13-C5

串

KMP算法：分摊分析

邓俊辉

deng@tsinghua.edu.cn

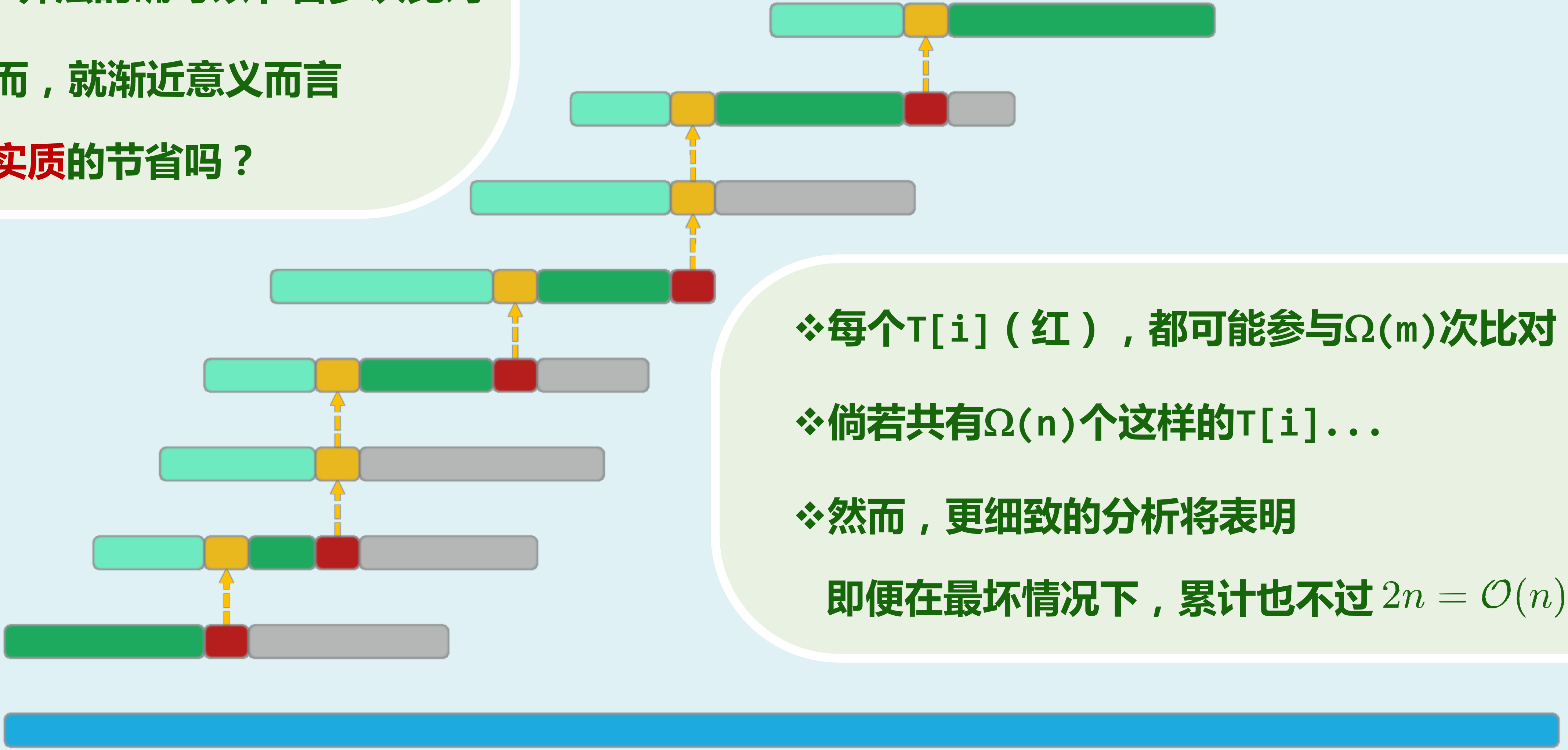
失之东隅，收之桑榆

Ω(n*m)？

❖KMP算法的确可以节省多次比对

❖然而，就渐近意义而言

有实质的节省吗？



❖每个T[i]（红），都可能参与Ω(m)次比对（黄）

❖倘若共有Ω(n)个这样的T[i]...

❖然而，更细致的分析将表明

即便在最坏情况下，累计也不过 $2n = \mathcal{O}(n)$ 次

$\mathcal{O}(n + m)$!

❖ 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 = next[j]; //k至少加1
```

❖ 初始k = 0

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