

## 11. 串

### (c1) KMP算法：从记忆力到预知力

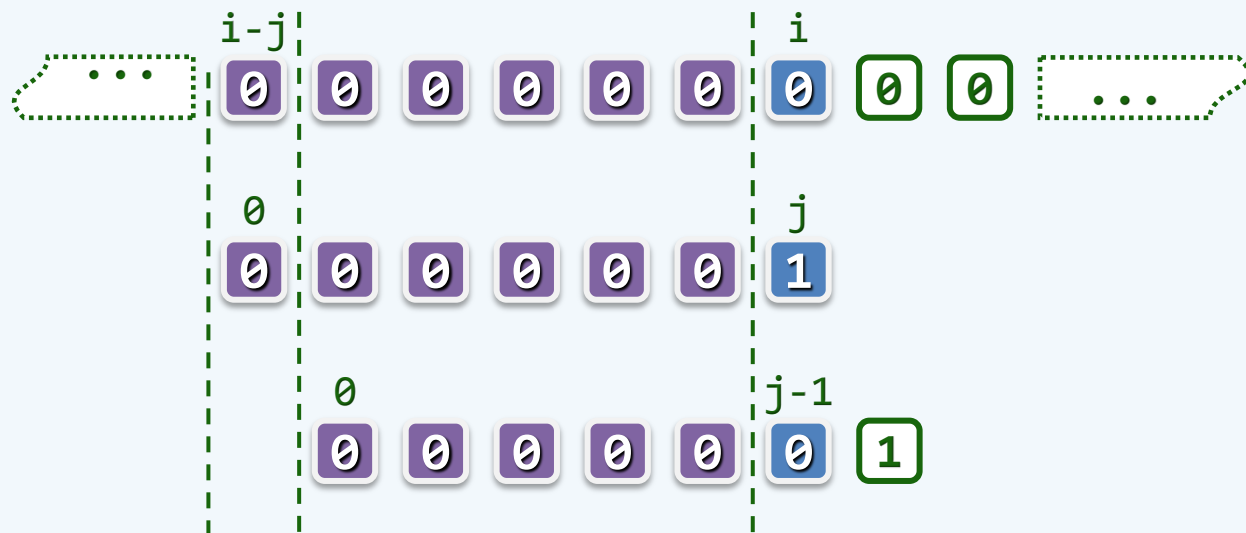
邓俊辉

知易者不占，善易者不卜

deng@tsinghua.edu.cn

## 蛮力，为何低效

❖ T回退、P复位之后，此前比对过的字符，将再次参与比对



❖ 最坏情况下，T/P中每个字符平均参加 $m/n$ 次比对——累计 $O(m*n)$ 次

❖ 于是，只要局部匹配很多，效率必将很低

❖ 其实，这类比对大多是**不必要**的，因为...

## 无论如何，还是不变性

$$\diamond T[i - j, i] == P[0, j]$$

$$T[i] == P[j]$$

?

$T[i - j, i]$

\*

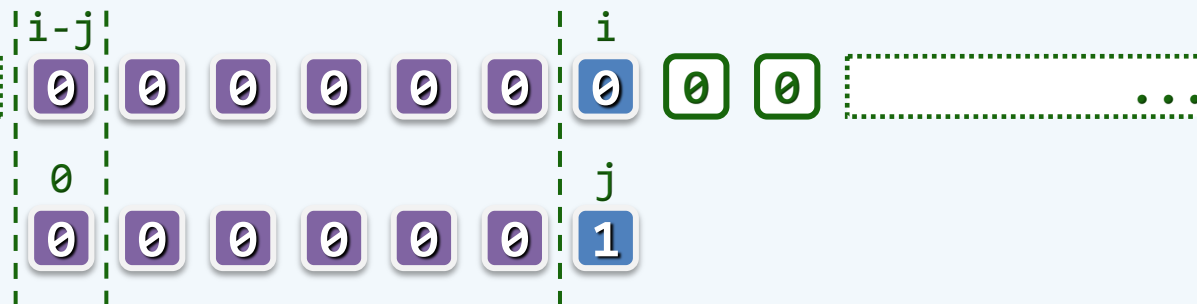
$T(i, n)$

$P[0, j]$

\*

$P(j, m)$

❖ 亦即，我们业已掌握  $T[i - j, i]$  的**全部信息**——其中的字符**各是什么**

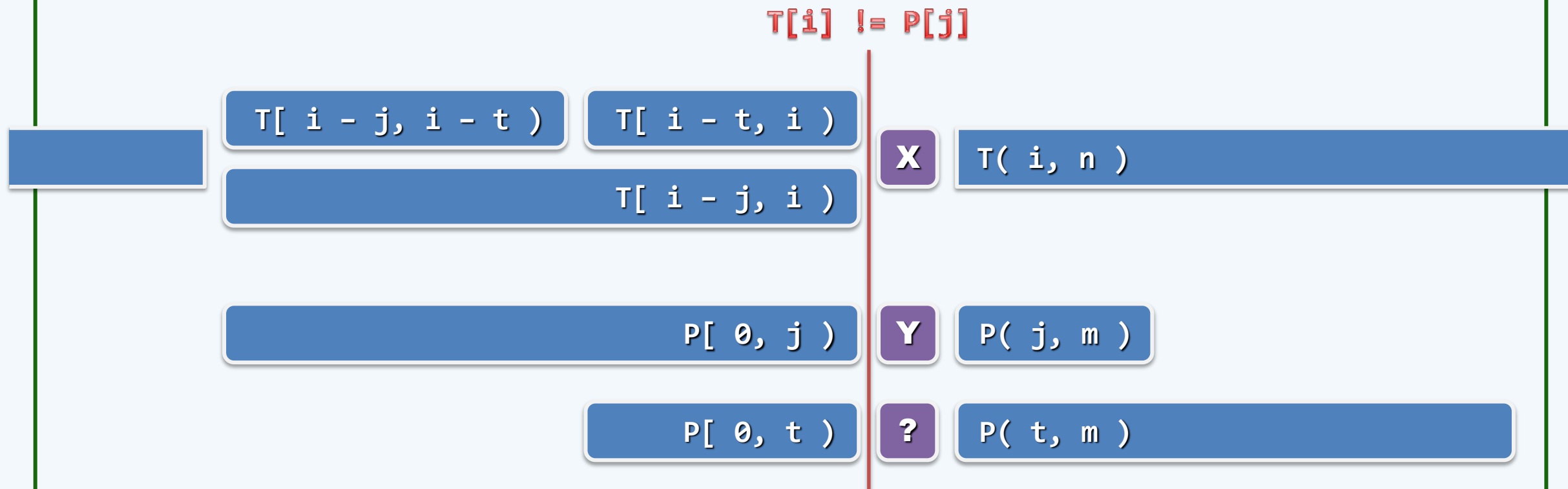


❖ 既如此...

## 只要记忆力足够强

❖ 在失败之后的下一轮比对中...

❖  $T[i-j, i]$  就不必再次接受比对，而是可以直接地...

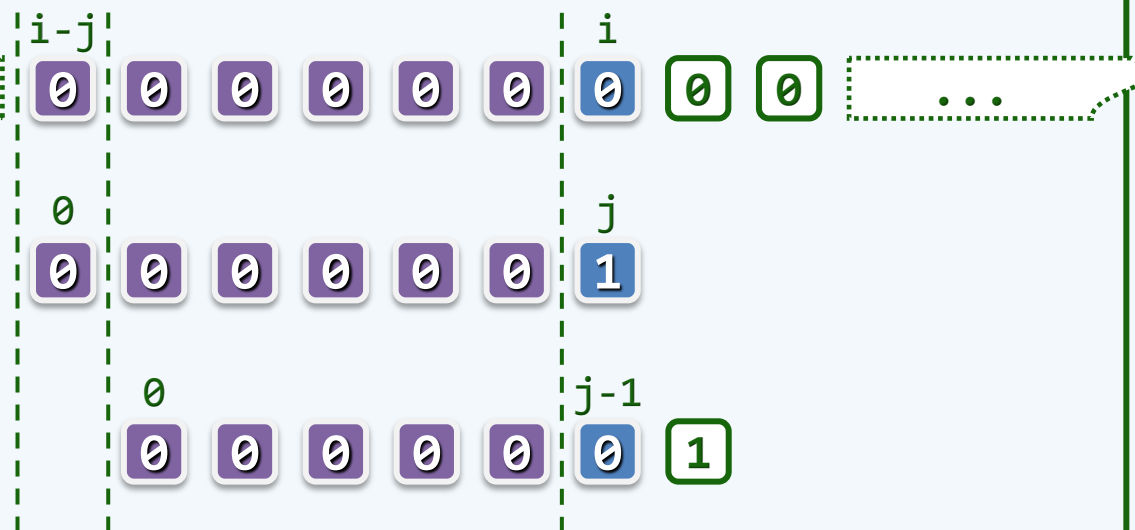


## 将记忆力，转化为预知力

❖ 如此， $i$ 将完全不必回退！

- 比对成功，则与 $j$ 同步前进一个字符

- 否则， $j$ 更新为某个更小的 $t$ ，并继续比对



❖ 即便是更为复杂的情况，依然可行

❖ 优化 = P可快速右移 + 避免重复比对

❖ 为确定 $t$ ，需花费多少时间和空间？



更重要地，可否在事先就确定？