Lecture 28

Friday, 22 October 2021 10:00 AM

only $O(k)$ addⁿ space.

Application of mod. arith in String Matching

Given: Text $T = (t_{n-1}\ t_{n-2} \ldots t_1\ t_0)$ $|T| = n$

Pattern $X = (x_{k-1}\ x_{k-2} \ldots x_0)$ $|X| = k$ $n \gg k$

binary string

KMP (lec 16) – $O(n+k)$ time algo, $O(n+k)$ space

k bits

$T =$ 1 1 1 0 1 0 1 1 0 0 1 0 1

Yes – $X_1 =$ 1 1 0 0 (12)

No – $X_2 =$ 1 1 0 0 0

Convert / Represent X as decimal:

$X = (x_{k-1} \ldots x_1\ x_0)$ k-bit binary no.

$N_X = 2^{k-1} x_{k-1} + \ldots + 2^1 x_1 + 2^0 x_0$

k-bits

$N_T(j) = 2^{k-1} t_{j+k-1} + \ldots + 2^1 t_{j+1} + 2^0 t_j$

↳ decimal eq of $(t_{j+k-1}, \ldots, t_j)$

Direct comparison — $O(nk)$ time complexity.

Take help of Hashing.

U → small no with less no of bits.
$[1, M]$

Hash Fn $H: \mathbb{Z} \to \mathbb{Z} \bmod p$

p — random prime in range $[2, n^4]$

(2) $\bmod p$ ← no of bits that will be dealing with is $(4 \log n)$

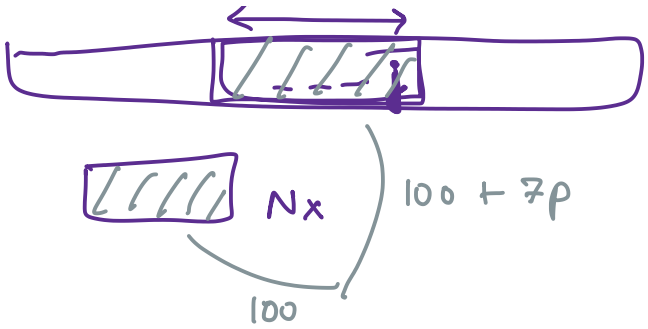
Sketch of Algo

(1) ans = False

(2) for all values of $j \le n-k$:

if $H(N_x) = H(N_T(j))$ ans "True" } $O(1)$ time
k bits

(3) Return ans.



SHOW

* ans is correct w.p. $\geq 1 - \frac{1}{n}$
* Algo can be implemented in $O(n+k)$ time.

Prime No. Theorem: No of primes in range $[1, L]$ is $\Theta\left(\frac{L}{\log L}\right)$

↑ density of primes no.

Error

Case 1 $N_T(j) = N_x$

$\Rightarrow H(N_T(j)) = H(N_x)$

ans = True

CASE 2 $N_T(j) \neq N_x$

but $N_T(j) \bmod p = N_x \bmod p$

$\Rightarrow \left(\frac{N_T(j) - N_x}{p}\right)$ is integer

How many choices for prime p in range $[2, n^4]$ — $\Theta\left(\frac{n^4}{\log n}\right) = \frac{cn^4}{\log n}$

$$\text{Prob}(\text{error at location } j) = \frac{\text{No of prime factors of } |N_{T_j} - N_x|}{cn^4/\log n}$$

For any $z \le 2^k$

if $z = p_1 \cdots p_\alpha \ge 2^\alpha$

$\Rightarrow \alpha \le k$

$\Rightarrow$ No of prime factors $\le k$

$$\le \frac{k}{cn^4/\log n}$$

$$\le \frac{k(c' \log n)}{n^4}$$

$k \le n$

$c' \log n \le n$, for large n

$$\le \frac{1}{n^2}$$

$$P(\text{error}) \le \sum_{j=1}^{n} \text{Prob}(\text{error at } j) \le \frac{n}{n^2} \le \frac{1}{n}$$

Union Bound

Success prob $\ge 1 - \frac{1}{n}$

Time complexity

$H(N_x)$ — $O(k)$ time ✓

$H(N_{x(i)})$ H: ?

$$N_T(j) = 2^{k-1} t_{j+k-1} + \cdots + 2^1 t_{j+1} + 2^0 t_j$$

$$N_T(j+1) = 2^{k-1} t_{j+1+k-1} + \cdots + 2^1 t_{j+2} + 2^0 t_{j+1}$$

Ques: Relation b/w $N_T(j+1)$ & $N_T(j)$?

$$N_T(j+1) = \frac{N_T(j) - t_j}{2} + 2^{k-1} t_{j+k}$$

GOAL

$$H(N_T(j+1)) = N_T(j+1) \bmod p =$$

$$= \left(\frac{1}{2}\left(N_T(j) - t_j\right)\right) \bmod p + \left(\left(2^{k-1} \bmod p\right) * t_{j+k}\right) \bmod p$$

$$= \left(\left(2^{p-2} \bmod p\right)\left(N_T(j) \bmod p - t_j \bmod p\right) + \left(\left(2^{k-1} \bmod p\right) * t_{j+k}\right)\right) \bmod p$$

A = $2^{p-2} \bmod p$; $N_T(j) \bmod p = H(N_T(j))$; B = $2^{k-1} \bmod p$

Lec 26 p-1

Claim 1: $2^{} = 1 \pmod p$

$\frac{1}{2} \equiv 2^{p-2} \pmod p$

✓

$[0, p-1]$

CLAIM: $H(N_T(j+1))$ can be computed from $H(N_T(j))$ in $O(1)$ time, if we know values of A & B.

KEY POINTS

1

$[1, n^4]$

random prime no.

generate sufficiently many numbers

- One of them will be prime.

(ii) Computing A, B using Divide & Conquer Approach:

$2^y \bmod b$ — $O(\log y)$ time

$$2^y = 2^{\lfloor y/2 \rfloor} \cdot 2^{\lfloor y/2 \rfloor} \cdot 2$$

OR

$$2^y = 2^{\lfloor y/2 \rfloor} \cdot 2^{\lfloor y/2 \rfloor}$$

* Time to compute A, B — $O(\log n + \log k)$.

(iii) WORD RAM — $4 \log n$

All add / sub / mult on nos with $4 \log n$ bits will take $O(1)$ time.

32 bit

64 bit