

# 13-C2

串

KMP算法：查询表

好记性不如烂笔头

你能看见多远的过去，就能看见多远的未来

邓俊辉

deng@tsinghua.edu.cn

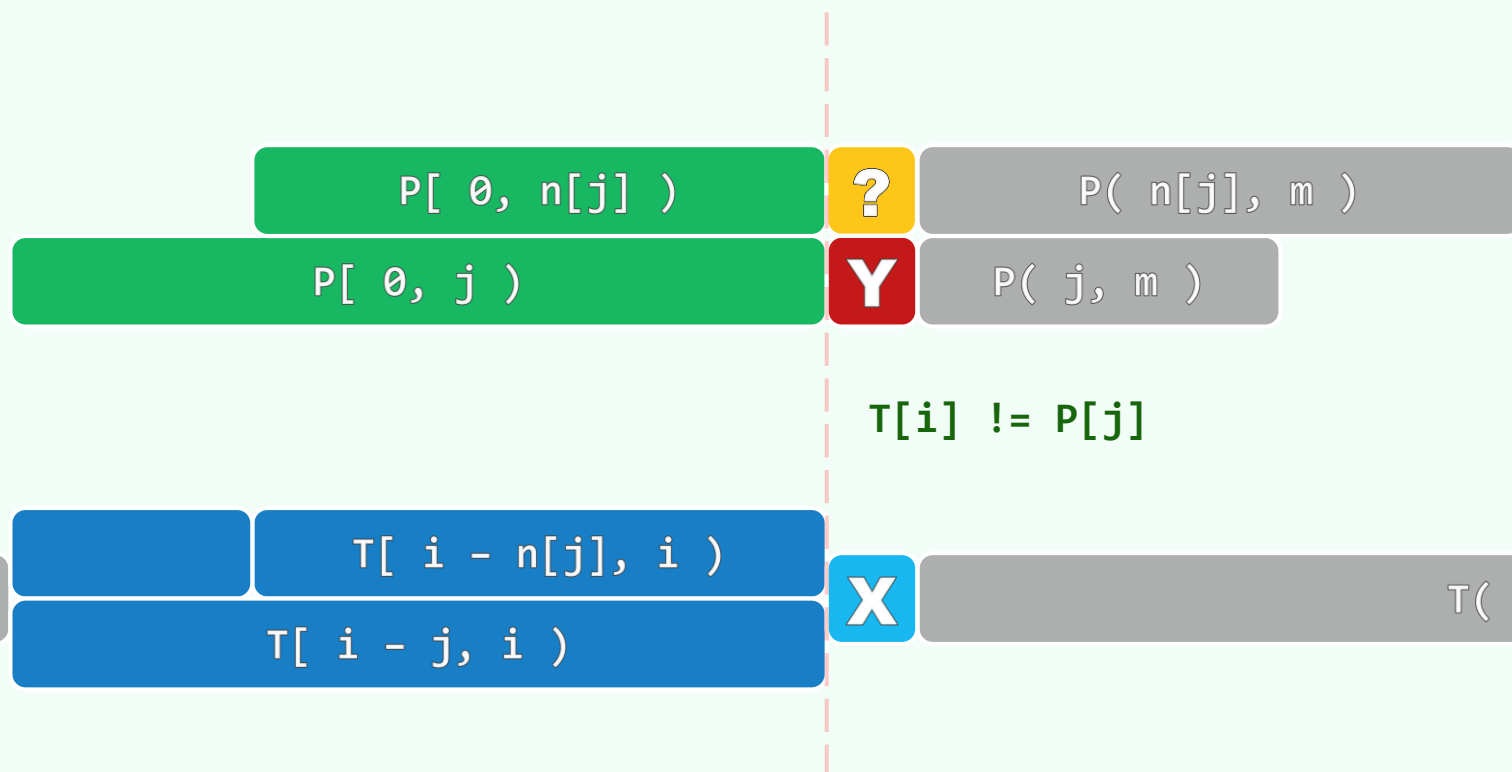
t: 不仅可以事先确定, 而且仅根据 $P[0, j) = T[i-j, i)$ 即可确定

❖ 视失败的位置j, 无非m种情况

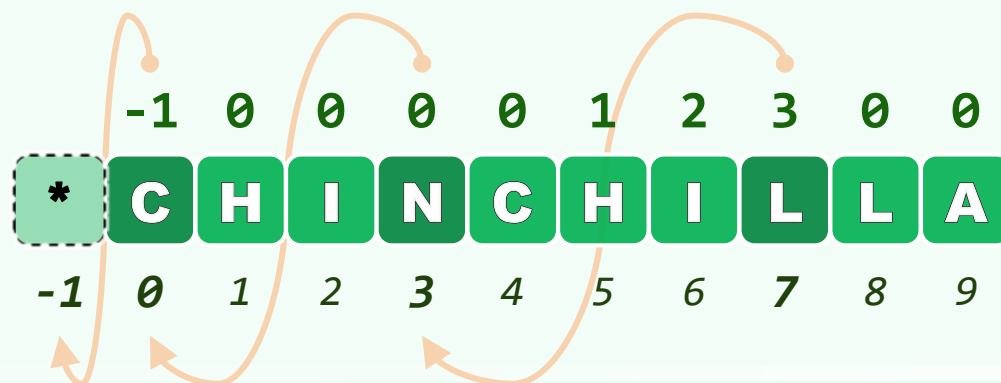
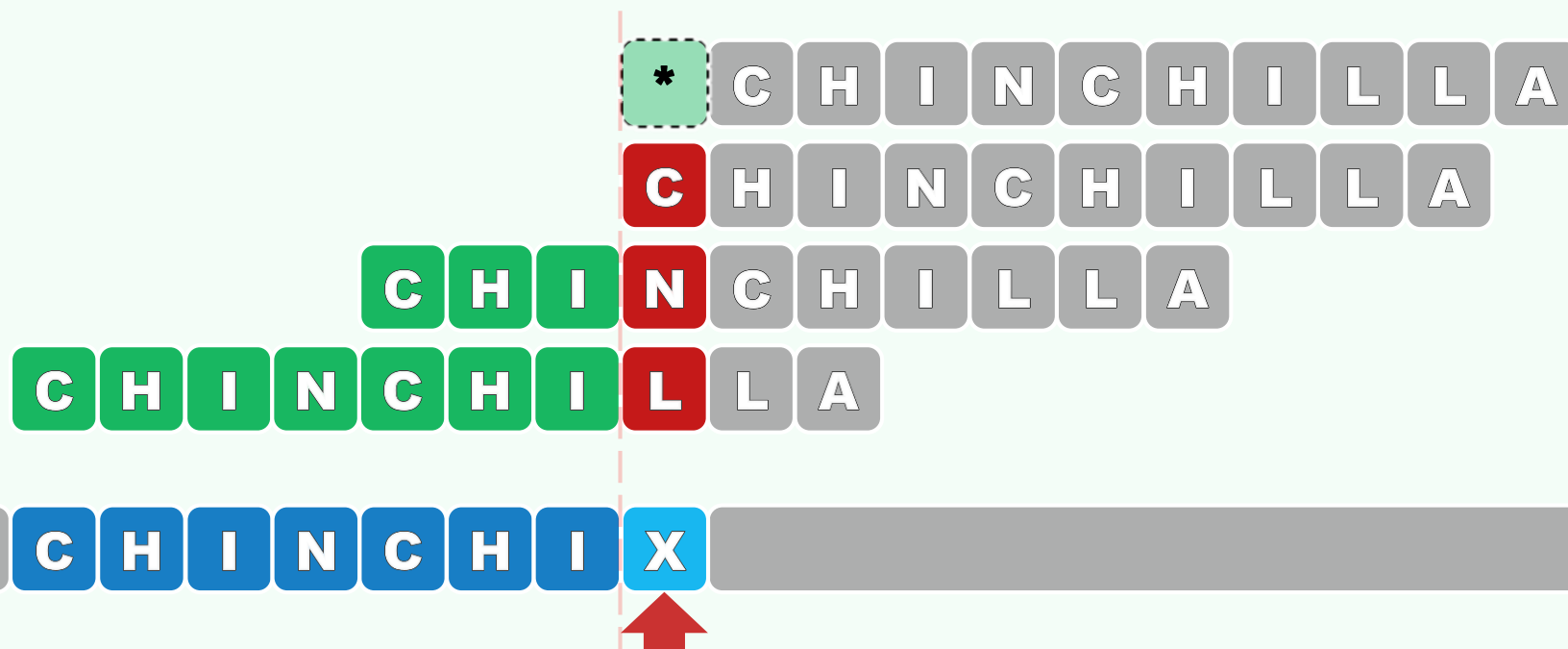
❖ 构造查询表 $next[0, m)$ , 做好预案

❖ 一旦在 $P[j]$ 处失配, 只需

将j替换为 $next[j]$ , 继续与 $T[i]$ 比对



# 实例



# KMP算法

```
int match( char * P, char * T ) {  
    int * next = buildNext(P);  
    int n = (int) strlen(T), i = 0;  
    int m = (int) strlen(P), j = 0;  
    while ( (j < m) && (i < n) ) //可优化  
        if ( 0 > j || T[i] == P[j] ) {  
            i++; j++;  
        } else  
            j = next[j];  
    delete [] next;  
    return i - j;  
}
```



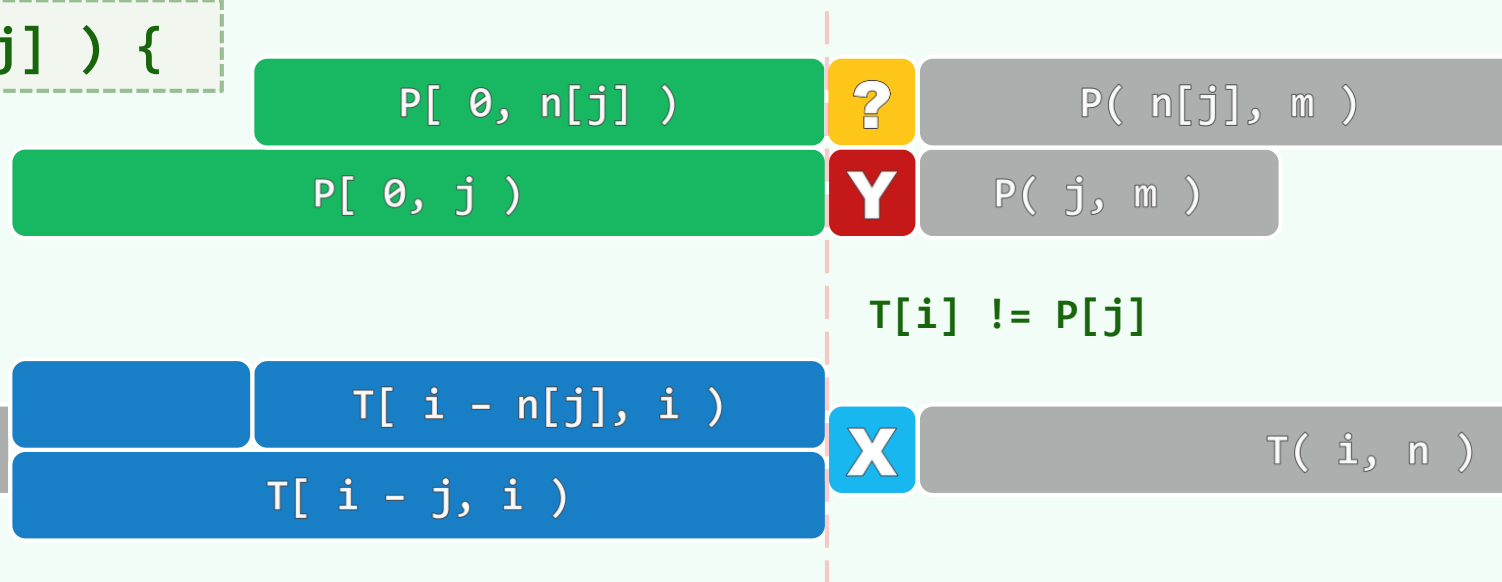
D. E. **K**nuth



J. H. **M**orris



V. R. **P**ratt



## 快速右移 + 绝不回退

