Tutorial XII

07 May 2020 14:05

1.) Fingerprinting and string comparison

2.) Pattern matching

Schwartz-Zippel thm.

$$\Pr\left(Q(r_1,\dots,r_n) = 0 \mid Q \not\equiv 0\right) \leq \frac{\deg Q}{|S|}$$

$r_i \in_R S$

$\boxed{\text{Problem}}$ Verify whether two strings X and Y $X, Y \in \{0,1\}^n$ are equal.

Deterministically $O(n)$ $X = (x_1,\dots,x_n)$ $x_i, y_i \in \{0,1\}$
$Y = (y_1,\dots,y_n)$

If comparison is expensive then S-Z thm gives us solution

→ Interpret X and Y as multivariate polynomials:

variable

$$X(z_1,\dots,z_n) = \sum_{i=1}^{n} x_i z_i \mod p$$

$$Y(z_1,\dots,z_n) = \sum_{i=1}^{n} y_i z_i \mod p$$

$$X(z_1,\dots,z_n) - Y(z_1,\dots,z_n) \overset{?}{\equiv} 0$$

Choose $\vec{v} \in \{0,1\}^n$ by S-Z

$$\Pr\left(X(v_1,\dots,v_n) - Y(v_1,\dots,v_n) = 0 \;\middle|\; [X-Y](\vec{z}) \not\equiv 0\right) \leq \frac{\deg[X-Y]}{2} = \frac{1}{2}$$

Context: Database comparison

- two distant databases X and Y. Are they the same?
- Expensive operation: transmitting a bit (sending messages between the databases)

  Is the method above efficient?

NO! Random v needs to be distributed. It is as long as the database (n bits need to be transmitted)

Solution 1.

Interpret both X and Y as numbers:

$$num(X) = \sum_{i=1}^{n} x_i 2^{i-1}$$

$$num(Y) = \sum_{i=1}^{n} y_i 2^{i-1}$$

Compare

$\underline{X \bmod p}$ and $\underline{Y \bmod p}$ fingerprints

for suitably chosen prime p. If p is small, fingerprints are also small (in bit). However there is a trade-off between the size of p and the probability of error.

Error can happen if $X \neq Y$ but $X \equiv Y \mod p$

$X - Y \equiv 0 \mod p$ (read as $X-Y$ is divisible by p)

$\pi(k)$ – number of all primes smaller than k

$$\pi(k) = O\left(\frac{k}{\ln k}\right)$$

for $k \geq 29$ $\quad \pi(k) \leq 1.2\ldots \frac{k}{\ln k}$

$$\Pr(X - Y \equiv 0 \mod p \mid X \neq Y) = \frac{\#\text{ bad primes}}{\#\text{ primes we choose from}} = \frac{n}{\pi(k)} \leq \frac{\ln k \cdot n}{k}$$

\# bad primes: How many prime divisors can $X-Y$ have at most?

What is the largest value of $X-Y$? $\quad \underline{X - Y < 2^n}$

What is the smallest number with $\underline{n}$ prime divisors?

$$= \prod_{i=1}^{n} p_i > 2^n = \prod_{i=1}^{n} 2 \Rightarrow \#\text{ bad primes} < n$$

p_i – i-th smallest prime

for $k = \boxed{t \cdot n \log(t_n)}$

$$\Pr < \frac{\ln(t \cdot n \cdot \log(t_n)) \cdot \cancel{n}}{t \cdot \cancel{n} \log(t_n)} \in O\left(\frac{1}{t}\right)$$

How many bits does X need to send to Y?

for $t = n$ a prime of $O(\log n)$ bits and the fingerprint $O(\log n)$.

$$\boxed{X = \sum_{i=0}^{n} x_i \cdot z^{i-1} \mod p}$$

Solution 1 above:

choose $z = 2$ and randomize over p

Solution 2:

choose p and randomize over z.

Method 2 analysis: using $S - Z$

$X(z)$ and $Y(z)$ are polynomials mod p

$r_p \in_R S$

$$\Pr\left[(X-Y)(r) = 0 \mid (X-Y)(z) \neq 0\right] \leq \frac{\deg(X-Y)}{|S|} = \frac{n-1}{|S|}$$

to match the method 1 we would like this probability to be roughly $1/n$.

$\Rightarrow$ $|S|$ is roughly n^2. $\Rightarrow$ p needs to be larger than n^2

What needs to be sent?

$r < p \sim O(\log(n))$ bits

and $X(r) \bmod p$ $O(\log(n))$ bits

3rd method:

choose a random polynomial P mod p and evaluate

$P(\text{num}(X))$ and $P(\text{num}(Y))$ and compare.

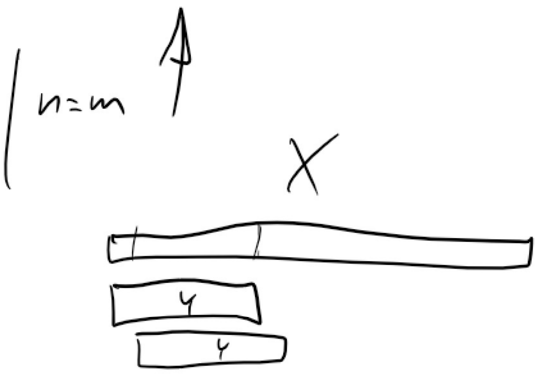
↳ this leads to universal hashing.

Pattern matching

X - a string of length n

Y - a string of length $m \leq n$

Is Y a substring of X?



Naive algorithm $\approx O(m \cdot n)$ comparisons

Better solution (Knuth-Morris) $\approx O(m+n)$ comparisons

Rabin-Karp monte carlo algorithm (1-sided error) $\boxed{O(m+n) \text{ time}}$

Main idea: use fingerprints (as defined above)

Imagine calculating fingerprints is free. How many comparisons do you need?

Each fingerprint $O(\log m)$ bits long

this is not is ment in the slides

$\approx O(n \cdot \log m)$

Strings are arrays of objects:

Both X and Y are in memory of a computer in an indexed array, that is each X_i and Y_i needs to be addressed separately in the memory.

In Rabin Karp algorithm the expensive operation is calculating the hash $num(Y) = \sum_{i=1}^{m} y_i 2^{m-i} \bmod p$. Because you need to access each bit of array $Y \approx O(m)$

need to access each bit of array $Y \approx O(m)$

Comparison of fingerprints is a comparison of two integers in $O(1)$.

Naive calculation of hashes leads to $O(m \cdot n)$ memory access instances

Trick of RK algorithm is efficient fingerprint calculation

let $X_j = (x_j, \dots, x_{j+m-1})$

$$F(x_j) = x_j \cdot 2^{m-1} + x_{j+1} 2^{m-2} + x_{j+2} 2^{m-3} + \dots + x_{j+m-1} \mod p$$

$$F(x_{j+1}) = x_{j+1} \cdot 2^{m-1} + x_{j+2} 2^{m-2} + x_{j+3} 2^{m-3} + \dots x_{j+m-1} \cdot 2 + x_{j+m}$$

$$F(x_j+1) = 2 \cdot \left[F(x_j) - 2^{m-1} \cdot x_j\right] + x_{j+m}$$

Time analyses:

The hash of Y : $F(Y)$ m memory fetches

The hash of X_1 : $F(X_1)$ m memory fetches

$n-m$ following hashes each in $O(1)$ $\approx$ n memory fetches

$\Bigg\}\ O(m+n)$