

本节内容

KMP算法

朴素模式匹配算法的优化

王道考研/CSKAOYAN.COM

1

KMP算法

由D.E.Knuth, J.H.Morris和V.R.Pratt提出, 因此称为 KMP算法



你不要凶我 我害怕。

王道考研/CSKAOYAN.COM

2

最坏情况

Diagram 1: Initial matching. S: [a][a][a][a][a][a][a][a][b], T: [a][a][a][b]. i=1, j=1. Match at (1,1). j moves to 2.

Diagram 2: i=2, j=2. Match at (2,2). j moves to 3.

Diagram 3: i=3, j=3. Match at (3,3). j moves to 4.

Diagram 4: i=4, j=4. No match at (4,4). i moves to 5, j stays at 4.

Diagram 5: i=5, j=1. Match at (5,1). j moves to 2.

Diagram 6: i=6, j=2. Match at (6,2). j moves to 3.

Diagram 7: i=7, j=3. Match at (7,3). j moves to 4.

Diagram 8: i=8, j=4. No match at (8,4). i moves to 8, j stays at 4.

Diagram 9: i=9, j=1. No match at (9,1). i moves to 9, j stays at 1.

主串的扫描指针 i:
前进m步, 后退m-1步, 再
前进m步, 再后退m-1步...

若模式串长度为m, 主串长度为n, 则直到匹配成功/匹配失败最多需要 $n*m$ 次比较
最坏时间复杂度: $O(nm)$

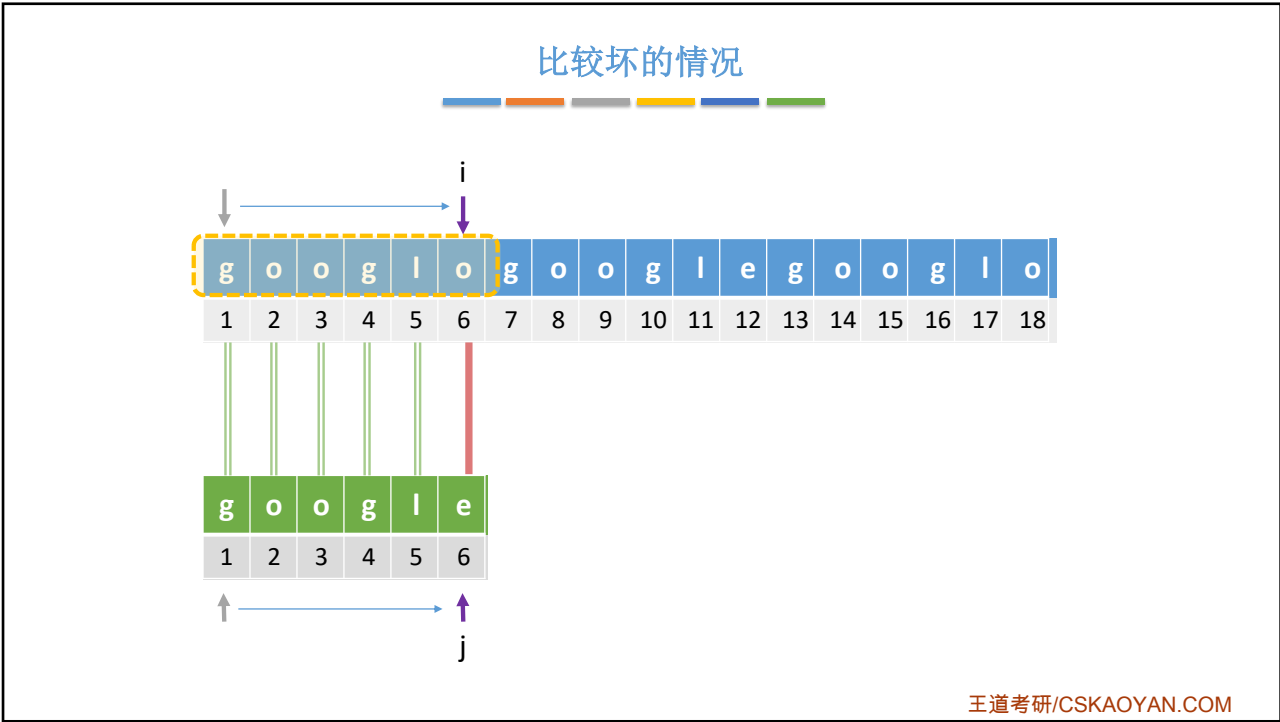
王道考研/CSKAOYAN.COM

3

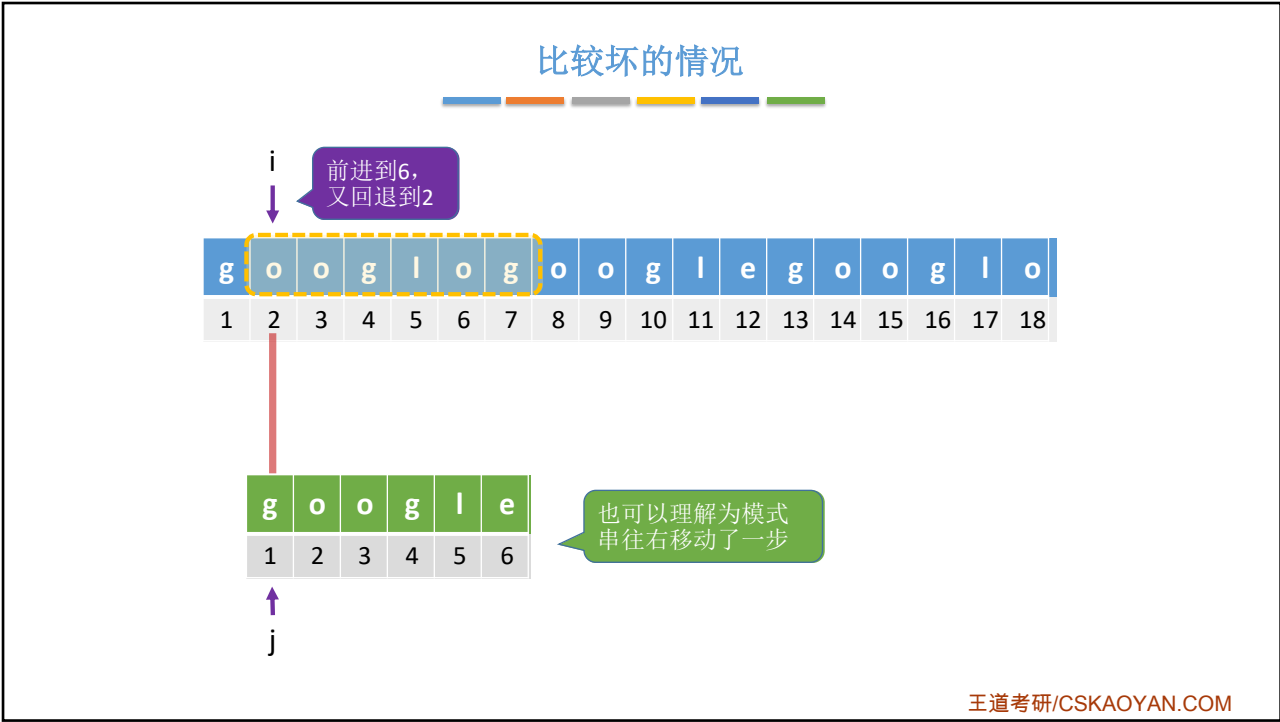
比较坏的情况

王道考研/CSKAOYAN.COM

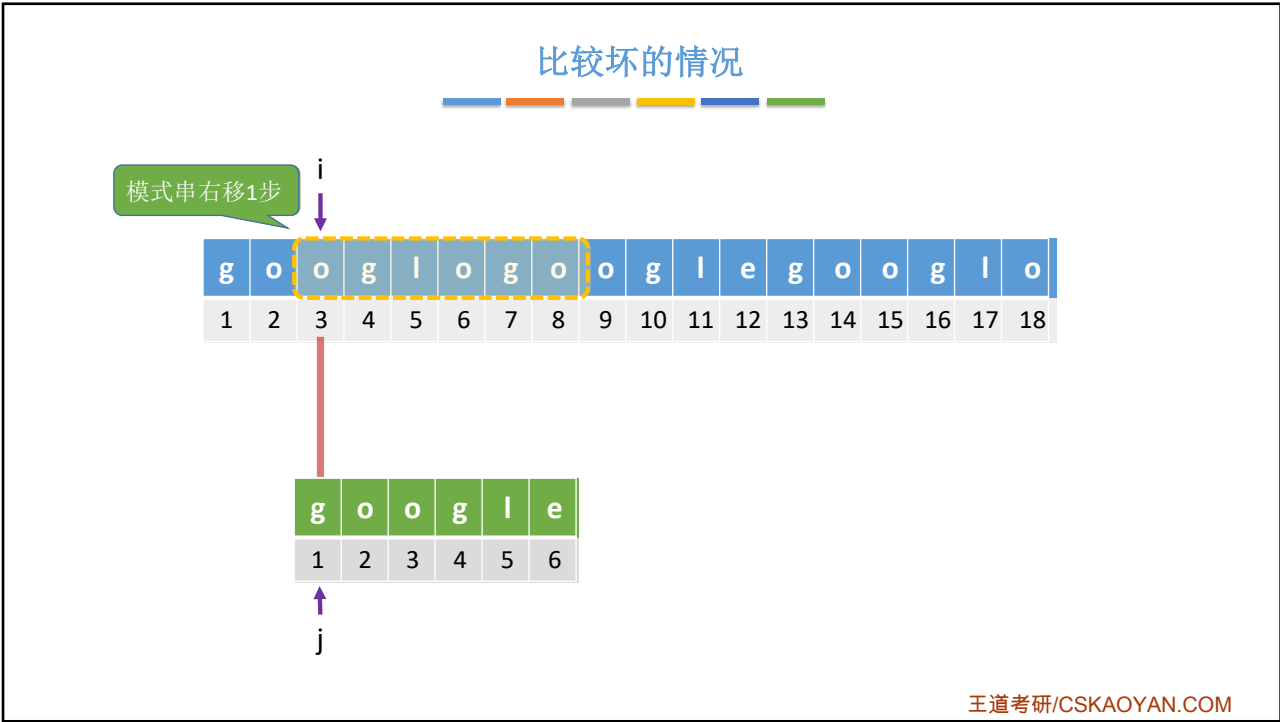
4



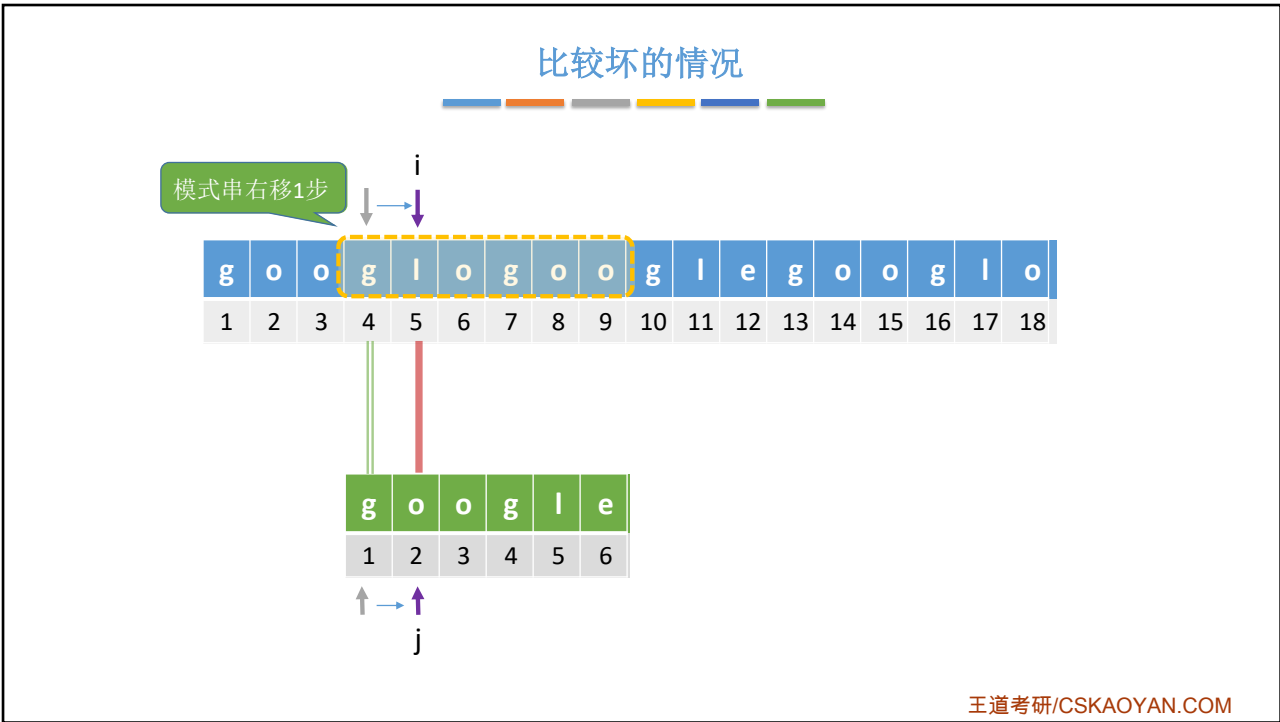
5



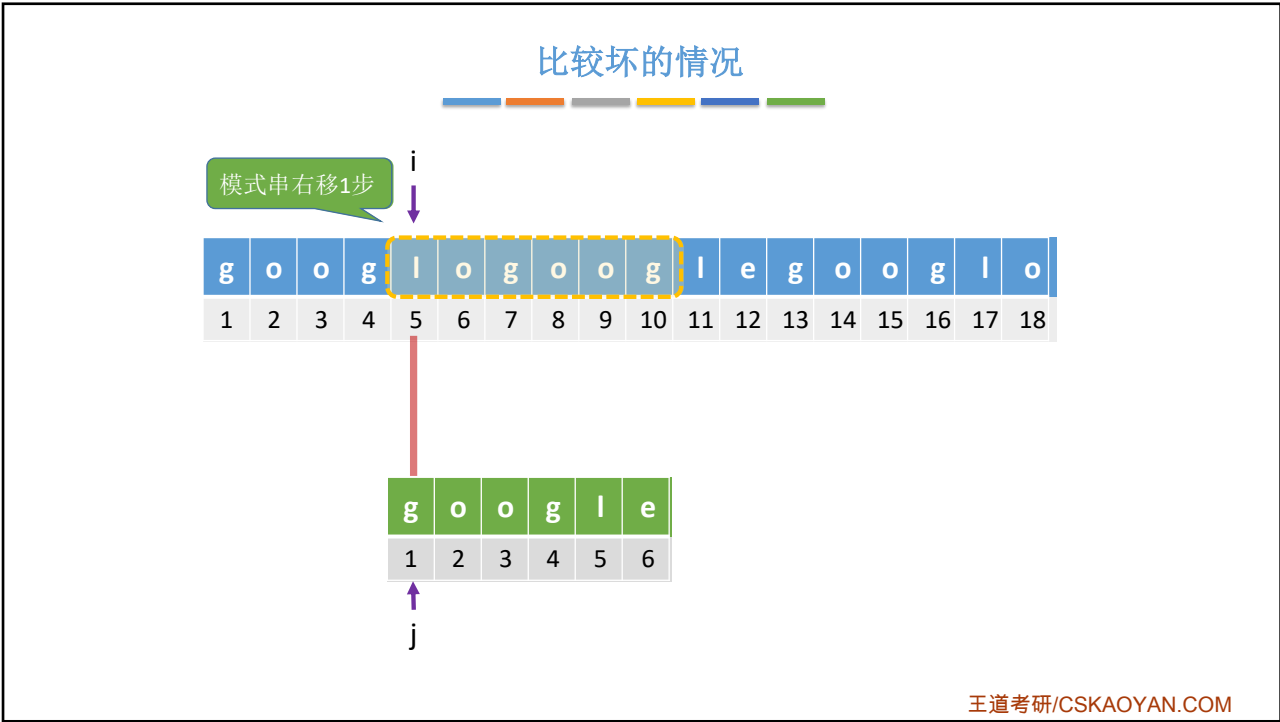
6



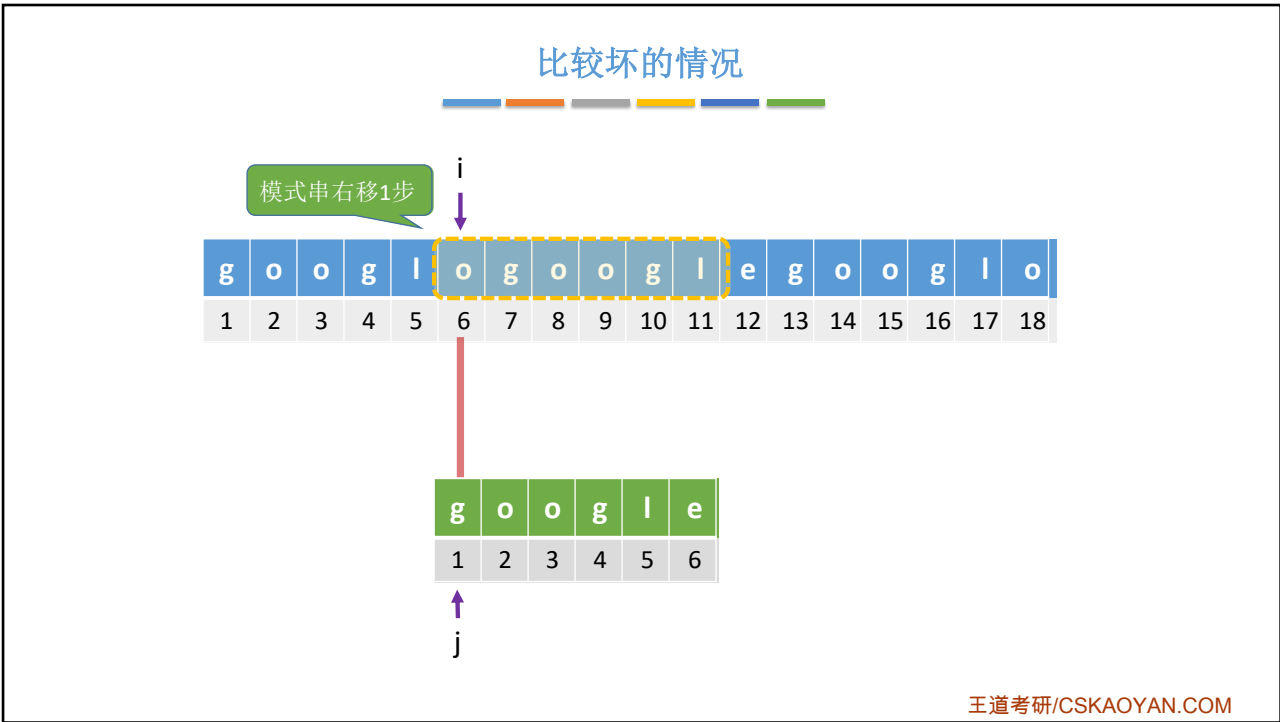
7



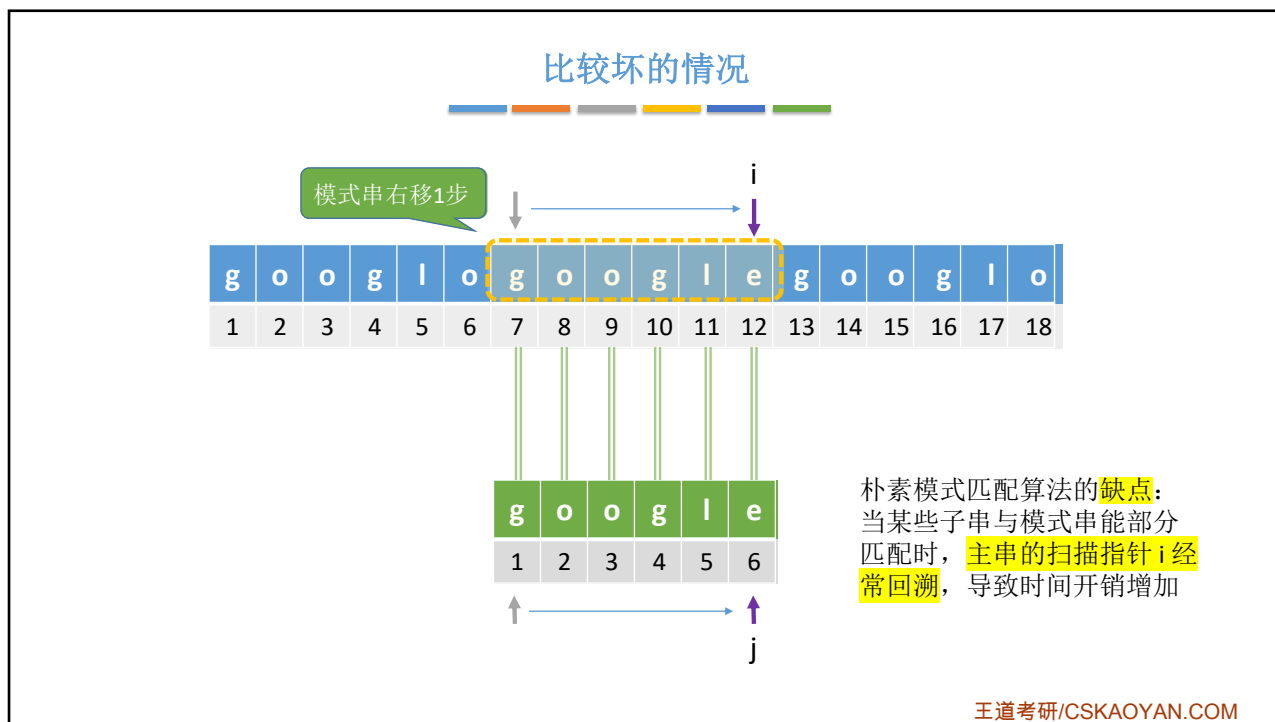
8



9



10



11

看看人脑的反应

任务：找到'google'

要求：你的目光不要回溯

你的目光相当于指针 i

王道考研/CSKAOYAN.COM

12

解决主串回溯问题

这个字符不是e, 你会怎么做?

googl?

值得期待, 是否是g

马赛克

任务: 找到'google'

↑

j

←

↑

i

改进思路: 主串指针不回溯, 只有模式串指针回溯

王道考研/CSKAOYAN.COM

13

模式串指针应该回溯到哪儿?

这个字符不是'e', 该怎么办?

↑

j

←

↑

i

值得期待 是否是g

如果 j = k 时才发现匹配失败, 说明 1~k-1 都匹配成功

x	x	x	x	x	g	o	o	g	l	?	?	?	?	?	?	?	?
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

↑

j

←

↑

i

g o o g l

e

↑

j

←

↑

i

若 j=6 时发生不匹配, 则应让 j 回到 1

王道考研/CSKAOYAN.COM

14

模式串指针应该回溯到哪儿?

王道考研/CSKAOYAN.COM

15

模式串指针应该回溯到哪儿?

王道考研/CSKAOYAN.COM

16

模式串指针应该回溯到哪儿?

这个字符不是'g', 该怎么做?

应该让 i 后移, 检查下一个字符是否是 g

若 j=1 时发生不匹配, 则应让 i++, 而 j 依然是 1

王道考研/CSKAOYAN.COM

17

模式串指针应该回溯到哪儿?

这个字符不是'l', 该怎么做?

值得期待 是否是 o

如果 j = k 时才发现匹配失败, 说明 1~k-1 都匹配成功

若 j=5 时发生不匹配, 则应让 j 回到 2

王道考研/CSKAOYAN.COM

18

模式串指针应该回溯到哪儿？

这个字符不是'l'，
该怎么做？

值得期待
是否是o

如果 $j = k$ 时才发现匹配失败，
说明 $1 \sim k-1$ 都匹配成功

x	x	x	x	g	o	o	g	?	?	?	?	?	x	x	x	x	x
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

少了3次对比

若 $j=5$ 时发生不匹配，则应让 j 回到 2

g	o	o	g	l	e
1	2	3	4	5	6

j

王道考研/CSKAOYAN.COM

19

模式串指针应该回溯到哪儿？

这个字符不是'g'，
该怎么做？

值得期待
是否是g

如果 $j = k$ 时才发现匹配失败，
说明 $1 \sim k-1$ 都匹配成功

x	x	g	o	o	?	?	?	?	?	?	?	?	x	x	x	x	x
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

若 $j=4$ 时发生不匹配，则应让 j 回到 1

g	o	o	g	l	e
1	2	3	4	5	6

j

王道考研/CSKAOYAN.COM

20

模式串指针应该回溯到哪儿?

这个字符不是'g', 该怎么做?

值得期待 是否是 g

如果 $j = k$ 时才发现匹配失败, 说明 $1 \sim k-1$ 都匹配成功

x	x	g	o	o	?	?	?	?	?	?	?	?	x	x	x	x	x
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

少了2次对比

g	o	o	g	l	e
1	2	3	4	5	6

若 $j=4$ 时发生不匹配, 则应让 j 回到 1

王道考研/CSKAOYAN.COM

21

模式串指针应该回溯到哪儿?

这个字符不是'o', 该怎么做?

值得期待 是否是 g

如果 $j = k$ 时才发现匹配失败, 说明 $1 \sim k-1$ 都匹配成功

x	x	g	o	?	?	?	?	?	?	?	?	?	x	x	x	x	x
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

g	o	o	g	l	e
1	2	3	4	5	6

若 $j=3$ 时发生不匹配, 则应让 j 回到 1

王道考研/CSKAOYAN.COM

22

模式串指针应该回溯到哪儿?

这个字符不是'o', 该怎么做?

值得期待 是否是 g

如果 $j = k$ 时才发现匹配失败, 说明 $1 \sim k-1$ 都匹配成功

x	x	g	o	?	?	?	?	?	?	?	?	?	x	x	x	x	x
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

少了1次对比

g	o	o	g	l	e
1	2	3	4	5	6

j

若 $j=3$ 时发生不匹配, 则应让 j 回到 1

王道考研/CSKAOYAN.COM

23

模式串指针应该回溯到哪儿?

这个字符不是'o', 该怎么做?

值得期待 是否是 g

如果 $j = k$ 时才发现匹配失败, 说明 $1 \sim k-1$ 都匹配成功

x	x	g	?	?	?	?	?	?	?	?	?	?	x	x	x	x	x
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

g	o	o	g	l	e
1	2	3	4	5	6

j

若 $j=2$ 时发生不匹配, 则应让 j 回到 1

王道考研/CSKAOYAN.COM

24

模式串指针应该回溯到哪儿?

这个字符不是'o', 该怎么做?

i

↓

x	x	g	?	?	?	?	?	?	?	?	?	?	x	x	x	x	x
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

值得期待 是否是 g

如果 j = k 时才发现匹配失败, 说明 1~k-1 都匹配成功

g	o	o	g	l	e
1	2	3	4	5	6

↑

j

若 j=2 时发生不匹配, 则应让 j 回到 1

王道考研/CSKAOYAN.COM

25

模式串指针应该回溯到哪儿?

如果 j = k 时才发现匹配失败, 说明 1~k-1 都匹配成功

g	o	o	g	l	e
1	2	3	4	5	6

↑

j

int next[7];

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

当 j=k 且发现字符不匹配时, 令 j=next[k]

若当前两个字符匹配, 则 i++, j++

若 j=1 时发生不匹配, 则应让 i++, 而 j 依然是 1

若 j=2 时发生不匹配, 则应让 j 回到 1

若 j=3 时发生不匹配, 则应让 j 回到 1

若 j=4 时发生不匹配, 则应让 j 回到 1

若 j=5 时发生不匹配, 则应让 j 回到 2

若 j=6 时发生不匹配, 则应让 j 回到 1

王道考研/CSKAOYAN.COM

26

KMP算法代码

```
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;
    }
}
```

g	o	o	g	l	e
1	2	3	4	5	6

↑
j

int next[7];

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

当 j=k 且发现字符不匹配时, 令 j=next[k]

王道考研/CSKAOYAN.COM