String Matching

- text editor
- dna sequencing $3 \cdot 10^9$
- web search

T: acabaabaaccaabaac

P: abaa

$\Sigma = \{a, b, c\}$

Σ^* all strings over Σ

$w \sqsubset x$ = w is a prefix of x

$w \sqsupset x$ = w is a suffix of x

$ab \sqsubset abcb$

$ba \sqsupset caba$

$P[j:k] = P[j]\,P[j+1] \ldots P[k]$

$P[:k] = P[1:k$ $\quad P[:0] = \{$

Lemma 32.1

Naive string matcher (T, P, n, m)

```
for s = 0 to n-m                      (n-m)
    if P[1:m] == T[s+1:s+m]           (m)
        print s
```

$\Theta((n-m+1)\cdot m) =$
$\Theta(n \cdot m$

Rabin-Karp

$\Sigma = \{a, b, c, d, e, f, g, h, i, j\}$
$0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9$

T: c c a d b e b f c b
2 2 0 3 1 4 1 4 2 1

P: d b e b f
3 1 4 1 5

3 1 4 1 5

$\Theta(m) + \Theta(n - m + 1)$

Trick:

$10^4 = 10^{m-1}$

$22031 \cdot 10^4 =$
$- \ 20000$
10×2031
20310
$+ \quad 4$
20314

4 ops

Heuristic

$d = |\Sigma|$

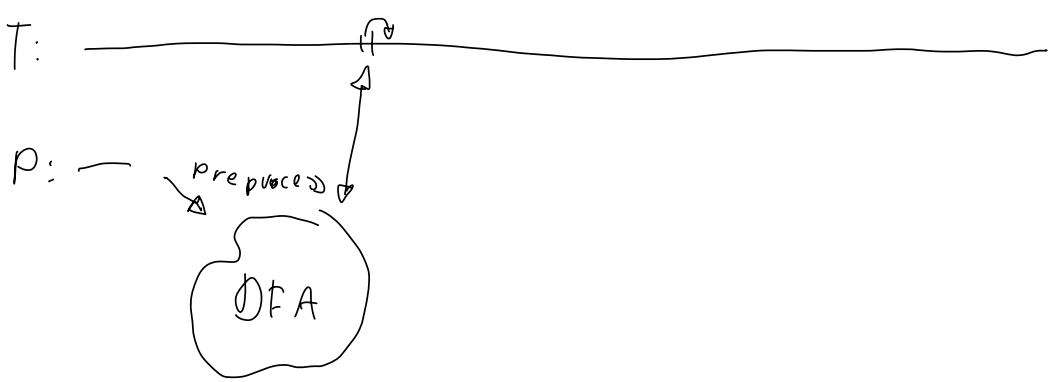
$t_{s+1} = (t_s - T[s+1] \times d^{m-1} + T[s+m+1]) \bmod q$

$t_s \to t_{s+1}$
d

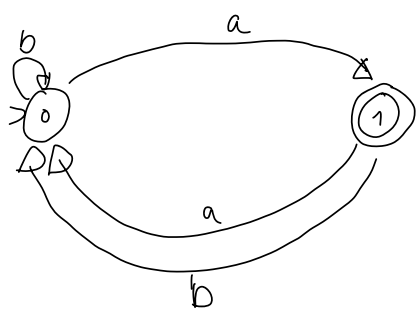
T[s+1 : s+m

(choose prime q as large as possible
so that $q \cdot d$ fits in 1 word

String Matching with a finite automata



DFA $M = (Q, q_0, A, \Sigma, \delta)$

Q finite set of states 

$= \{0, 1\}$

$q_0 \in Q$ start state $= 0$

$A \subseteq Q$ accepting state $\{1\}$

Σ finite alphabet $\Sigma = \{a, b\}$

$\delta: Q \times \Sigma \to Q$ transition function

	δ	Σ: a	b
Q	0	1	0
	1	0	0

Suffix function

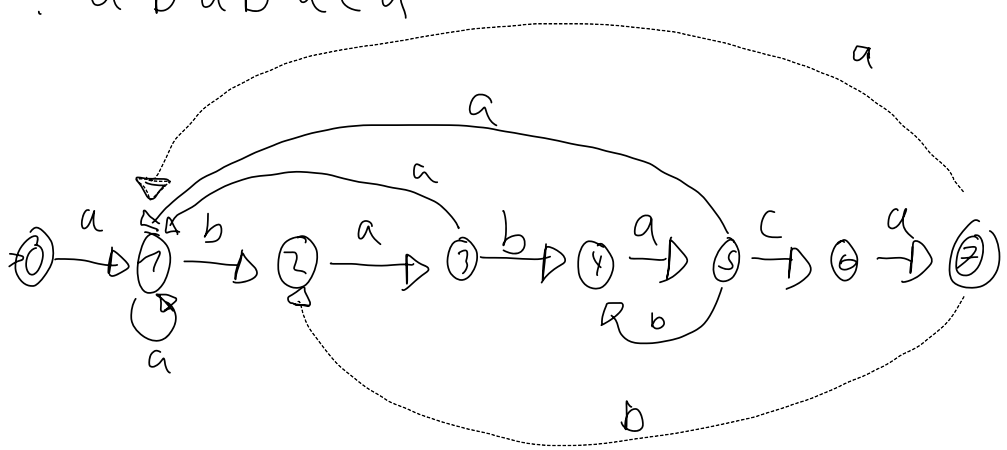
$\sigma : \Sigma^* \to \{0, 1, \ldots, m\}$

$\sigma(x) = \max \{ k \mid P[:k] \sqsupset x \}$

$\delta(q, a) = \sigma(P[:q]\, a)$

P: a b a b a c a

missing arrows are going to zero all states need an arrow



$$\Gamma(x) = \max\{k \mid P[:k] \supset x\}$$

$$\delta(q, a) = \Gamma(P[:q]\,a)$$

$$\delta(3, b) = \Gamma(abab) = 4$$

$$\delta(3, a) = \Gamma(abaa) = 1$$

$$\delta(3, c) = \Gamma(abac) = 0$$

$$\delta(5, b) = \Gamma(ababab) = 4$$

$P[:k] = \varepsilon$

Compute-Transition-Function(P, Σ, m)

```
for q = 0 to m                                   → δ(q, a)
    for each a ∈ Σ
        k = min{m, q+1}
        while P[:k] ≠ P[:q] a                    (≠ : m)
            k = k - 1
        δ(q, a) = k
```

m · q · |Σ| · m

$O(m^3 \cdot |\Sigma|)$

Knuth-Morris-Pratt

Preprocess the pattern

- not δ in $O(m^3 \cdot |\Sigma|)$
  instead ignore Σ, compute π in $O(m)$

Prefix function

$\delta(q, a)$ - compute in amortized constant time via π giving $O(n)$

T: b a c b a b a b a a b c b a b

P: a b a b a c a

a b a b a c a

a b a . . . .

$\pi[q] = \max\{k \mid k < q \text{ and } P[:k] \supset P[:q]\}$

i	1	2	3	4	5	6	7
P[i]	a	b	a	b	a	c	a
π[i]	0	0	1	2	3	0	1

KMP-Matcher(T, P, n, m) put code

Compute-Prefix-Function(P, m) put code

DFA vs. KMP