n为主串长度, m为模式串长度,求 模式匹配 (**子串或模式串**在**主串**中的位置)

朴素模式匹配算法

时间复杂度为 $O(nm)$

```
int StrIndex(SString Str1,SString Str2)//定位操作
{
    int pos1=1,pos2=1;//分别指向Str1和Str2的第一个元素
    while(pos1<=Str1.length&&pos2<=Str2.length)
    {
        if(Str1.ch[pos1]==Str2.ch[pos2])
            pos1++,pos2++;
        else pos1=pos1-pos2+2,pos2=1;
    }
    if(pos2>Str2.length)
        return pos1-Str2.length;
    else return 0;
}
```

KMP算法

时间复杂度为 $O(m+n)$,求 $next$数组时间复杂度为 $O(m)$,模式匹配过程最坏时间复杂度为 $O(n)$.

```
int StrIndex_KMP(SString Str1,SString Str2,int next[])//定位操作
{
    int i=1,j=1;//分别指向Str1和Str2的第一个元素
    while(i<=Str1.length&&j<=Str2.length)
    {
        if(j==0||Str1.ch[i]==Str2.ch[j])//j==0时也需要
            i++,j++;
        else j=next[j];
    }
    if(j>Str2.length)
        return i-Str2.length;
    else return 0;
}
```

计算next数组

手算思想: $next[1]$填 0; $next[2]$填 1;其他的 $next$,在不匹配的位置前,划一根分界线,模式串一步一步往后退,直到分界线之前 能对上 ,或模式串完全跨过分界线为止.此时 j指向哪, $next$数组值就是多少.

```
//教材写法
void get_next(SString Str,int next[])
{
    int i=1,j=0;
    next[1]=0;

    while(i<Str.length)
    {
        if(j==0||Str.ch[i]==Str.ch[j])
            i++,j++,next[i]=j;
        else j=next[j];//往前找
    }
}
```

计算nextval数组

手算思想: 先求 $next$数组,再由 $next$数组求 $nextval$数组.当 $s[next[j]] == s[j]$,则 $nextval[j] = nextval[next[j]]$;反之, $nextval[j] = next[j]$.

```
void get_next_val(SString Str,int next[],int nextval[])//可通过next数组直接求
{
    nextval[1]=0;
    for(int j=2;j<=Str.length;j++)
    {
        if(Str.ch[next[j]]==Str.ch[j])
            nextval[j]=nextval[next[j]];
        else nextval[j]=next[j];
    }
}

//教材写法
void get_next_val(SString Str,int nextval[])
{
    int i=1,j=0;
    nextval[1]=0;

    while(i<Str.length)
    {
        if(j==0||Str.ch[i]==Str.ch[j])
        {
            i++,j++;
            if(Str.ch[i]!=Str.ch[j])
                nextval[i]=j;
            else nextval[i]=nextval[j];

        }
        else j=nextval[j];//往前找
    }
}
```

关于计算 $next$和 $nextval$可参考上面的**手算**思想

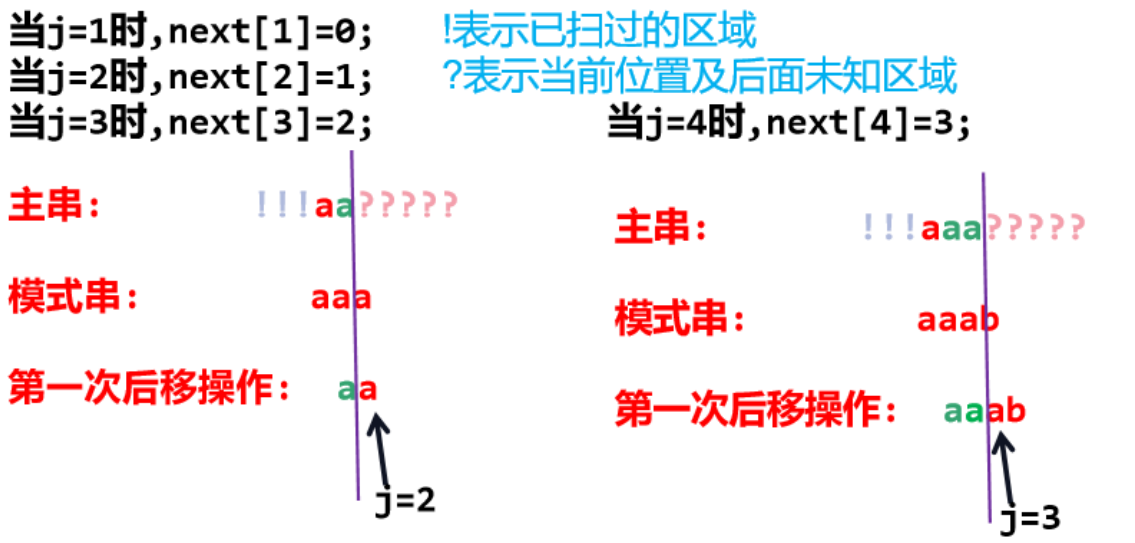
已知串 $S =' aaab'$，其 $next$数组值为().

A. 0123

B. 0112

C. 0231

D. 1211



编号j	1	2	3	4
模式串	a	a	a	b
next[j]	0	1	2	3

教材做法

1）设 next[1]=0，next[2]=1。

编号	1	2	3	4
S	a	a	a	b
next	0	1		

2）j=3 时 k=next[j-1]=next[2]=1，观察 S[j-1]（S[2]）与 S[k]（S[1]）是否相等，S[2]=a，S[1]=a，S[2]=S[1]，所以 next[j]=k+1=2。

```
     ↓j-1=2
a    a    a    b
     a    a    a    b
     ↑k=1
```

3）j=4 时 k=next[j-1]=next[3]=2，观察 S[j-1]（S[3]）与 S[k]（S[2]）是否相等，S[3]=a，S[2]=a，S[3]=S[2]，所以 next[j]=k+1=3。

```
          ↓j-1=3
a    a    a    b
     a    a    a    b
          ↑k=2
```

最后的结果如下，选 A。

编号	1	2	3	4
S	a	a	a	b
next	0	1	2	3

串 $'ababaaababaa'$ 的 $next$ 数组值为().

A. 012345678999

B. 012121111212

C. 011234223456

D. 012301232234

串 $'ababaaababaa'$ 的 $next$ 数组为().

A. $-1,0,1,2,3,4,5,6,7,8,8,8$

B. $-1,0,1,0,1,0,0,0,0,1,0,1$

C. $-1,0,0,1,2,3,1,1,2,3,4,5$

D. $-1,0,1,2,-1,0,1,2,1,1,2,3$

串 $'ababaaababaa'$ 的 $nextval$ 数组为().

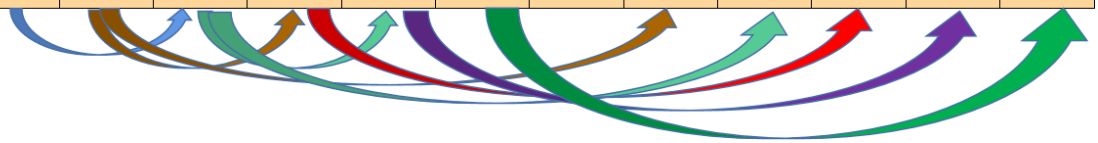
A. $0,1,0,1,1,2,0,1,0,1,0,2$

B. $0,1,0,1,1,4,1,1,0,1,0,2$

C. $0,1,0,1,0,4,2,1,0,1,0,4$

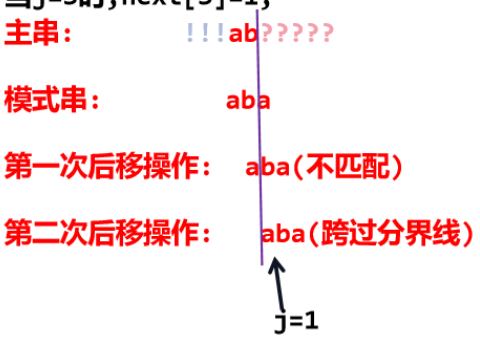
D. $0,1,1,1,0,2,1,1,0,1,0,4$

编号j	1	2	3	4	5	6	7	8	9	10	11	12
模式串	a	b	a	b	a	a	a	b	a	b	a	a
next[j] 位序	0	1	1	2	3	4	2	2	3	4	5	6
next[j-1] 下标	-1	0	0	1	2	3	1	1	2	3	4	5
nextval[j] 位序	0	1	0	1	0	4	2	1	0	1	0	4

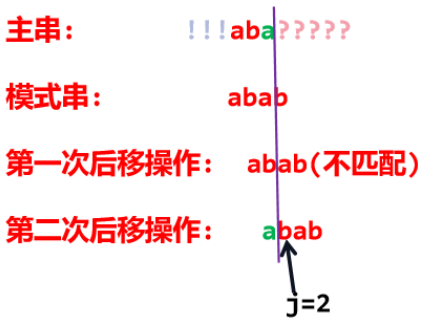


- 求$next$数组

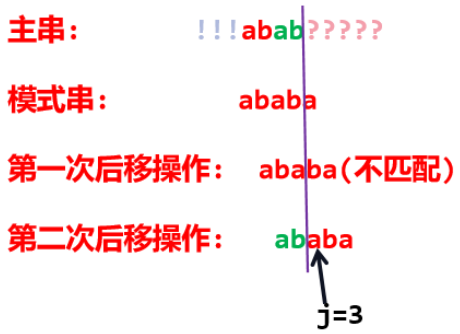
求next过程
当j=1时,next[1]=0;
当j=2时,next[2]=1;
当j=3时,next[3]=1;



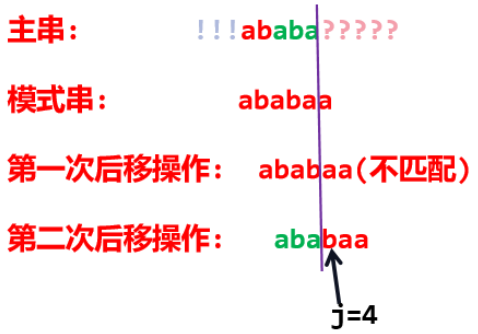
当j=4时,next[4]=2;



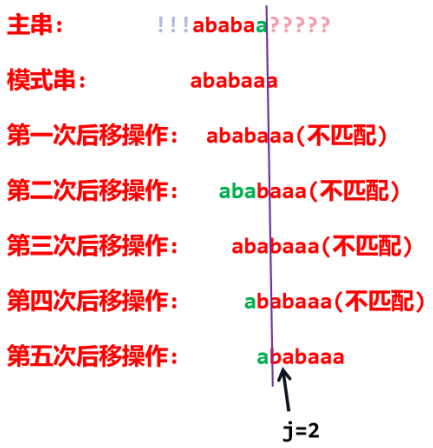
当j=5时,next[5]=3;



当j=6时,next[6]=4;

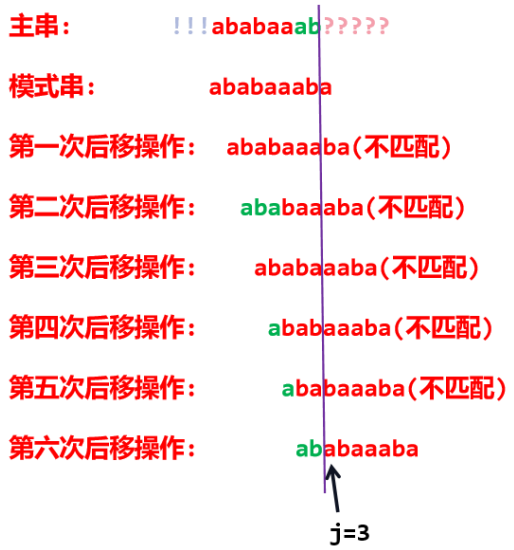


当j=7时,next[7]=2;

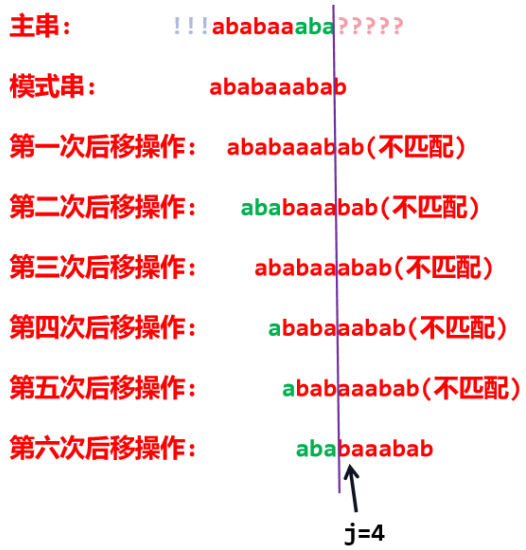


当j=8时,next[8]=2;

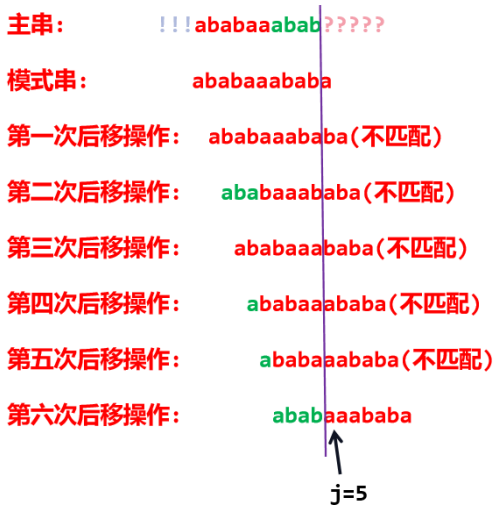
当j=9时,next[9]=3;



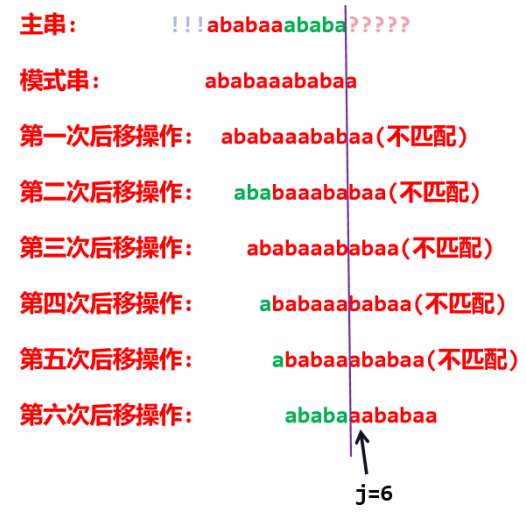
当j=10时,next[10]=4;



当j=11时,next[11]=5;



当j=12时,next[12]=6;



当求**位序**的 $next$数组,过程如上;当求**下标**的 $next$数组,无非在位序的 $next$数组基础上每个数 -1即可.

- 求 $nextval$数组

求nextval数组

当j=1时,nextval[1]=0;
当j=2时,next[2]=1,s[1]!=s[2],nextval[2]=next[2]=1;
当j=3时,next[3]=1,s[1]==s[3],nextval[3]=nextval[1]=0;
当j=4时,next[4]=2,s[2]==s[4],nextval[4]=nextval[2]=1;
当j=5时,next[5]=3,s[3]==s[5],nextval[5]=nextval[3]=0;
当j=6时,next[6]=4,s[4]!=s[6],nextval[6]=next[6]=4;
当j=7时,next[7]=2,s[2]!=s[7],nextval[7]=next[7]=2;
当j=8时,next[8]=2,s[2]==s[8],nextval[8]=nextval[2]=1;
当j=9时,next[9]=3,s[3]==s[9],nextval[9]=nextval[3]=0;
当j=10时,next[10]=4,s[4]==s[10],nextval[10]=nextval[4]=1;
当j=11时,next[11]=5,s[5]==s[11],nextval[11]=nextval[5]=0;
当j=12时,next[12]=6,s[6]==s[12],nextval[12]=nextval[6]=4.

教材做法(忽略水印)

- 'a'的前后缀都为空，最长相等前后缀长度为 0。
- 'ab'的前缀{a}∩后缀{b}=∅，最长相等前后缀长度为 0。
- 'aba'的前缀{a,ab}∩后缀{a,ba}={a}，最长相等前后缀长度为 1。
- 'abab'的前缀{a,ab,aba}∩后缀{b,ab,bab}={ab}，最长相等前后缀长度为 2。
- ……

依次求出的部分匹配值如下表第三行所示，将其整体右移一位，低位用-1 填充，如下表第四行所示。

编号	1	2	3	4	5	6	7	8	9	10	12	12
S	a	b	a	b	a	a	a	b	a	b	a	a
PM	0	0	1	2	3	1	1	2	3	4	5	6
next	-1	0	0	1	2	3	1	1	2	3	4	5

nextval 从 0 开始，可知串的位序从 1 开始。第一步，令 nextval[1]=next[1]=0。

编号	1	2	3	4	5	6	7	8	9	10	11	12
S	a	b	a	b	a	a	a	b	a	b	a	a
next	0	1	1	2	3	4	2	2	3	4	5	6
nextval	0	1	0	1	0	4	2	1	0	1	0	4

从 j=2 开始，依次判断 p_j 是否等于 $p_{next[j]}$？否则将 next[j] 修正为 next[next[j]]，直至两者不相等为止。由下述推理可知，答案选 C。

第 2 步：$p_2=b$、$p_{next[2]}=a$，$p_2 \neq p_{next[2]}$，nextval[2]=next[2]=1；

第 3 步：$p_3=a$、$p_{next[3]}=a$，$p_3=p_{next[3]}$，nextval[3]=nextval[next[3]]=nextval[1]=0；

第 4 步：$p_4=b$、$p_{next[4]}=b$，$p_4=p_{next[4]}$，nextval[4]=nextval[next[4]]=nextval[2]=1；

第 5 步：$p_5=a$、$p_{next[5]}=a$，$p_5=p_{next[5]}$，nextval[5]=nextval[next[5]]=nextval[3]=0；

第 6 步：$p_6=a$、$p_{next[6]}=b$，$p_6 \neq p_{next[6]}$，nextval[6]=next[6]=4；

第 7 步：$p_7=a$、$p_{next[7]}=b$，$p_7 \neq p_{next[7]}$，nextval[7]=next[7]=2；

第 8 步：$p_8=b$、$p_{next[8]}=b$，$p_8=p_{next[8]}$，nextval[8]=nextval[next[8]]=nextval[2]=1；

第 9 步：$p_9=a$、$p_{next[9]}=a$，$p_9=p_{next[9]}$，nextval[9]=nextval[next[9]]=nextval[3]=0；

第 10 步：$p_{10}=b$、$p_{next[10]}=b$，$p_{10}=p_{next[10]}$，nextval[10]=nextval[next[10]]=nextval[4]=1；

第 11 步：$p_{11}=a$、$p_{next[11]}=a$，$p_{11}=p_{next[11]}$，nextval[11]=nextval[next[11]]=nextval[5]=0；

第 12 步：$p_{12}=a$、$p_{next[12]}=a$，$p_{10}=p_{next[12]}$，nextval[12]=nextval[next[12]]=nextval[6]=4；

在第 5 步的推理中，$p_5=p_{next[5]}=a$，按前面的讲解部分，应该继续让 p_3 和 $p_{next[3]}$ 比较（恰好 $p_3=p_{next[3]=1}$），注意到此时 nextval[3] 的值已存在，故直接将 nextval[5] 赋值为 nextval[3]。

对于一般情况，nextval 数组是从前往后逐步求解的，发生 $p_j=p_{next[j]}$ 时，因为 nextval[next[j]] 早已求得，所以直接将 nextval[j] 赋值为 nextval[next[j]]。

2015统考真题：已知字符串 S为 $'abaabaabacacaabaabcc'$，模式串 t为 $'abaabc'$.采用 KMP算法进行匹配，第一次出现"失配"($s[i] \neq s[j]$)时， $i=j=5$，则下次开始匹配时， i和 j的值分别是().

A. $i=1, j=0$

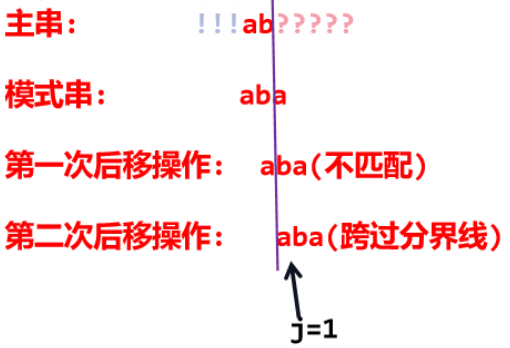
B. $i=5, j=0$

C. $i=5, j=2$

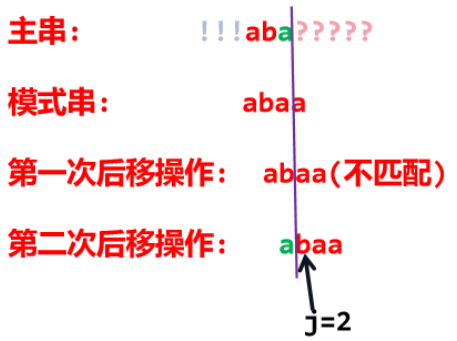
D. $i=6, j=2$

编号j	1	2	3	4	5	6
模式串	a	b	a	a	b	c
next[j]	0	1	1	2	2	3
next[j-1]	-1	0	0	1	1	2

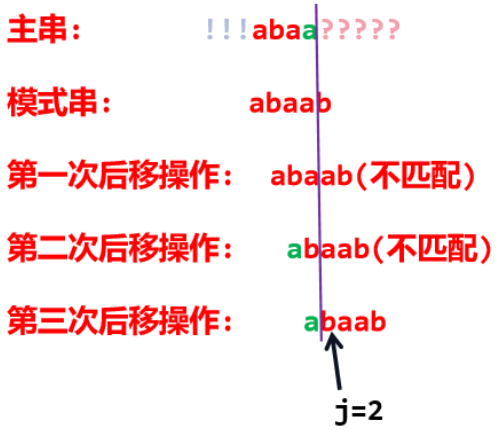
求next过程
当j=1时,next[1]=0;
当j=2时,next[2]=1;
当j=3时,next[3]=1;



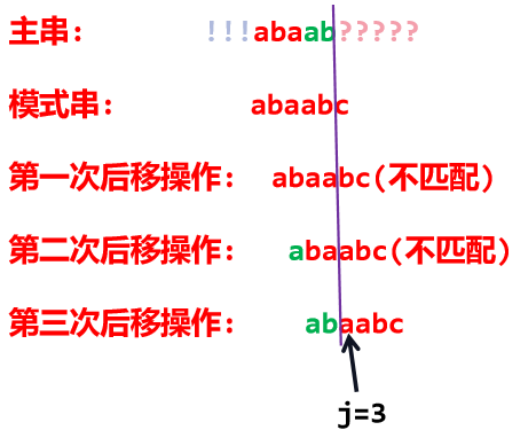
当j=4时,next[4]=2;



当j=5时,next[5]=2;



当j=6时,next[6]=3;



注意题目中 $i=j=5$时失配,可发现是下标法(对应编号是 6),由表知 i不变, j变成对应的 $next$数组值,故 $i=5, j=2$

> **2019统考真题**：设主串 $T=' abaabaabcabaabc'$，模式串 $s=' abaabc'$，采用 KMP算法进行模式匹配，到匹配成功时为止，在匹配过程中进行的单个字符间的比较次数是().
>
> A. 9
>
> B. 10
>
> C. 12
>
> D. 15

编号j	1	2	3	4	5	6
模式串	a	b	a	a	b	c
next[j]	0	1	1	2	2	3
next[j-1]	-1	0	0	1	1	2

模式串同上题的一样.

编号j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
主串	a	b	a	a	b	a	a	b	c	a	b	a	a	b	c
第一次	a	b	a	a	b	c									
第二次				a	b	a	a	b	c						

下划线处表示比较的次数,共 $6+4=10$次.

4.2.1

> 在字符串模式匹配的 KMP算法中，求模式的 $next$数组值的定义如下:
>
> $$next[j]=\begin{cases}0, j=1\\ \max\{k \mid 1<k<j且' p_1Lp_{k-1}'='p_{j-k+1}Lp_{j-1}'\},此空集不为空\\ 1,其他情况\end{cases}$$
>
> 1)当 $j=1$时,为什么要取 $next[1]=0$?
>
> 2)为什么要取 `max{k}`， k最大是多少?
>
> 3)其他情况是什么情况，为什么取 $next[j]=1$?

1)当模式串中的第一个字符和主串的当前字符比较**不相等**时， $next[1]=0$，表示模式串应该右移一位，主串当前指针也要后移一位，再和模式串中的第一字符进行比较;

2)当主串的第 i个字符与模式串的第 j个字符失配时，主串 i不回溯，则假定模式串的第 k个字符与主串的第 i个字符比较， k值应满足条件 $1<k<j$且 $'p_1\cdots p'_{k-1}='p_{j-k+1}\cdots p'_{j-1}$，即 k为模式串的下次比较位置。 k值可能有多个，为了不使向右移动丢失可能的匹配，右移距离应该取最小，由于 $j-k$表示右移的距离，所以取 `max{k}`。

3)除上面两种情况外，发生失配时，主串指针 i不回溯，在最坏情况下，模式串从第 1个字符开始与主串的第 i个字符比较。

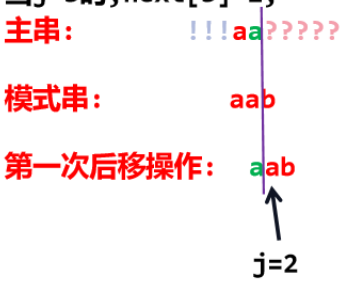
4.2.2

> 设有字符串 $S='aabaabaabaac'$， $P='aabaac'$.
>
> 1)求出 P的 $next$数组.
>
> 2)若 S作主串， P作模式串，试给出 KMP算法的匹配过程。

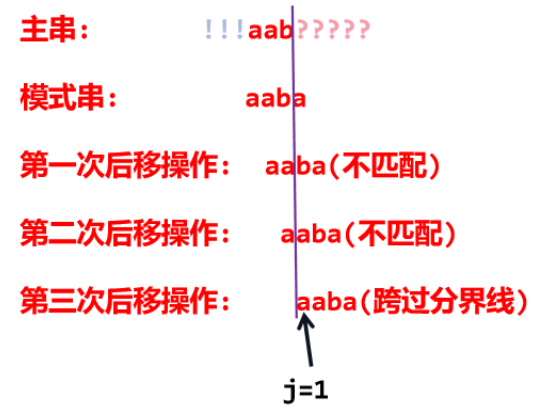
1)

编号j	1	2	3	4	5	6
模式串	a	a	b	a	a	c
next[j]	0	1	2	1	2	3

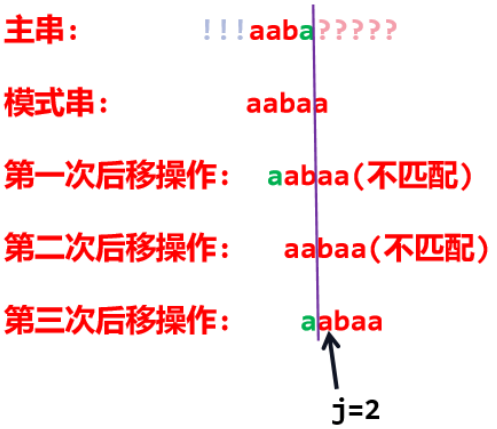
求next过程
当j=1时,next[1]=0;
当j=2时,next[2]=1;
当j=3时,next[3]=2;



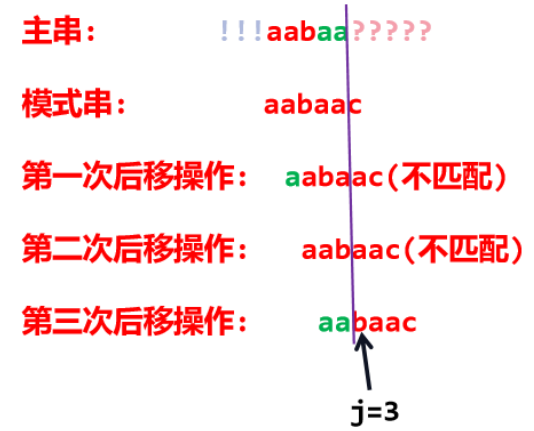
当j=4时,next[4]=1;



当j=5时,next[5]=2;



当j=6时,next[6]=3;



2)

编号j	1	2	3	4	5	6	7	8	9	10	11	12
主串	a	a	b	a	a	b	a	a	b	a	a	c
第一次	a	a	b	a	a	c						
第二次				a	a	b	a	a	c			
第三次							a	a	b	a	a	c