

CS7800: Advanced Algorithms

Class 23 : Randomized Algorithms IV

- Pattern matching

Jonathan Ullman

December 2, 2025

Pattern Matching

e.g. $\Sigma = \{0, 1\}$ or $\{A, B, \dots, Z\}$

Input: A string $s = s_{n-1} \dots s_0 \in \overset{\sim}{\Sigma}^n$

A pattern $t = t_{m-1} \dots t_0 \in \overset{\sim}{\Sigma}^m$ for $1 \leq m \leq n$

Output: Either i such that $s_i \dots s_{i+m-1} = t_{m-1} \dots t_0$
or \emptyset if there is no match

$s = 101 \underbrace{1001}_{i=3}$
 $t = \underline{100}$

output \emptyset
 $s = 10101010$
 $t = 11$

First Attempt

Input: $s \in \Sigma^n$ $t \in \Sigma^m$

For $i = n-1, \dots, m$

Counting down is useful later

$n-m$ iterations

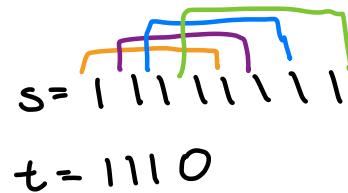
If $s_{i-m+j} = t_j$ for all $j = 0, 1, \dots, m-1$:

Return i

1 operation per symbol

m symbols

Return \emptyset



What is the running time?

$O(nm)$ in the worst case (quadratic time)

$O((n-m)m)$

Strings to Numbers

- Can assume $\Sigma = \{0, 1\}$ for simplicity
 - Everything gets written in binary at some level anyway
- A string $s_{n-1} \dots s_0 \in \{0, 1\}^n$ is also an n-digit number

$$\begin{array}{ccccc} s_5 & \dots & s_1 & s_0 \\ \boxed{1 | 0 | 0 | 1 | 0 | 1} & = & s_5 \times 2^5 + s_4 \times 2^4 + s_3 \times 2^3 + s_2 \times 2^2 + s_1 \times 2^1 + s_0 \times 2^0 \\ & = & 32 + 0 + 0 + 4 + 0 + 1 \\ & = & 37 \end{array}$$

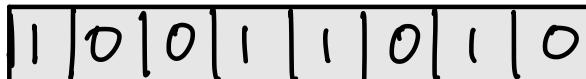
Strings to Numbers

- Can go from one substring to the next easily

$$n=8$$

$$m=3$$

$s_7 \dots s_2 \ s_1 \ s_0$



Notice funny notation

$$[s_2 s_1 s_0] = s_2 \times 2^2 + s_1 \times 2^1 + s_0 \times 2^0$$

$[s_3 s_2 s_1] = s_3 \times 2^2 + s_2 \times 2^1 + s_1 \times 2^0$

$$[s_2 s_1 s_0] = ([s_3 s_2 s_1] - \underline{\underline{s_3 \times 2^2}}) \times 2 + \underline{\underline{s_0}}$$

Three steps to slide
the window over

Second Attempt

Input: $s \in \Sigma^n$ $t = \Sigma^m$

$w = [s_{n-1} s_{n-2} \dots s_{n-m}]$

$t = [t_{m-1} t_{m-2} \dots t_0]$

For $i = n-1, \dots, m$

If $w = t$ return i

$w \leftarrow (w - s_i \times 2^m) \times 2 + s_{i-m}$

Return \emptyset

Time $O(m)$

Equal as numbers

Time $O(m)$ because really you're comparing strings

Time $O(1)$ using fast bit shifts

slide window over one position

w goes from $[s_i \dots s_{i-m+1}]$
to $[s_{i-1} \dots s_{i-m}]$

What is the running time?
 $O(nm)$

Aside: Randomized Fingerprints

- Can we use hashing to make comparison faster?

$$h: \{0,1\}^m \rightarrow \{0,1, \dots, B-1\} \quad (\text{h from a universal hash})$$

$$x, y \in \{0,1\}^m \text{ and } x \neq y$$

$$\underset{h}{P}(h(x) = h(y)) = \frac{1}{B}$$

If $x = y$, then $h(x) = h(y)$

If $x \neq y$, then $\underset{h}{P}(h(x) = h(y)) \leq \frac{1}{B}$

Only have to check $\frac{1}{B}$ fraction of non matching windows

Aside: Randomized Fingerprints

- Suppose we pick a random prime number p with k bits
 x and y are m -bit numbers and $x \neq y$

What is $\underset{P}{\mathbb{P}}(\underbrace{x = y \bmod p}_{\text{happens if } x-y \text{ divisible by } p})$

Random Prime Numbers

① (Prime Number Theorem) The number of primes with at most k bits (i.e. $\leq 2^k - 1$) is $\Theta\left(\frac{2^k}{k}\right)$

Hard to Prove

② An m -bit integer has at most m distinct prime factors

$$2^m \geq x = p_1^{a_1} \times p_2^{a_2} \times \dots \times p_f^{a_f} \geq 2^{\textcircled{f}} \stackrel{\text{\# of distinct factors}}{\Rightarrow f \leq m}$$

③ There is an efficient randomized primality test

Hard to Prove

Aside: Randomized Fingerprints

- Suppose we pick a random prime number p with k bits
 x and y are m -bit numbers and $x \neq y$

What is $\underset{P}{\mathbb{P}}(x \equiv y \pmod{p})$

happens if $x-y$ divisible by p

$$\mathbb{P}(x-y=0 \pmod{p}) \leq \frac{m}{\left(\frac{2^k}{\pi}\right)} \quad \begin{array}{l} \text{Number of distinct prime factors} \\ \text{Number of } k\text{-bit primes} \end{array}$$

$$\mathbb{P}_p(x=y \pmod{p}) \leq \frac{m \cdot k}{2^k} \quad \text{if } k \approx 2 \log_2 m \text{ then } \mathbb{P} \leq \frac{2m \cdot \log_2 m}{m^2} = \frac{2 \log_2 m}{m}$$

Randomized String Matching

Input: $s \in \Sigma^n$ $t = \Sigma^m$

$k \leq m$
set later

What is the running time?

Let p be a random k -bit prime $\{$ poly in k , poly in $\log m$

$$\sigma = 2^m \bmod p \quad \{ \text{Time } O(m)$$

$$w = [s_{n-1} s_{n-2} \dots s_{n-m}] \bmod p \quad \{ \text{Time } O(m)$$

$$t = [t_{m-1} t_{m-2} \dots t_0] \bmod p \quad \{ \text{Time } O(m)$$

For $i = n-1, \dots, m$

If $w = t \bmod p : \{ \text{Time } O(k)$

$$\{ [s_i \dots s_{i-m+1}] = [t_{m-1} \dots t_0]$$

Check for false match

return $i \{ \text{Time } O(m) \text{ if } \{ \text{have to}$

$$w \leftarrow (w - s_i \times \sigma) \times 2 + s_{i-m} \bmod p \quad \{ \text{Time } O(k)$$

Return \emptyset

Randomized String Matching

Input: $s \in \Sigma^n$ $t = \Sigma^m$ $k \leq m$
set later

Let p be a random k -bit prime

$$\sigma = 2^m \bmod p$$

$$w = [s_{n-1} s_{n-2} \dots s_{n-m}] \bmod p$$

$$t = [t_{m-1} t_{m-2} \dots t_0] \bmod p$$

For $i = n-1, \dots, m$

If $w = t \bmod p$:

(Check for
false match)

L If $w = t$: return i

$$w \leftarrow (w - s_i \times \sigma) \times 2 + s_{i-m} \bmod p$$

Return \emptyset

What is the running time?

$$\mathbb{E} \left(\underbrace{\mathcal{O}(m)}_{\text{Initialize}} + \underbrace{\mathcal{O}(nk)}_{\text{Mandatory part of loop}} + \mathcal{O}(m) \cdot \underbrace{\# \text{ of false matches}}_{\mathcal{O}(\frac{n}{m} \log m)} \right)$$

$$\mathbb{E} (\# \text{ of false matches})$$

$$\lesssim n \cdot \frac{m \cdot k}{2^k}$$

$$\text{set } k = 2 \log_2 m$$

$$\lesssim \frac{n \cdot m \cdot 2 \log_2 m}{m^2}$$

$$= 2 \cdot \frac{n \cdot \log_2 m}{m}$$