CS 124 Lecture 13 Monday, March 23rd

Today: Fingerprinting for Pattern Matching

- Find occurances of pattern P in long document D
- O(|P||D|) obvious algorithm
- Lots of methods

Approach 1: Hash Pattern to x-bit string

Compare hash value to hash value

Hash Function: mod prime

P = 17935 p = 251

P mod p = 114

638617935 7342

63861 mod p = 107 17935 mod p = 114

38617 mod p = 214 57342 mod p = 114

86179 mod p = 86...

$N \bmod p \rightarrow N' \bmod p$ (first digit a, last digit b)

$N' = (N - 10^{|P|-1} \times a) \times 10 + b$

$N' \bmod p = ((N - 10^{|P|-1} \times a) \times 10 + b) \bmod p$

Assuming mod p operations are O(1) → not constant but only calculate once.

$\pi(x)$ = # of primes $\leq x$

$\lim_{x \to \infty} \pi(x)$ grows $x / \ln(x)$

$\frac{x}{\ln(x)} \leq \pi(x) \leq 1.26\, x / \ln x$

$\longrightarrow$ for 10^{100}, $10^{100} / 100 \times \ln(10)$ primes

Random Prime

FP: two substrings P and A have $(P-A) \bmod p = 0$

Pr[any false positives in document] =

$$\frac{\#\text{ primes } p \text{ s.t. } p \mid P-A}{\pi(z)}$$

pick a prime $1 \leq p \leq z$

$|P-A| \leq 10^{|P|}$

How many primes are a factor of n, where $n \leq 10^{|P|}$

Claim. n has at most $\log_2(n)$ distinct factors

$n \geq \pi$ prime factors

$\geq 2^{\#\text{ prime factors}}$

$\#\text{ prime factors} \leq \log_2 n$

$$Pr[FP] \leq \frac{\log_2 10^{|P|}}{\pi(z)}$$

Expected # of FP = $|D| \times Pr[FP] = |D| \times \log_2 10^{|P|} / \pi(z)$

* can decrease by using multiple hash functions (primes)

$$Pr[FP] = \left(\frac{\log_2 10^{|P|}}{\pi(z)} \right)^k$$ where k primes used

Primality Test

TO pick random prime:

↗ pick random #
test if prime
↳ repeat if necessary

Test if prime:

$n \to \log_2 n$ digits ← want polynomial in $\log_2 n$

is 2 a factor
3 a factor
⋮
$\sqrt{n}$ a factor

Fermat's Little Theorem:

If p is a prime, $1 \le a < p$, $a^{p-1} = 1 \bmod p$

Pf. $\{1, \dots p-1\}$ and $\{a \cdot 1, a \cdot 2, \dots, a(p-1)\}$ mod p mod p

In the second set, all #s between 1, p-1

All different because $i \cdot j$ and j were same:

$a \cdot i = a \cdot j \bmod p$

↳ $i = j \bmod p$ → contradiction

Mod p there are multiplicative inverses

$1 \times 2 \times 3 \dots \cdot p-1 =$

$(a \times 1)(a \times 2) \dots (a(p-1)) \bmod p$

$= a^{p-1} \times 1 \times 2 \times \dots \times (p-1) \bmod p$

$1 = a^{p-1} \bmod p$

FLT Primality Test

Given n, is n prime?

Test: compute 2^{n-1} — square using repeated squaring

If $2^{n-1} \equiv 1 \bmod n$ say prime

$2^{n-1} \not\equiv 1 \bmod n$ say composite

Correctness:

only one direction, 1 means not prime

FP possible

2-pseudoprime numbers (satisfies Fermat's Little Thm but not prime)

Carmichael numbers: composites s.t. $a^{n-1} \equiv 1 \bmod n$ for any a that does not share a factor with n.

↑ infinite #!

Improvement: Rabin-Miller Primality Test

Composites have nontrivial square roots of 1

trivial: $(-6)^2 \equiv 1 \bmod 7$ → trivial

nontrivial: $(4)^2 \equiv 1 \bmod 15 \equiv (11)^2$

$x^2 \equiv 1 \bmod n$

$x^2 - 1 \equiv 0 \bmod n$

$(x+1)(x-1) \equiv 0 \bmod n$

compute a^{n-1} — $a^u, a^{2u}, a^{4u}, a^{8u}, \ldots a^{2^t u} = a^{n-1}$

$n-1 = 2^t u$ where u is odd

if $a^{n-1} \neq 1$, composite!

till $a^{n-1} \equiv 1$, go back through the chain as long as you see a 1. if see -1 at end, prime. If see anything else, composite.

For any composite odd n, for at least $3/4$ of the values $1 \leq a < n$, the test will return composite

$\leq (1/4)^{50 \text{ times}} = 2^{-100}$