

串

KMP算法：构造next[]表

13-C4

一切都是暂时的，转瞬即逝  
而那逝去的将变为可爱

在这些胡思乱想里，我还怕我的记忆力不听我的使唤，怕记忆力出于疏忽而让我把同一件事写上两次。我讨厌在文章里再次认出自己，我炒冷饭向来是违心的。

邓俊辉

deng@tsinghua.edu.cn

# 减而治之：由next[0]、next[1]、...、next[j]，如何得到next[j+1]？

P在j+1处的自匹配，只比在j处的自匹配多出一个字符...

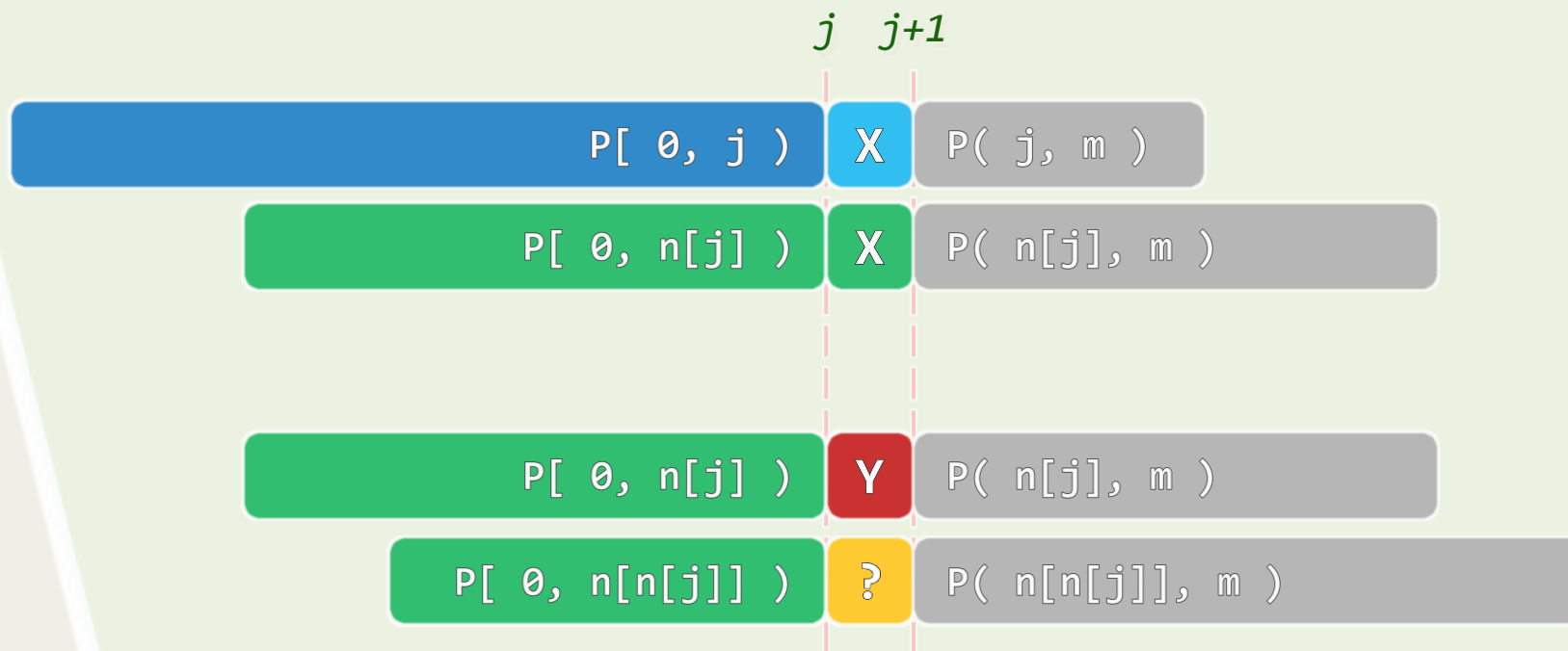
-1 0 0 1 2 3 1 0  
M A M A M M I A

-1 0 0 1  
M A M A M M I A  
M A M A M M I A

-1 0 0 1 2  
M A M A M M I A  
M A M A M M I A

-1 0 0 1 2 3  
M A M A M M I A  
M A M A M M I A

-1 0 0 1 2 3 1  
M A M A M M I A  
M A M A M M I A  
M A M A M M I A  
M A M A M M I A



$$next[j+1] = next[j] + 1 \quad \text{iff} \quad P[j] = P[next[j]]$$

# 算法实现

```
int * buildNext( char * P ) {  
    size_t m = strlen(P), j = 0;  
    int *N = new int[m];  
    int t = N[0] = -1;  
    while ( j < m - 1 )  
        ( 0 > t || P[j] == P[t] ) ?  
            N[ ++j ] = ++t : t = N[t];  
    return N;  
}
```

$j$   $j+1$

\*

$P(-1, m) = P[0, m)$

$P[0, n[n[n[j]]])$

X

$P(n[n[n[j]]], m)$

$P[0, n[n[j]])$

Z

$P(n[n[j]], m)$

$P[0, n[j])$

Y

$P(n[j], m)$

$P[0, j)$

X

$P(j, m)$