CSE 632: Analysis of Algorithms II: Randomized Algorithms **Spring 2024**

Lecture 13: Fingerprinting

Lecturer: Zongchen Chen

1 Fingerprinting Algorithm

Alice has an n-bit number $x = x_1x_2\dots x_n$. Bob also has an n-bit number $y = y_1y_2\dots y_n$. Their goal is to check if $x = y$ or not efficiently, in the sense of low communication complexity.

Algorithm 1 Fingerprinting algorithm

1: Alice picks a prime number p uniformly at random from $\{2, 3, \dots, T\}$, computes $F_p(x) = (x \bmod p)$, and sends $(p, F_p(x))$ to Bob
2: Bob computes $F_p(y) = (y \bmod p)$, and checks if $F_p(x) = F_p(y)$ or not.

Observe that if $x = y$, then $F_p(x) = F_p(y)$ for any choice of p, and Algorithm 1 outputs Yes always. If $x \neq y$, we need to show that $\Pr(F_p(x) \neq F_p(y)) \geq 1/2$ for a suitable choice of T.

We need some knowledge of number theory to establish this. Let $\log(\cdot)$ denote the natural logarithm (commonly notated as $\ln(\cdot)$ or $\log_e(\cdot)$).

Theorem 1 (Prime Number Theorem)**.**

(1) Let $\pi(x)$ be the number of prime numbers less than or equal to x. Then it holds

$$\pi(x) \sim \frac{x}{\log(x)}.$$

Moreover, for $x \geq 17$, it holds

$$\frac{x}{\log x} \leq \pi(x) \leq \frac{1.26x}{\log x}.$$

(2) Let p_n be the nth prime number. Then it holds

$$p_n \sim n \log n.$$

(3) Let p denote prime numbers in the equation below, and it holds

$$\sum_{p \leq x} \log p = \log \left(\prod_{p \leq x} p \right) \sim x.$$

Lemma 2. *Suppose $T = 8n$. If $x \neq y$, then*

$$\Pr(F_p(x) = F_p(y)) \leq \frac{1}{2}.$$

Proof. Observe that, $F_p(x) = F_p(y)$ if and only if $p \mid |x - y|$. We need to count the number of primes that divides $|x - y|$; denote this count by k. Since $|x - y|$ has at most n bits, k is at most n (more specifically, $2^n \geq |x - y| \geq p_1 \cdots p_k \geq 2^k$ and thus $n \geq k$). In fact, by the Prime Number Theorem, we have $k \leq \pi(n)$

(more specifically, $2^n \geq |x-y| \geq p_1 \cdots p_k \gtrsim e^{p_k} \geq 2^{p_k}$ and thus $n \geq p_k$ which is equivalent to $\pi(n) \geq k$). Therefore, we deduce again from the Prime Number Theorem that

$$\Pr\left(F_p(x) = F_p(y)\right) \leq \frac{\pi(n)}{\pi(T)} \leq \frac{2n/\log n}{T/\log T} = \frac{2n}{T} \cdot \frac{\log T}{\log n} \leq \frac{1}{4} \cdot 2 = \frac{1}{2},$$

as wanted. □

Since it suffices to choose $T = O(n)$ by Lemma 2, both p and $F_p(x)$ have $O(\log n)$ bits. Therefore, in Algorithm 1 Alice only needs to send $O(\log n)$ bits to Bob.

Remark 3. We can combine the polynomial identity testing algorithm with our fingerprinting algorithm. Intermediate computations when evaluating polynomials may result in huge numbers, and we can use modulo a small prime number as in fingerprinting to reduce computational cost.

2 Pattern Matching

Given an n-bit string $x = x_1 \dots x_n$ and a shorter m-bit string $y = y_1 \dots y_m$ where $m \leq n$, check if y occurs as a substring of x or not. In other words, for each j let $w(j) = x_j x_{j+1} \dots x_{j+m-1}$, and we want to check if there exists j such that $w(j) = y$.

Algorithm 2 Pattern matching

```
Pick a random prime p ∈ {2, 3, ..., T}
Compute F_p(y) = (y mod p)
for j = 1 to n − m + 1 do
    Compute F_p(w(j)) = (w(j) mod p)
    if F_p(w(j)) = F_p(y) then
        return Yes
    end if
end for
    return No
```

Correctness. If there exists j such that $w(j) = y$, then $F_p(w(j)) = F_p(y)$ for this j, and Algorithm 2 will output Yes. Now suppose $w(j) \neq y$ for all j. Then for each j, we have

$$\Pr\left(F_p(w(j)) = F_p(y)\right) = \Pr\left(p \mid |w(j) - y|\right) \leq \frac{\pi(m)}{\pi(T)},$$

as before. By the union bound, we have

$$\Pr\left(\text{output Yes}\right) = \Pr\left(\exists j : F_p(w(j)) = F_p(y)\right) \leq \frac{n\pi(m)}{\pi(T)}.$$

In this case we can actually do better than union bound:

$$\begin{aligned}
\Pr\left(\text{output Yes}\right) &= \Pr\left(\exists j : F_p(w(j)) = F_p(y)\right) \\
&= \Pr\left(\exists j : p \mid |w(j) - y|\right) \\
&= \Pr\left(p \,\middle|\, \underbrace{\prod_{j=1}^{n-m+1} |w(j) - y|}_{\leq mn \text{ bits}}\right) \\
&\leq \frac{\pi(mn)}{\pi(T)} \leq \frac{1}{2},
\end{aligned}$$

where the last inequality holds when we set $T = 8mn$.

Running time. Since $T = O(mn)$, the prime p has $O(\log(mn)) = O(\log n)$ bits. Computing $F_p(y)$ takes $O(m)$ time since y has m bits. Computing $F_p(w(j))$ for all j can be done recursively:

$$\begin{aligned} & w(j+1) = 2w(j) + x_{j+m} - 2^m x_j \\ \Longrightarrow \qquad & F_p(w(j+1)) = ((2F_p(w(j)) + x_{j+m} - F_p(2^m)x_j) \mod p) \end{aligned}$$

Thus, each iteration takes $O(1)$ time (note that we can compute $F_p(2^m) = (2^m \mod p)$ once for all). The total running time is $O(n+m)$.