CS7800: Advanced Algorithms

Class 23: Randomized Algorithms IV

- Pattern matching

Jonathan Ullman

December 1, 2025

Pattern Matching

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

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

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

First Attempt

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

For $i = n-1, \dots, m$ Counting down is useful later

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

Return i

Return $\emptyset$

What is the running time?

Strings to Numbers

- Can assume $\Sigma = \{0,1\}$ for simplicity

  Everything gets written in binary at some level anyway

- A string $s_n \dots s_1 \in \{0,1\}^n$ is also an n-digit number

$s_5 \quad \dots \quad s_1 \quad s_0$

1	0	0	1	0	1

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

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

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

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

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

  $w \leftarrow (w - s_i \times 2^m) \times 2 + s_{i-m+1}$

Return $\emptyset$

Equal as numbers

What is the running time?

Aside: Randomized Fingerprints

- Can we use hashing to make comparison faster?

$$h : \{0,1\}^m \rightarrow \{0,1,\ldots,B-1\}$$

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

$$\mathbb{P}_h(h(x) = h(y)) = ???$$

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

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

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

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

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

Number of distinct prime factors

Number of k-bit primes

Randomized String Matching

What is the running time?

<u>Input:</u> $s \in \Sigma^n$ $t = \Sigma^m$

$k \ll m$
set later

Let p be a random k-bit prime

$\sigma = 2^m \bmod p$

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

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

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

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

  Check for false match

- $w \leftarrow (w - s_i \times \sigma) \times 2 + s_{i-m+1} \bmod p$

Return $\emptyset$

Randomized String Matching

What is the running time?

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

$k \ll m$ set later

Let p be a random k-bit prime

$\sigma = 2^m \bmod p$

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

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

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

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

  Check for false match

- $w \leftarrow (w - s_i \times \sigma) \times 2 + s_{i-m+1} \bmod p$

Return $\emptyset$