

串

KMP算法：分摊分析

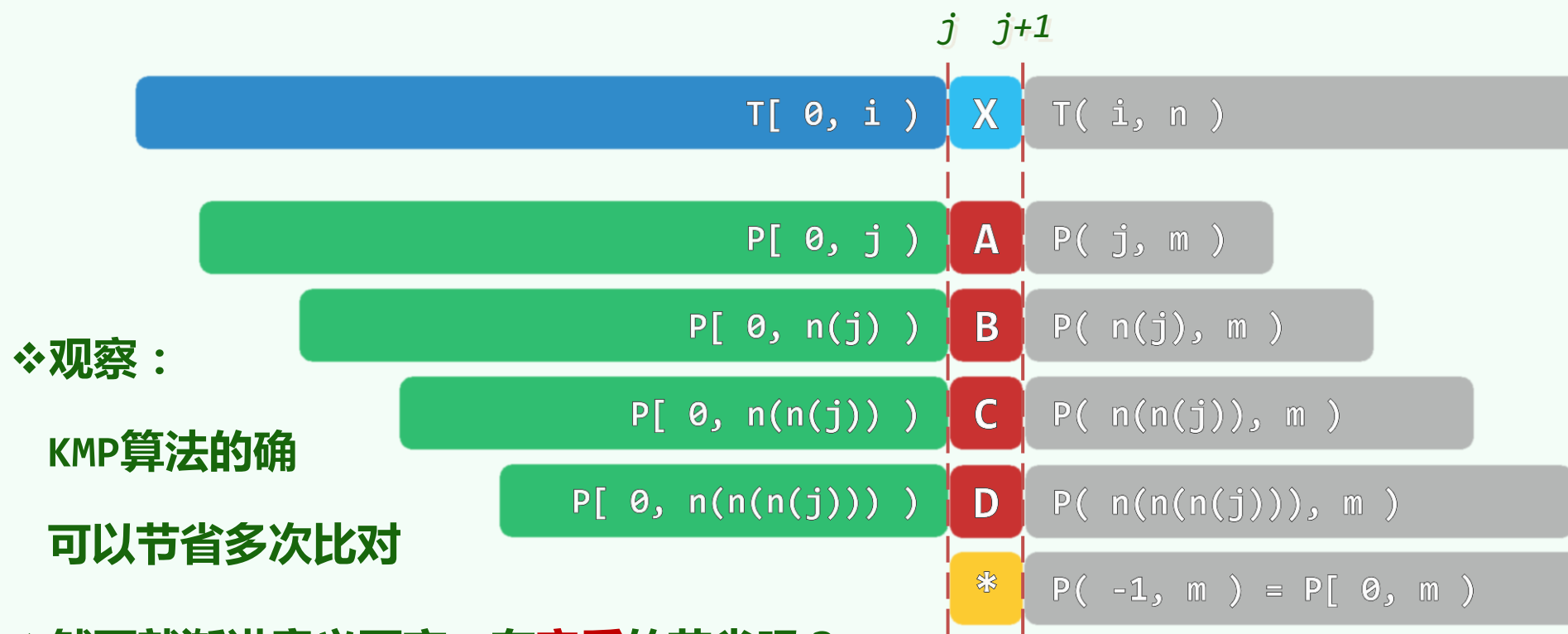
13-C5

邓俊辉

deng@tsinghua.edu.cn

失之东隅，收之桑榆

$\Omega(n * m)$  ?



❖观察：

KMP算法的确

可以节省多次比对

❖然而就渐进意义而言，有**实质**的节省吗？

❖在每一 $T[i]$ 处， $P$ 都可能比对 $\Omega(m)$ 次

❖于是， 倘若有 $\Omega(n)$ 个 $T[i]$ 如此...

❖然而，更细致的分析将表明

即便是最坏情况，也不过 $\mathcal{O}(n)$ 时间

❖同理，建立 $next[]$ 也只需 $\mathcal{O}(m)$ 时间

$$O(n + m) !$$

❖ 令 :  $k = 2*i - j$  //具体含义 , 详见习题[11-4]

while (  $j < m \ \&\& \ i < n$  ) //k必随迭代而单调递增 , 故也是迭代步数的上界

if (  $0 > j \ || \ T[i] == P[j]$  )

{  $i++;$   $j++;$  } //k恰好加1

else

$j = \text{next}[j];$  //k至少加1

❖ 初始  $k = 0$

算法结束时 , 必有 :  $k = 2*i - j \leq 2(n - 1) - (-1) = 2n - 1$