

Problem String matching

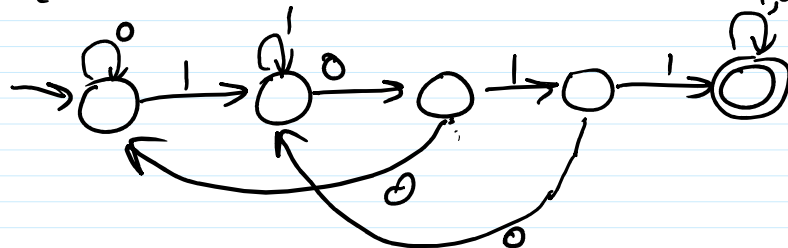
Given strings $u = a_1 a_2 \dots a_n \in \{0,1\}^*$ "text"
 $v = b_1 b_2 \dots b_m \in \{0,1\}^*$ "pattern"
 $(m < n)$
 is v a substring of u ?

e.g. $u = 01101011010$
 $v = 10110$

brute force: $O(mn)$ time

DFA method: $O(f(m) + n)$ time

$L = \{x : x \text{ contains } 011 \text{ as substring}\}$



011010...

$f(m) = O(m|\Sigma|)$. if done carefully

Knuth-Morris-Pratt '77: $O(n)$ time

using a "compressed" form of DFA
 messy.

today: simpler $O(n)$ time randomized

Digression...

Subproblem

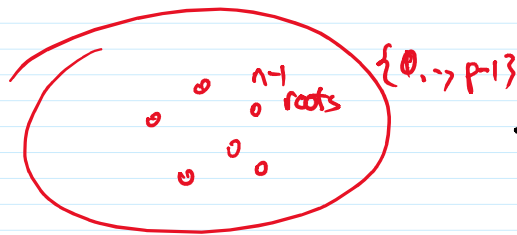
Alice has a string $u \in \{0,1\}^*$

Subproblem

$$\Leftrightarrow \sum_{i=0}^{n-1} (a_i - b_i) x^i \equiv 0 \pmod{p}$$

Counting Lemma A nonzero polynomial of $\deg \leq n-1$ has $\leq \underline{n-1}$ roots, mod p .
 $\nwarrow$ prime

$$\Rightarrow \Pr[\text{algm errs}] \leq \frac{n-1}{p}$$



pick $p \approx n^{d+1}$

$$\Rightarrow \text{prob err} \leq O\left(\frac{1}{n^d}\right)$$

bits transmitted $\log p = O(d \log n)$

Back to string matching problem ...

¹⁷⁷
Rabin-Karp Rand. Alg'm (Monte Carlo)

$$x = \text{rand}(0, p-1)$$

$$A = F_x(a_1 a_2 \dots a_m)$$

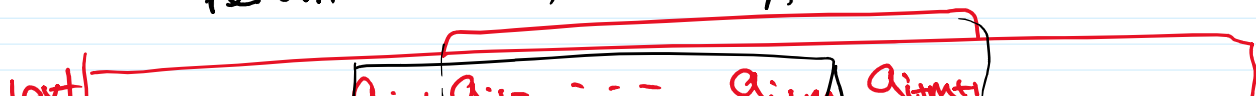
$$B = F_x(b_1 b_2 \dots b_m)$$

for $i = 0$ to $n-m$ do {

if $A = B$ return "possible match at i "

$$A = (A x + a_{i+m+1} - a_{i+1} x^m) \pmod{p}$$

}
 return "definitely no match"



$$O(n^2) \text{ time prob } O\left(\frac{1}{nd}\right)$$

$$\text{expected runtime} = O\left(n \cdot \left(1 - \frac{1}{nd}\right) + n^2 \cdot \frac{1}{nd}\right)$$

$$d=2$$

$$\Rightarrow \boxed{O(n)}$$