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 $s = s_{n-1} \cdots s_0 \in \Sigma^n$

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

Output: Either i such that $s_i \cdots s_{i-m+1} = t_{m-1} \cdots t_0$

or $\emptyset$ if there is no match

$i = 3$

$s = 1011\,\overbrace{100}\,1$

$t = \underbrace{100}$

output $\emptyset$

$s = 10101010$

$t = 111$

First Attempt

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

For $i = n-1, \ldots, m$ — Counting down is useful later — n−m iterations

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

    Return i

Return $\emptyset$

1 operation per symbol

m symbols

$s = 111111111$

$t = 1110$

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 $s_{n-1}\cdots s_0 \in \{0,1\}^n$ is also an n-digit number

$$\begin{array}{cccccc} s_5 & & \cdots & & s_1 & s_0 \\ \hline \multicolumn{1}{|c|}{1} & \multicolumn{1}{c|}{0} & \multicolumn{1}{c|}{0} & \multicolumn{1}{c|}{1} & \multicolumn{1}{c|}{0} & \multicolumn{1}{c|}{1} \\ \hline \end{array} = s_5 \times 2^5 + s_4 \times 2^4 + s_3 \times 2^3 + s_2 \times 2^2 + s_1 \times 2^1 + s_0 \times 2^0$$

$$= 32 + 0 + 0 + 4 + 0 + 1$$

$$= 37$$

Strings to Numbers

- Can go from one substring to the next easily

$n = 8$

$m = 3$

s_7	...				s_2	s_1	s_0
1	0	0	1	1	0	1	0

Notice funny notation

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

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

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

Three steps to slide the window over

Second Attempt

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

Time $O(m)$

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

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

Equal as numbers

What is the running time?

$O(nm)$

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

If $w = t$ return i } Time $O(m)$ because really you're comparing strings

$w \leftarrow \underbrace{(w - s_i \times 2^m) \times 2 + s_{i-m}}$ } Time $O(1)$ using fast bit shifts

Return $\emptyset$

slide window over one position

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

to $[s_{i-1} \dots s_{i-m}]$

Aside: Randomized Fingerprints

- Can we use hashing to make comparison faster?

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

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

$$\mathbb{P}_h(h(x)=h(y)) = 1/B$$

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

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

Only have to check 1/B fraction of non matching windows

Aside: Randomized Fingerprints

- Suppose we pick a <u>random</u> prime number p with k bits

  x and y are m-bit numbers and $x \neq y$

What is $\underset{p}{\mathbb{P}}\left(\underbrace{x = y \bmod p}_{}\right)$

happens if $x-y$ 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

$$2^m \geq x = p_1^{a_1} \times p_2^{a_2} \times \ldots \times p_f^{a_f} \geq 2^{f}$$

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

$$\mathbb{P}(x-y = 0 \bmod p) \lesssim \frac{m}{\left(\frac{2^k}{k}\right)}$$

$\leftarrow$ Number of distinct prime factors

$\leftarrow$ Number of k-bit primes

$$\underset{p}{\mathbb{P}}(x=y \bmod p) \lesssim \frac{m \cdot k}{2^k}$$

if $k \approx 2\log_2 m$ then $\mathbb{P} \lesssim \frac{2m \cdot \log_2 m}{m^2}$

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

Randomized String Matching

What is the running time?

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

$k \ll m$
set later

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

$\sigma = 2^m \bmod p \;\}$ Time $O(m)$

$$\left.\begin{array}{l} w = [s_{n-1} s_{n-2} \cdots s_{n-m}] \bmod p \\ t = [t_{m-1} t_{m-2} \cdots t_0] \bmod p \end{array}\right\} \text{Time } O(m)$$

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

- If $w = t \bmod p$: $\}$ Time $O(k)$
  - $[s_i \cdots s_{i-m+1}] = [t_{m-1} \cdots t_0]$ return $i \;\}$ Time $O(m)$ if I have to

  Check for false match
- $w \leftarrow (w - s_i \times \sigma) \times 2 + s_{i-m} \bmod p \;\}$ Time $O(k)$

Return $\emptyset$

Randomized String Matching

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

$k \ll m$ 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 $w = t \bmod p$:
  - If $w = t$: return i

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

Return $\emptyset$

What is the running time?

$$\mathbb{E}\Big(\underbrace{O(m)}_{\text{Initialize}} + \underbrace{O(nk)}_{\text{Mandatory part of loop}} + O(m) \cdot \text{\# of false matches}\Big)$$

$O\left(\frac{n}{m} \cdot \log_2 m\right)$

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

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

set $k = 2\log_2 m$

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

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