

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...

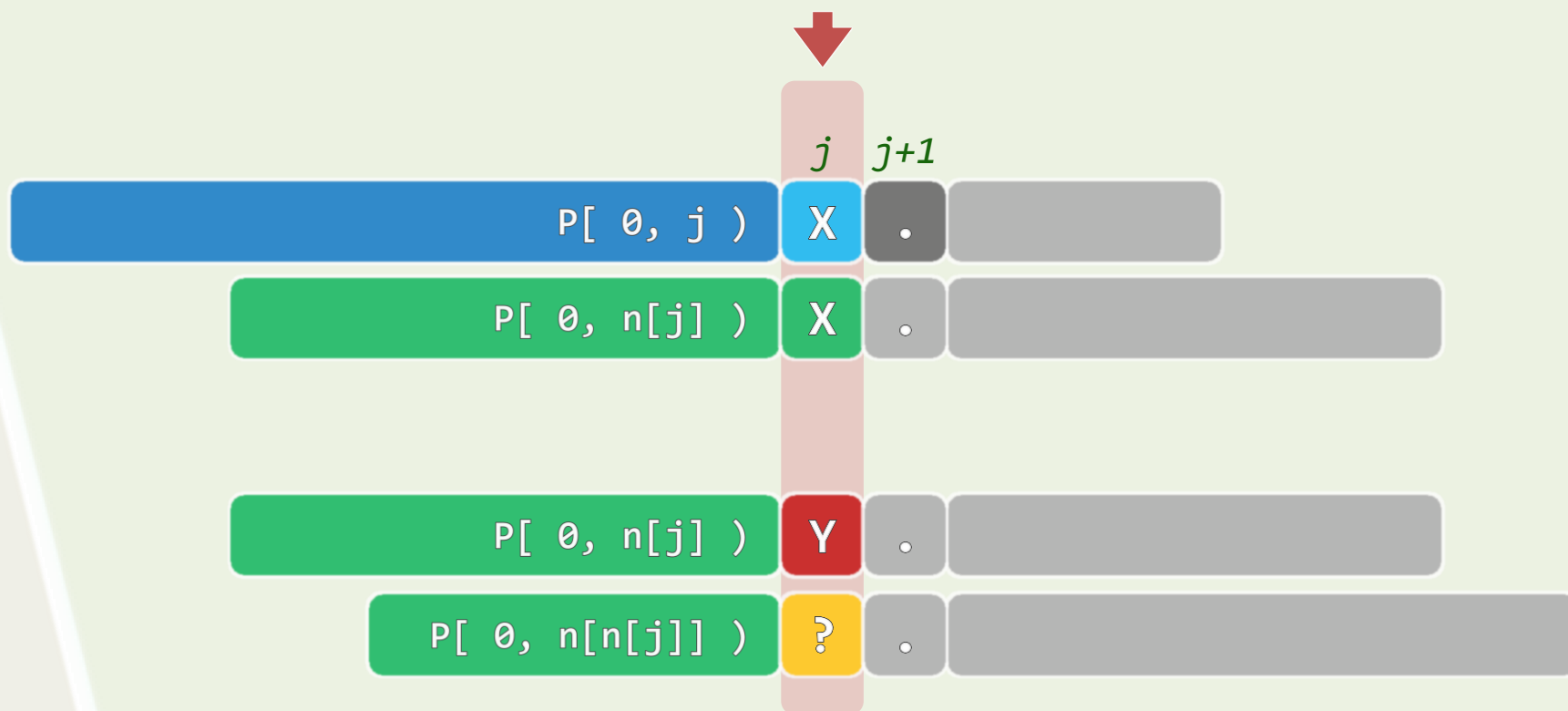
-1 0 0 1 2 3 1 0  
M A M A M M I A

-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* next = new int[m];
```

```
    int t = next[0] = -1;
```

```
    while ( j < m - 1 )
```

```
        if ( 0 > t || P[t] == P[j] ) { //匹配
```

```
            ++t; ++j; next[j] = t; //则递增赋值
```

```
        } else //否则
```

```
            t = next[t]; //继续尝试下一值得尝试的位置
```

```
    return next;
```

```
}
```



\*

X

$n^3[j]$

Z

$n^2[j]$

Y

$n[j]$

X

j

j+1