# Karp-Rabin fingerprint

- How to check equality in files?
- How to check for computer viruses?

It can be done deterministically, but it is slow!

Idea: $\Sigma = \{0,1\}$ binary sequences

sequence $S = 0111011101110111011$, $w = |S|$

$S$ can be seen as a huge number $\in [0 .. 2^n - 1]$

another sequence $S' = \ldots - - -$

Check whether $S = S'$? It can be done in $O(n)$ time

Same file but different names, same Qsynth but different file,

$F(s) = s \,\%\, p$ for a randomly chosen prime (see pseudocode)

$s$: both a sequence and a number

Issue: collision $k_1, k_2 \in \Sigma^*$

$F(k_1) \neq F(k_2) \Rightarrow k_1 \neq k_2$

$F(k_1) = F(k_2) \Rightarrow k_1 = k_2$

$F(k_1) = F(k_2) \Rightarrow k_1 \neq k_2$ error $\Rightarrow$ probability

1-sided error

Let $n$ be the number of bits representing $k_1, k_2$

$\Rightarrow k_1, k_2 \in [0..2^n - 1]$

error = $k_1 \neq k_2 \wedge F(k_1) = F(k_2)$ 1-side error

Idea:

- choose a parameter $z$ sufficiently large ← see later
- choose uniformly and randomly a prime in $[2 \dots z]$ ← BAD prime?

Before starting the computation

$$F(k_1) = F(k_2) \Rightarrow k_1 \% p = k_2 \% p \qquad (k_1 \neq k_2)$$

we call such a prime $p$ BAD

$$\Pr(\text{error}) = \Pr_{\substack{\text{primes} \\ p \leq z}}\left(k_1 \neq k_2 \wedge F(k_1) = F(k_2)\right)$$

$$= \frac{\#\,\text{BAD primes} \leq z}{\#\,\text{primes } p \leq z}$$

$$= \frac{?}{z / \ln z} \qquad (*)$$

Obs both $k_1$ and $k_2$ require $n$ bits of representation

Fact Given an integer $k$ on $n$ bits, there are at most $\leq n$ primes in its prime factorization

$$k = p_1^{e_1} \cdot p_2^{e_2} \cdots p_r^{e_r} \quad , \; r \leq n$$

Proof $p_i \geq 2 \Rightarrow e_1 + e_2 + \cdots + e_r \leq n$ as otherwise

$k \geq 2^{e_1 + e_2 + \cdots + e_r}$ would be greater than $2^n$ which is impossible as $k$ requires $n$ bits

CLAIM: there are $\leq n$ BAD primes

- Common primes for $k_1$ and $k_2$ are $\leq n$ in number

$$\Pr(\text{error}) = (*) \leq \frac{n}{r/\ln r} = \frac{1}{n^c} \qquad r \sim n^{c+1} \ln n$$

Probabilistic algorithms : w.h.p. with high probability $= \frac{1}{\text{poly}(n)}$

| Las Vegas | Monte Carlo | |
|---|---|---|
| no errors | 1-side / 2-side errors | answer |
| running time is expected (random var.) | running time is worst-case | time |

$$\frac{r}{\ln r} = n^{c+1}$$

Pattern matching:

pattern $P[0..m-1]$

text $T[0..n-1]$ $\quad m \le w$

$P$ occurs at position $i$ in $T$ if $P = T[i..i+m-1]$

GOAL: find whether $P$ occurs in $T$ (virus scan has several simultaneous patterns)

(See pseudo code)

Montecarlo: if $F(P) = F(T[i..i+m-1])$ then YES

pr. error: UNION BOUND $Pr(A \cup B) \le Pr(A) + Pr(B)$

1-side error: no occurrence is missed

false positive

$\tau \sim n^{c+2} \ln w$

$$Pr(\text{error}) \le \frac{m}{\tau/\ln \tau} \times (n-m+1) \le \frac{nm}{\tau/\ln \tau} \le \frac{n^2}{\tau/\ln \tau} \le \frac{1}{n^c}$$

Las Vegas: if $F(P) = F(T[i..i+m-1])$ then

if $\underbrace{P = T[i..i+m-1]}_{O(m) \text{ time}}$ then $Y \in S$

Small change: when $F(P) = F(T[i..i+m-1])$ and $P \neq T[i..i+m-1]$

we "switch" (conceptually) to the trivial $O(nm)$ time alg.

$m \leq n$

Expected cost:

$$O(nm) \cdot \frac{1}{n^c} + O(n+m) \cdot \underbrace{\left(1 - \frac{1}{n^c}\right)}_{\leq 1} = O(n)$$

$c > 2$

$$X_i = \begin{cases} 1 & \text{if } F(P) = F(T[i..i+m-1]) \wedge P \neq T[i..i+m-1] \\ 0 & \text{o.w.} \end{cases}$$

$$X = \sum_{i=0}^{n-m} X_i$$

Real cost $O(n + Xm)$ time

Expect cost $O(n + E[X]m)$ time $(\Delta)$

$E[X] = (n-m+1) \cdot E[X_i] \leq n/m^c$

$\smile \Pr[X_i = 1] \leq \frac{1}{m^c}$

(A) $O\left(n + \underbrace{\frac{n}{m^c} \cdot m}\right) = O(n)$ time

$$\frac{n}{m^{c-1}} \leq n$$

$$c > 1$$

---

HANDS-ON

- set of sequences $s_1, s_2, \ldots, s_k$, $m > 0$
- change a sequence by adding/removing characters
- issue an alert when any two sequence $s_i, s_j$ $(i \neq j)$ share a common substring of length $m$