

11. 串

(d2) BM_BC算法：坏字符

墩儿了个墩儿啊

怎么这么厚啊？

后娘当的呀

五六毛啊

我给你两块二呀

我给你三块八呀

我给你一百八十六块三毛二分五哇

大了棉地墩儿啊

亲娘絮的呀

当多少钱啊？

一块一啊

我找你一块一呀

我找你两块七呀

我留下一块一啊，剩下全给你呀！

邓俊辉

deng@tsinghua.edu.cn

Bad-Character

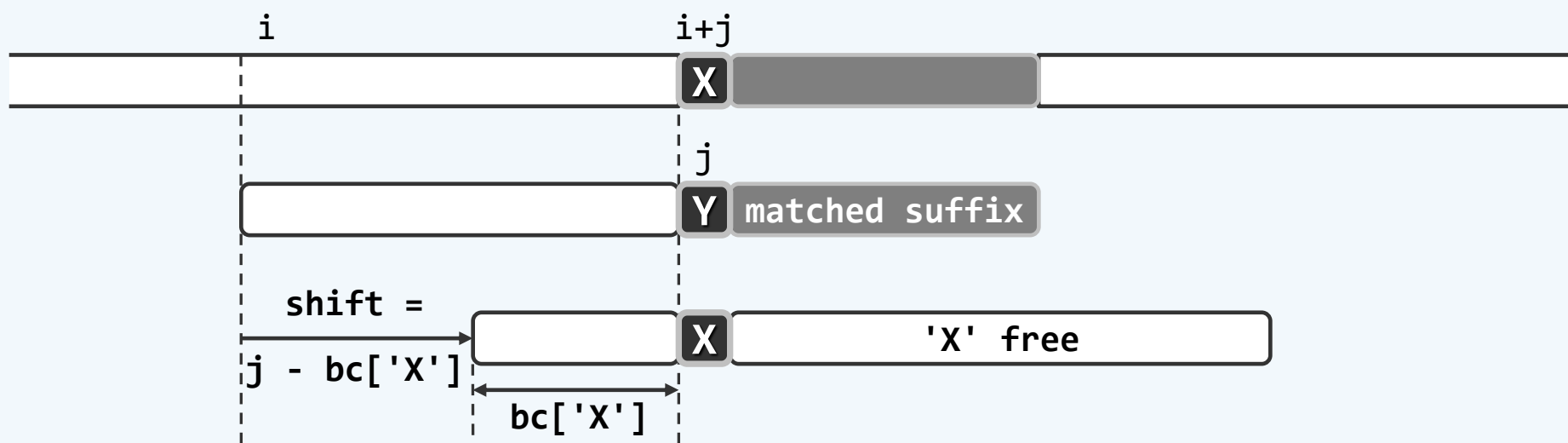
❖ 某趟扫描中，一旦发现

$$T[i + j] = \boxed{X} \neq \boxed{Y} = P[j]$$

// \boxed{Y} 称作坏字符

则 P 相应地右移，并启动新一轮扫描比对

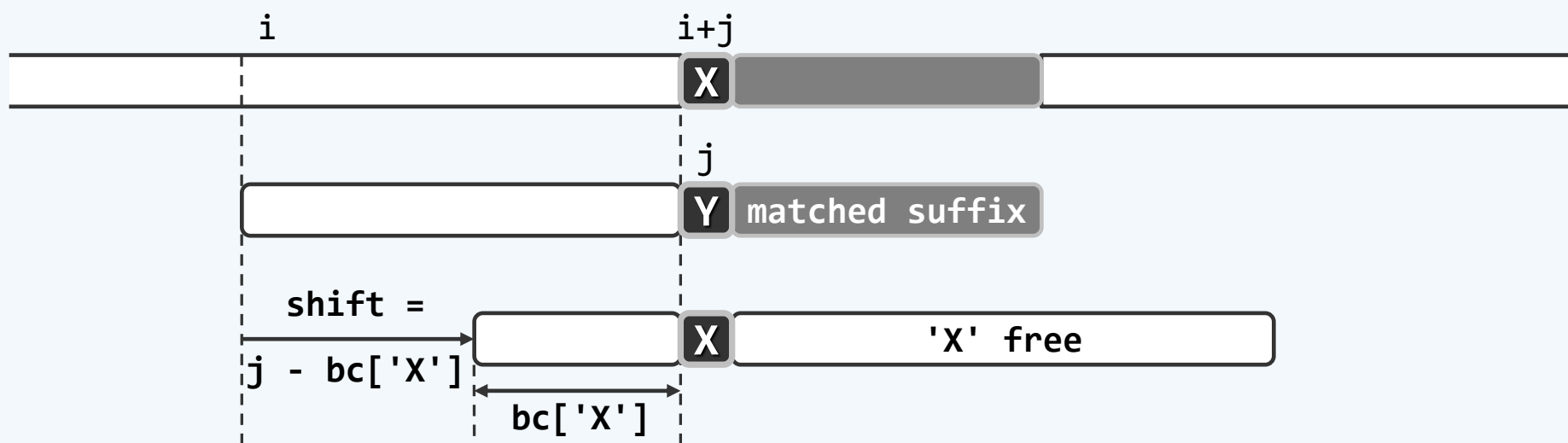
// 具体如何右移？



❖ 必要条件：至少坏字符本身应得以恢复匹配——因此...

Bad-Character Shift

❖ 只需：找出P中的 X ，使之与 $T[i + j] = X$ 对准，并做一轮比对



❖ 注意：位移量取决于失配位置 j ，以及 X 在P中的秩，而与T和i无关！

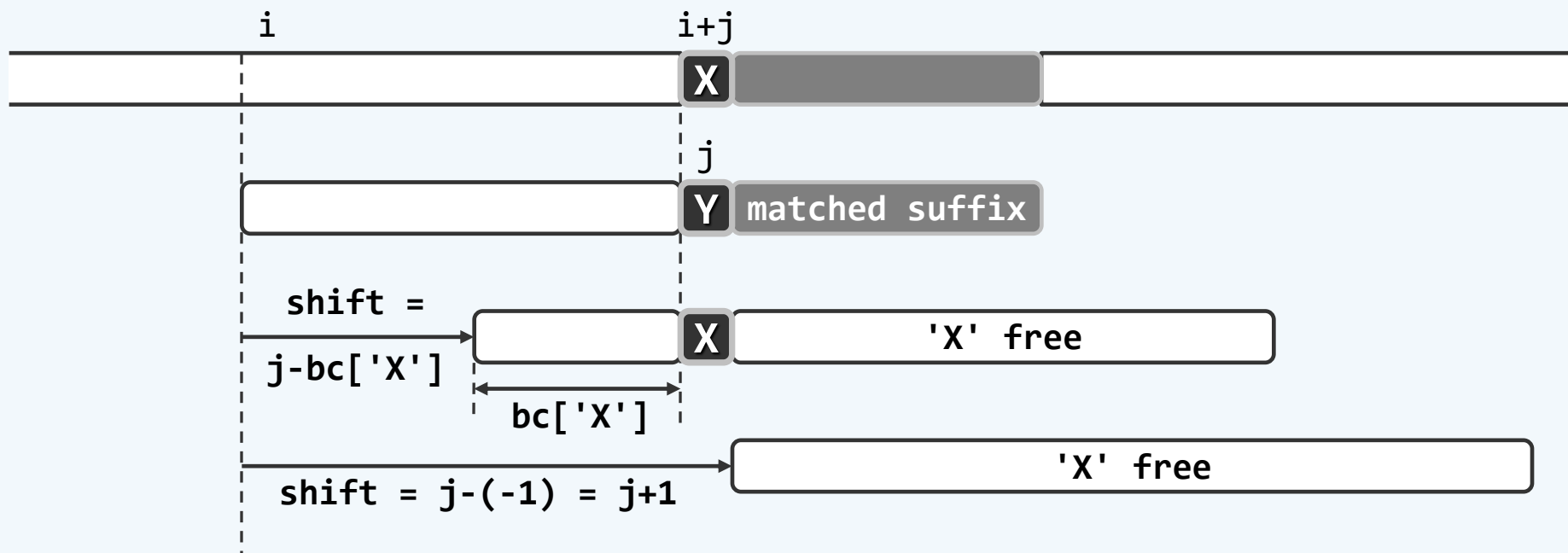
//与KMP类似

❖ 若：令 $bc[X] = \text{rank}[X] = j - \text{shift}$

则： $bc[]$ 总计有 $s = |\Sigma|$ 项，且可事先计算，并制表待查

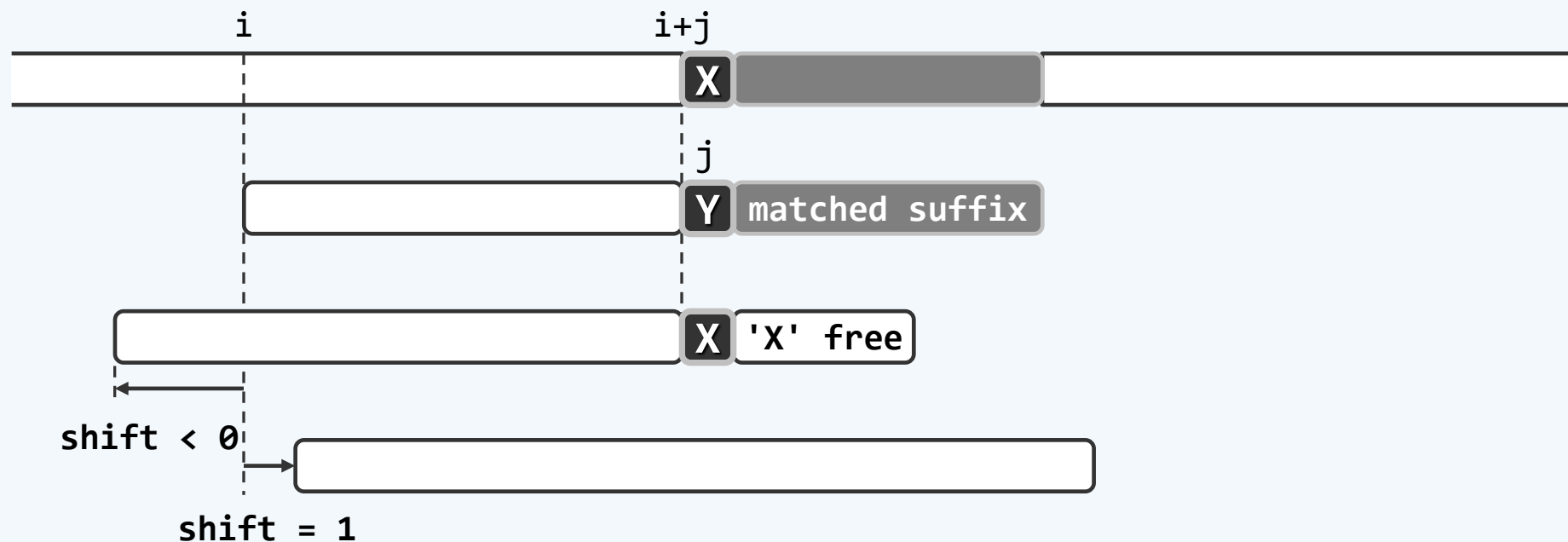
Bad-Character Shift

- ❖ 若P包含多个X，取其中最靠右者 //如此可确保无需回溯
- ❖ 若P不含任何X，将P整体移过T[i + j] //与KMP类似，这等效于，与假想的通配符对齐



Bad-Character Shift

- ❖ 即使 'X' 在 P 中出现，但最右侧者过于靠右（以至于位移量为负）呢？
- ❖ 将 P[] 右移一个字符！
- ❖ 为何...不选用在 P[j] 左侧并与之最近的那个 X？



实例