

11. 串

(d4) BM_BC算法：性能分析

邓俊辉

deng@tsinghua.edu.cn

最好情况

❖ $O(n / m)$ —— 除法？没错！

❖ 比如：T =

x	x	x	x	1
---	---	---	---	---

x	x	x	x	1
---	---	---	---	---

x	x	x	x	1
---	---	---	---	---

 . . .

x	x	x	x	1
---	---	---	---	---

x	x	x	x	1
---	---	---	---	---

P =

0	0	0	0	0
---	---	---	---	---

❖ 一般地：只要P不含T[i+j]，即可直接移动m个字符

仅需单次比较，即可排除m个对齐位置

❖ 单次匹配概率越小，性能优势越明显

// 大字母表：ASCII + UniCode

❖ P越长，这类移动的效果越明显

最差情况

❖ 最坏 = $O(n \times m)$ —— 退化为蛮力算法？是的！

❖ 比如：T =

0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	-----	---	---	---	---	---	---	---

P =

1	0	0	0	0
---	---	---	---	---

❖ 每轮迭代，都要在扫过整个P之后，方能确定右移一个字符

此时，须经m次比较，方能排除单个对齐位置

❖ 单次匹配概率越大的场合，性能越接近于蛮力算法

// 小字母表Bitmap + DNA

❖ 反思：借助以上bc表，仅仅利用了失配比对提供的信息（教训）！

类比：可否仿照KMP，同时利用起匹配比对提供的信息（经验）？