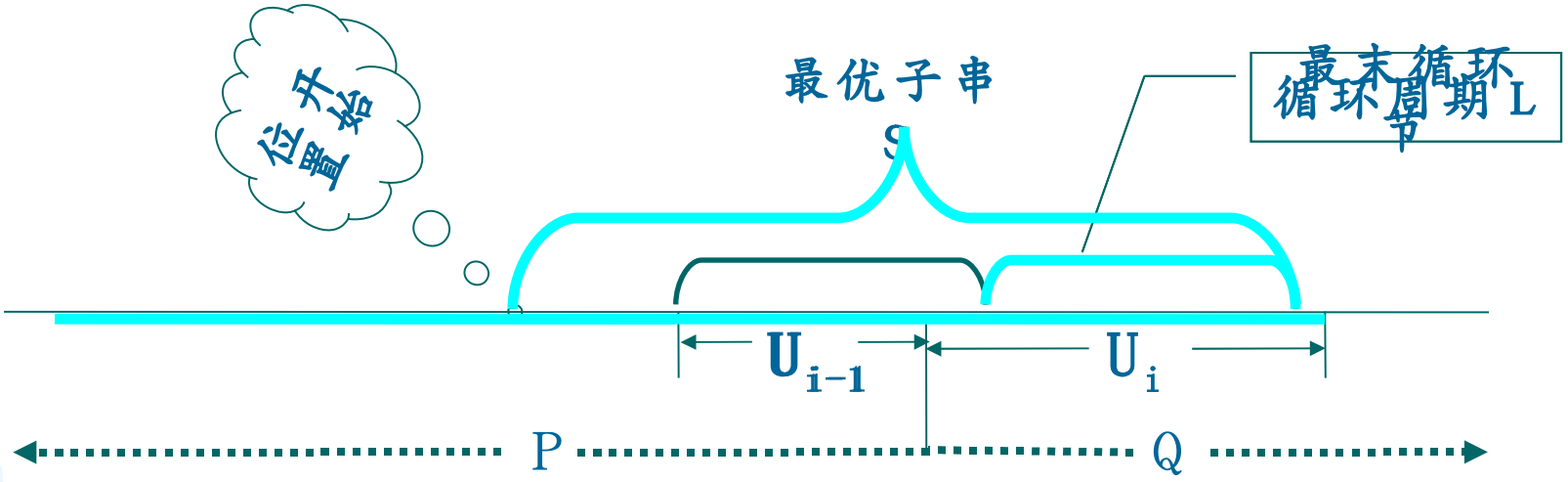
红色和绿色线段标示了相同子串

根据定义，$|U_{i-1}|$ >= 红色线段

矛盾，情况 c 也不存在。

这里又一次用到了字符串分解的定义

重要结论 1



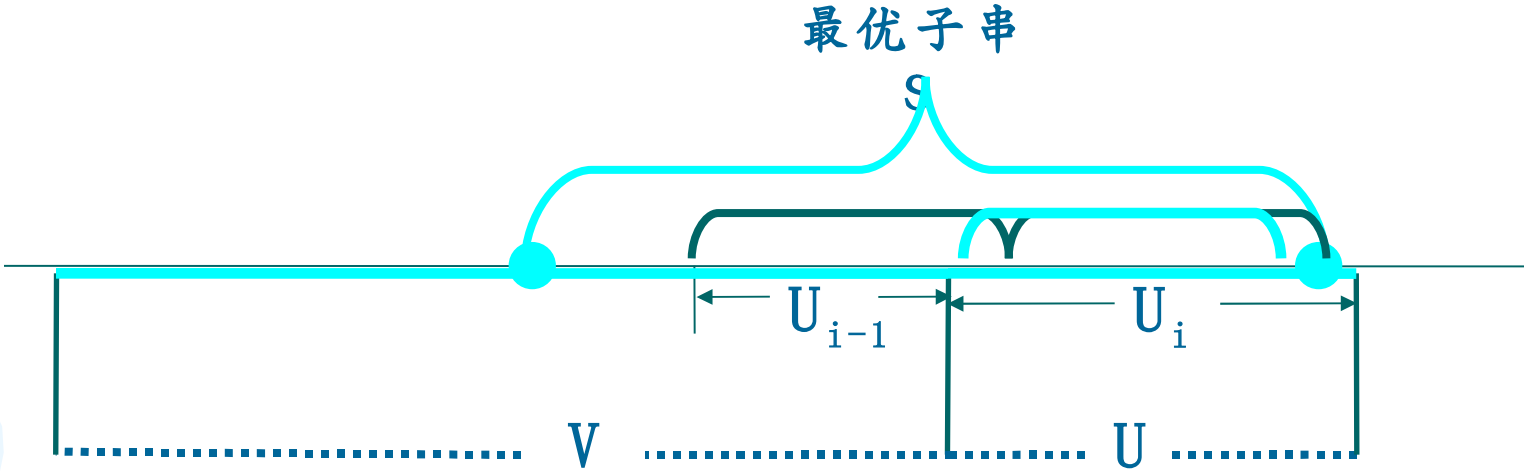
1. S 的开始位置早于 U_i 且最末循环节没有将 U_{i-1} 包含在内，故

$$L < |U_{i-1} + U_i|$$

2. |S 位于 U_{i-1} 之前的子串| < 循环周期 L，故

$$|S| < 2|U_{i-1} + U_i|$$

重要结论 1



进一步分类

因为 |S| >= 2 L，实际就是S的一个完整循环节

故下列两种情况S必居其一：

情况1. S在V中的长度 >= L

情况2. S在U中的长度 >= L

U(1,L)

找到循环节了！！！

这个结论很重要！！

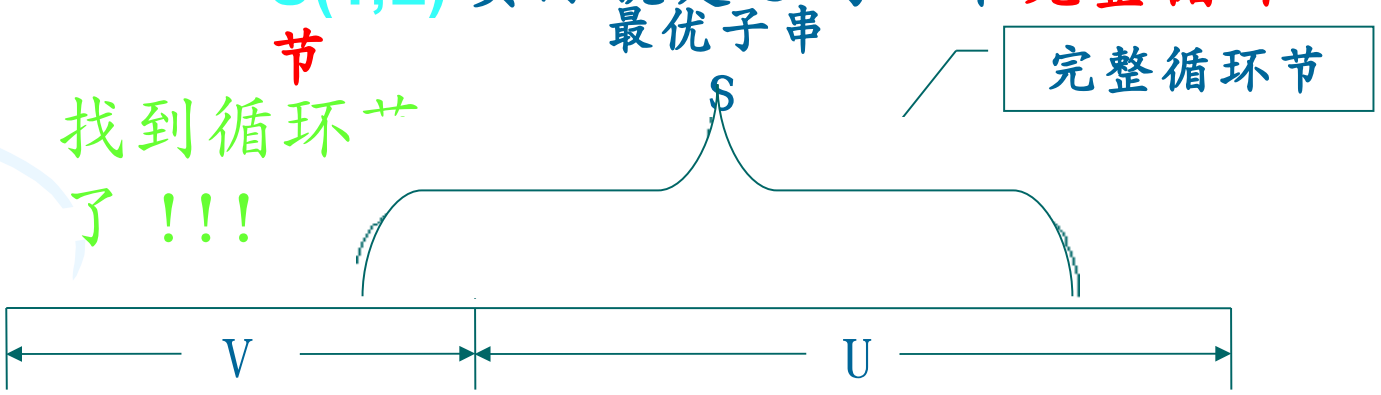
三、循环节扩展和长度判定

1、尽量向右扩展

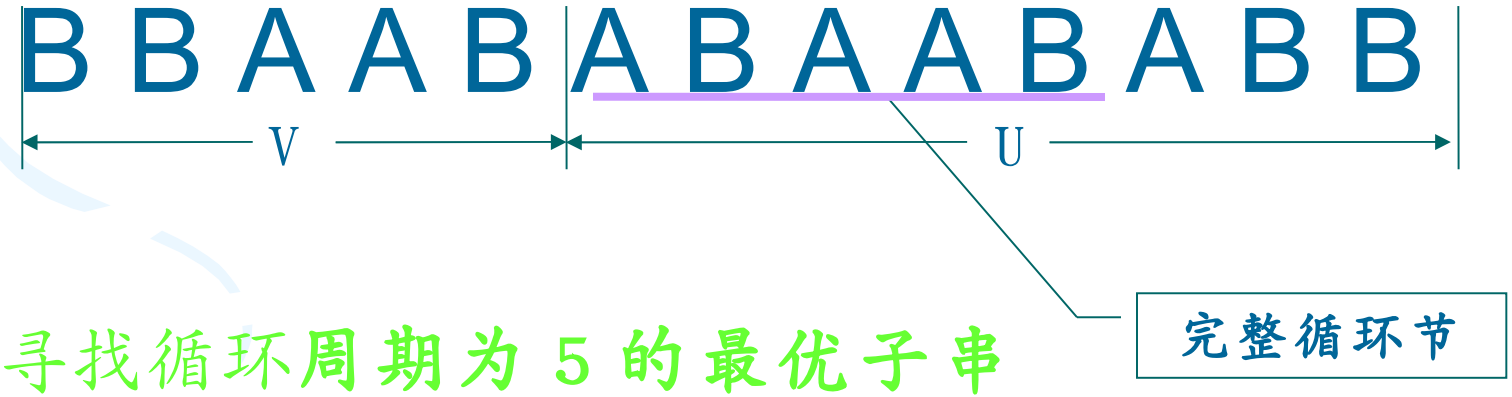
2、尽量向左扩展

3、如果扩展以后的 |S| >= 2L，那么 S 是**最优子串**。

U(1,L) 实际就是 S 的一个**完整循环节**

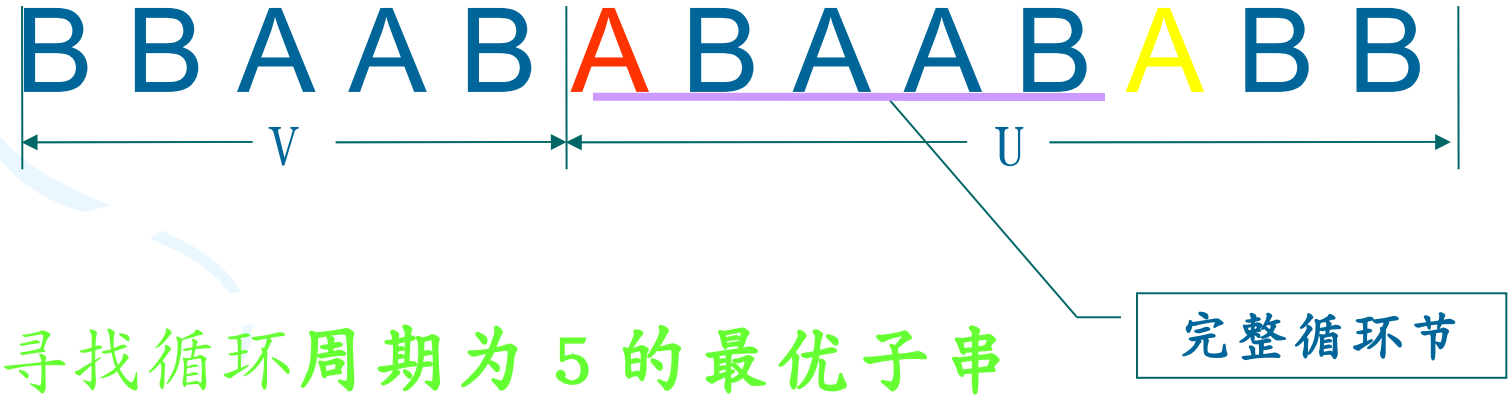


举例



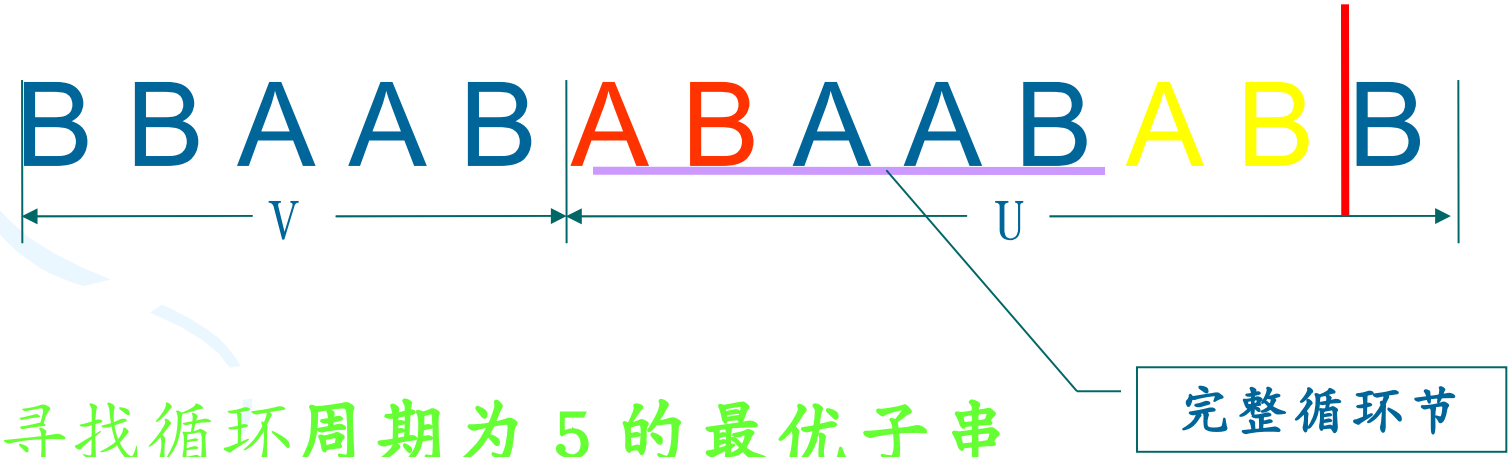
寻找循环周期为 5 的最优子串

举例



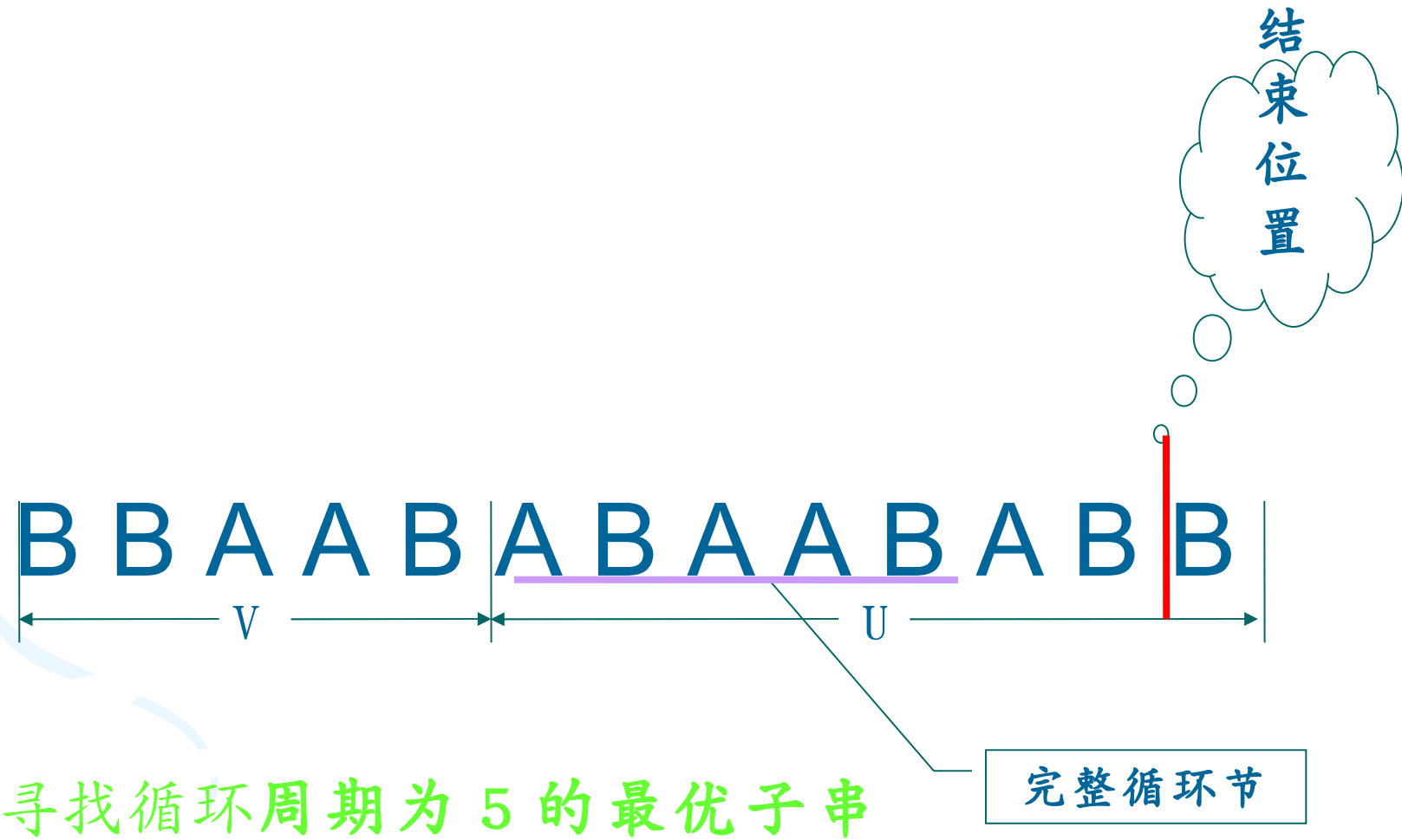
寻找循环周期为 5 的最优子串

举例



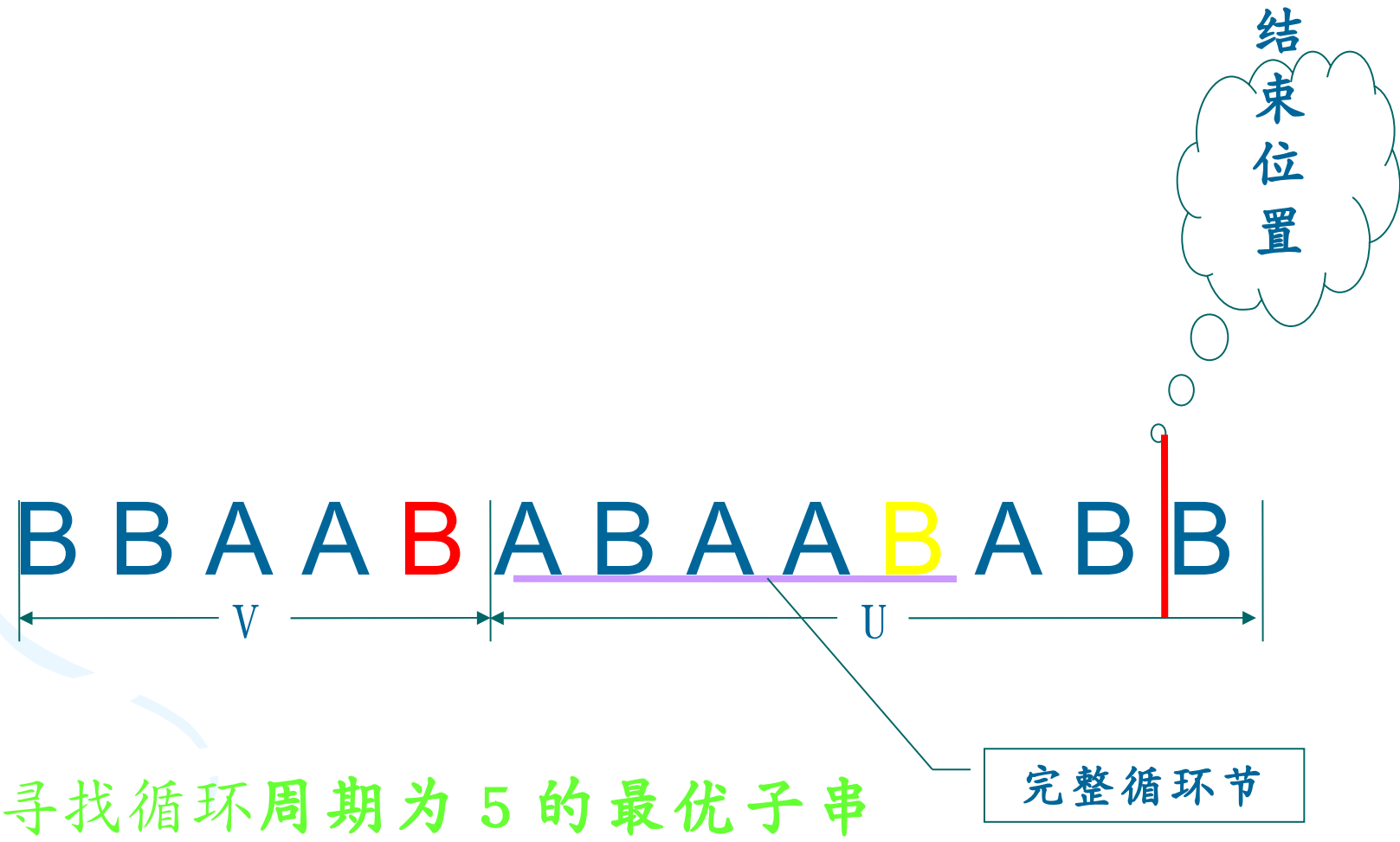
寻找循环周期为 5 的最优子串

举例



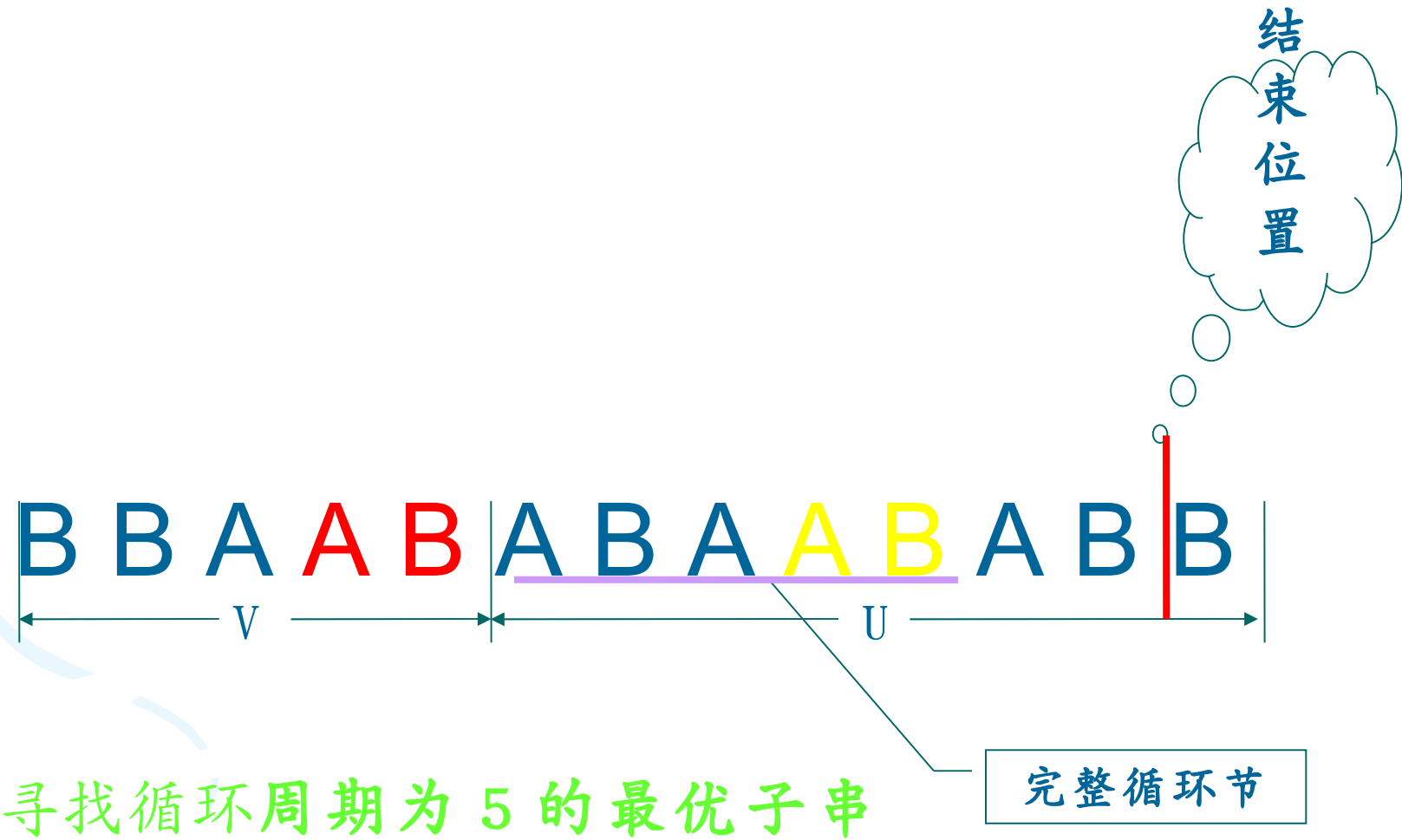
寻找循环周期为 5 的最优子串

举例



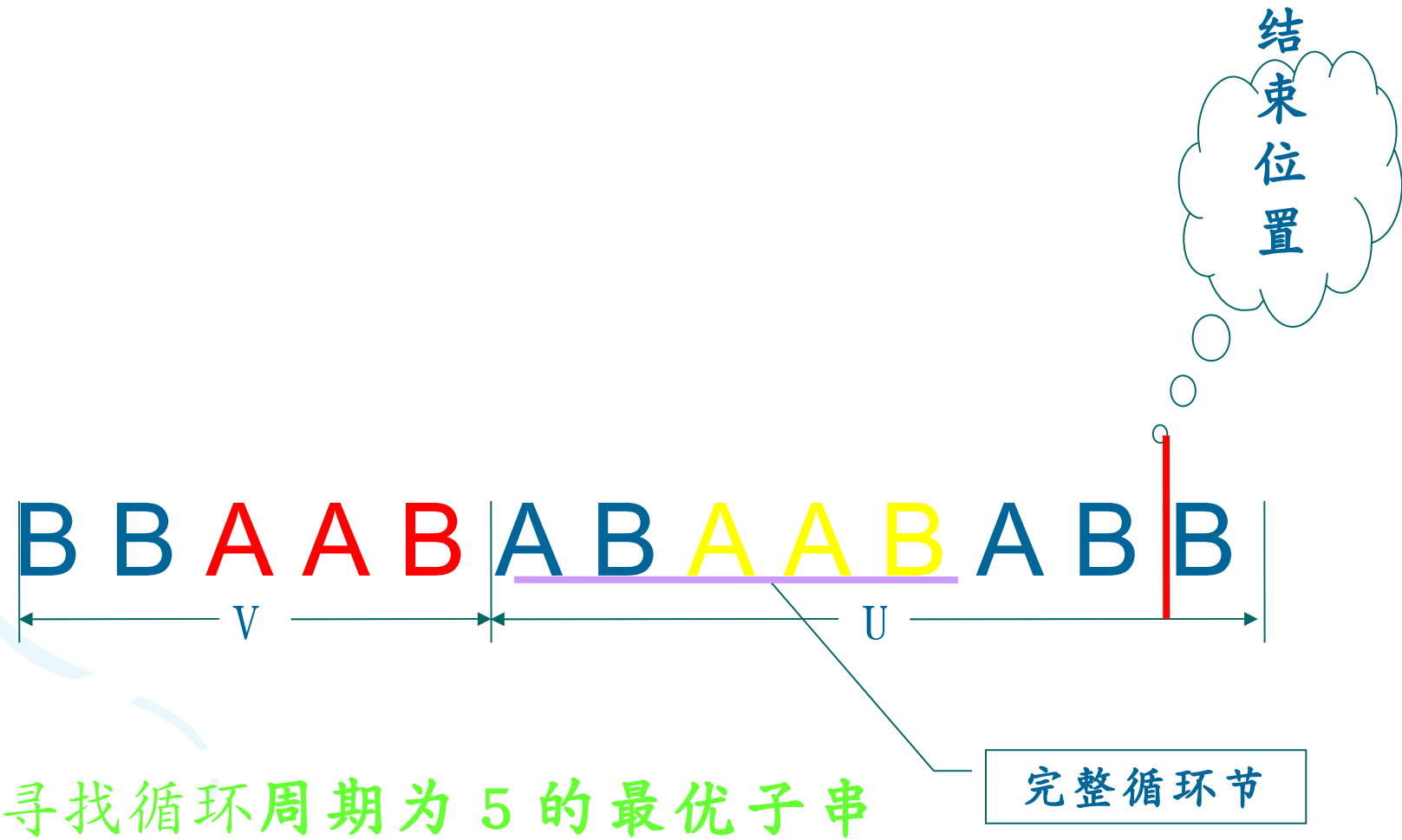
寻找循环周期为 5 的最优子串

举例



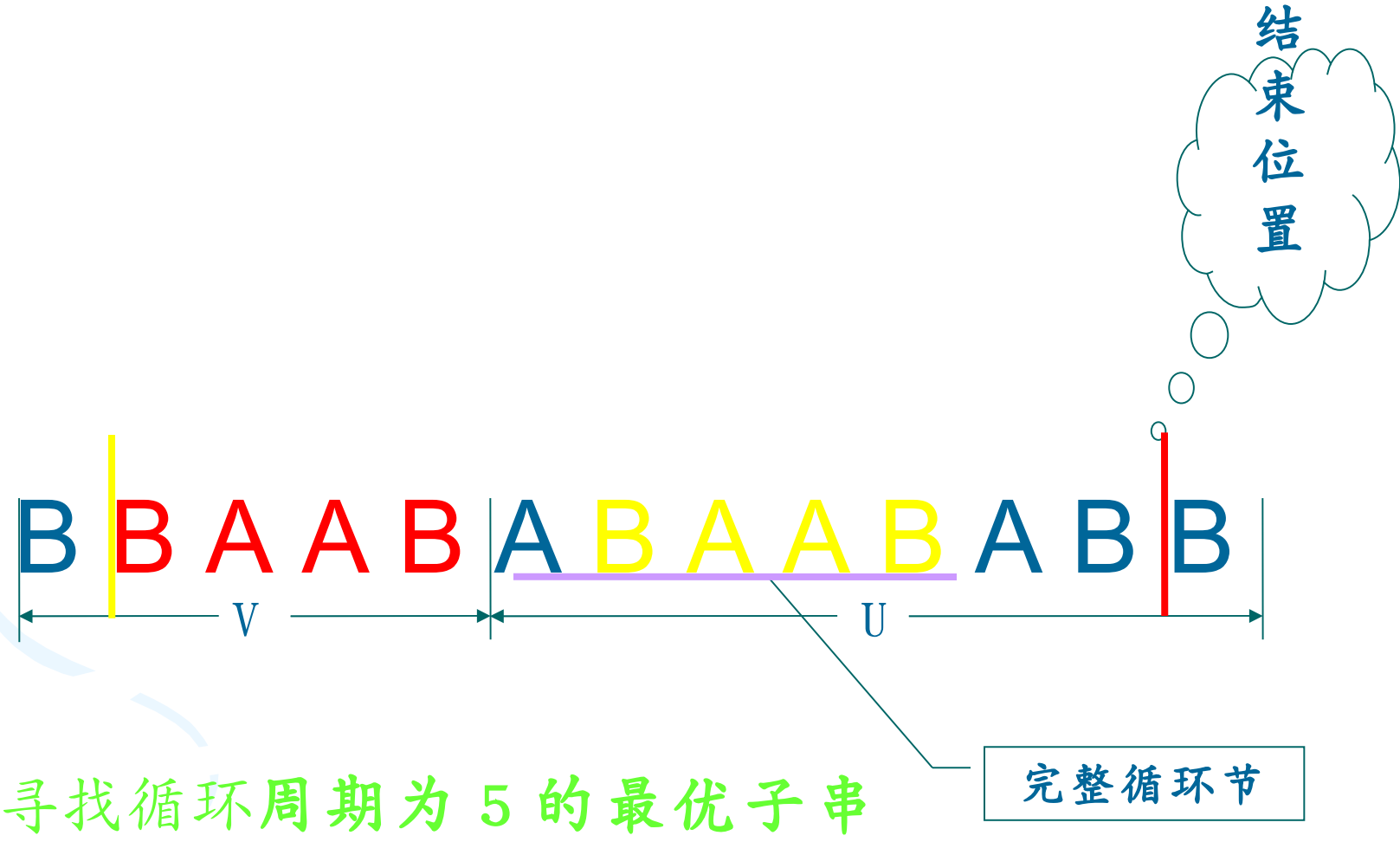
寻找循环周期为 5 的最优子串

举例



寻找循环周期为 5 的最优子串

举例



寻找循环周期为 5 的最优子串

举例

开始位置

结束位置

长度判定：

$|S| = 11 >= 2 * 5$

S 是合法最优子串

完整循环节

寻找循环周期为 5 的最优子串

辅助函数和重要结论 2

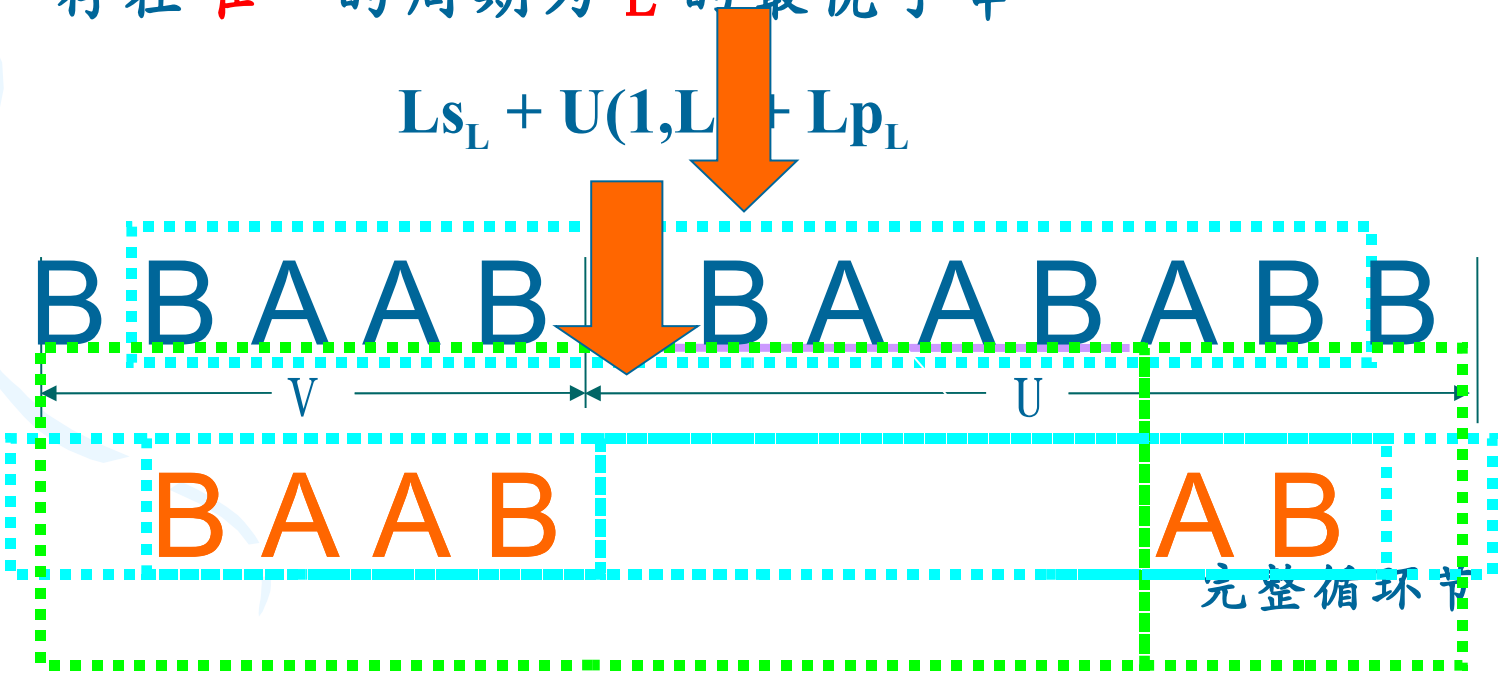
Ls_L = U 与 U(1+L， |U|) 的最长公共前缀 ⇒ 向右扩展

Lp_L = V 与 V + U(1，L) 的最长公共后缀 ⇒ 向左扩展

当且仅当 $|Ls_L + Lp_L| \geq L$ 时 ⇒ 长度判定

存在唯一的周期为 L 的最优字串

$Ls_L + U(1,L) + Lp_L$



长度判定：|S| = 11 >= 2 * 5

四、枚举所有最优子串

这样 Lp 和 Ls 函数的平摊求解复杂度为 O(1)

$Ls_L = U$ 与 U(1+L， |U|) 的最长公共前缀

$Lp_L = V$ 与 V+U(1，L) 的最长公共后缀

因为：Ls 函数定义中，第二个有待比较前缀的字符串总是 U

Lp 函数定义中，第一个有待比较后缀的字符串总是 V

使用一次“KMP 模式匹配的推广”算法，在线性时间内求出所有 Lp 和 Ls 的函数值

所以：我们可以

然后：从 1 到 |Ui + Ui-1| 枚举循环节的长度 L，

并在枚举的同时判断是否 $|Ls_L + Lp_L| >= L$，

即可：找出所有最优子串连同它们的周期。

算法基本框架回顾和完善

字符串分解

answer = 0

For i = 2 to m do

令　V = 长度为 $|U_i| + 2 * |U_{i-1}|$ 的 P 的后缀

$U = U_i$

针对情况 1：S 在 V 中的长度 >= L

… …

End 情况 1

针对情况 2：S 在 U 中的长度 >= L

1、 求出函数 Ls 和函数 Lp 的值

2、 For L=1 to $|U_{i-1} + U_i|$-1 do

If $|Ls_L| + |Lp_L|$ >= L

Then 用 L 更新 answer 的值

End 情况 2

End For

输出 answer

算法性能分析

程序步骤	算法名称	复杂度	常数因子
1、字符串分解	后缀树算法	O(n)	较大
2、辅助函数 $Sum\{2(\|U_{i-1}\|+\|U_i\|)\} = 4n$	KMP 模式匹配	O (n)	< 20
3、枚举所有最优子串 $Sum\{\|U_{i-1}\|+\|U_i\|\} = 2n$	枚举	O (n)	< 10

总结

- 掌握基础算法
- 善于分化问题
- 融会贯通

谢 谢 ！