COL 351: Analysis and Design of Algorithms

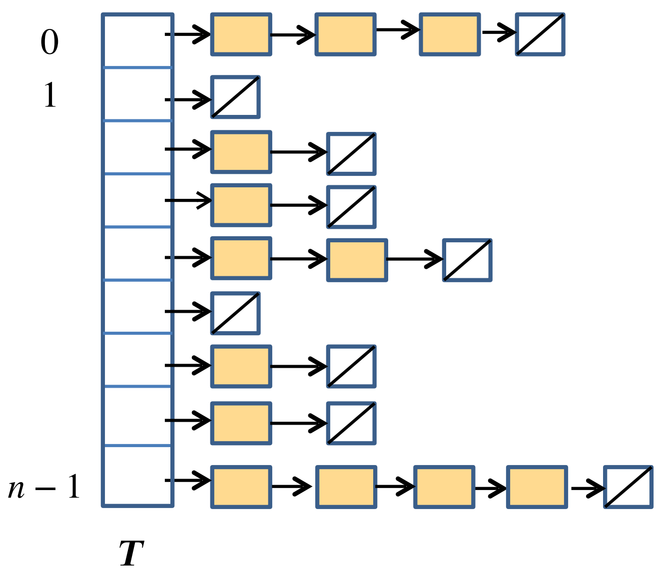
Lecture 21

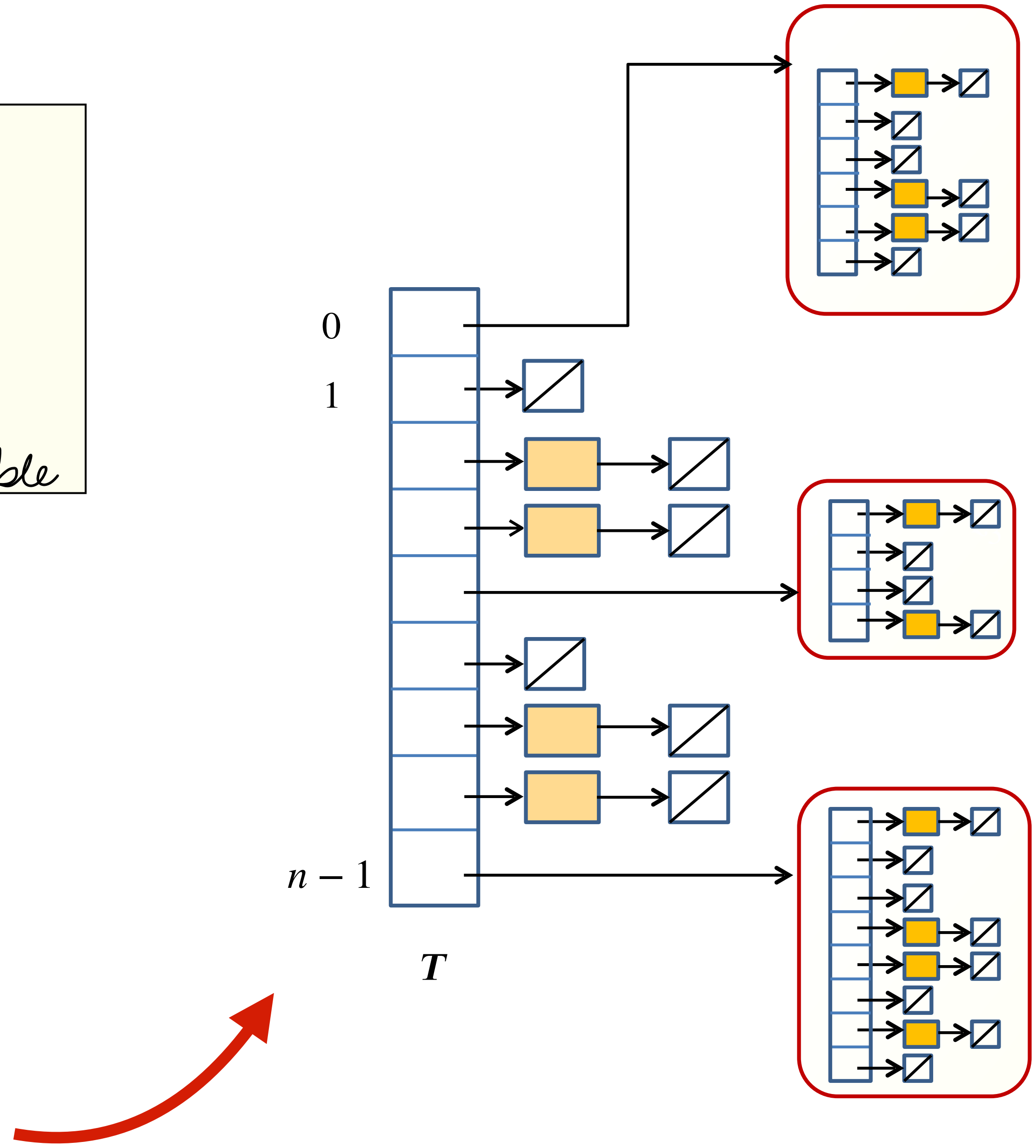
Perfect Hashing

Goals:

1. Expected size = $O(n)$.

2. Expected number of total collisions is $O(1)$, for each secondary table





Lemma 1

- Universe $U = [1,\ M]$.
- $p =$ prime in range $[M+1,\ 2M]$, $\quad r =$ integer in range $[1,\ p-1]$

Hash Function:

$$H_r(z) \;=\; \big((r \cdot z) \;\bmod p\big) \;\bmod n$$

Lemma 1: Let $x, y \in U$, and 'r' be randomly chosen. Then, $\text{Prob}\Big(H_r(x) = H_r(y)\Big) \;\leq\; \dfrac{2}{n}$

Lemma 2

Hash Function:

$$H_r(z) \;=\; \big((r \cdot z) \;\bmod p\big) \;\bmod n$$

Lemma 2: For a set S of size n, the expected number of total collisions is:

$$\sum_{\substack{x, y \in S \\ x \neq y}} \text{Prob}\big(H_r(y) = H_r(x)\big) \;\leq\; {}^nC_2 \cdot \frac{2}{n} \;\leq\; n$$

Two-Level Hash Table

Outer Hash Function:

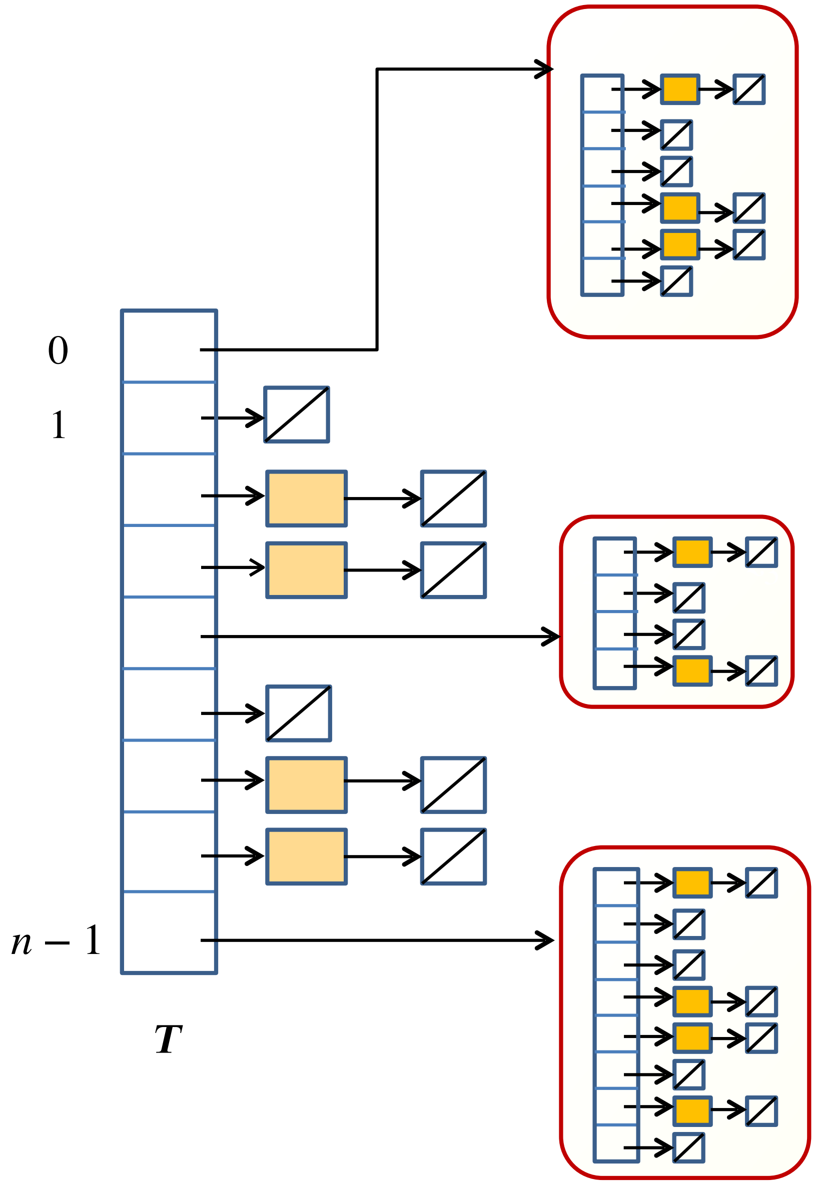
$$H_r(z) \;=\; \big((r \cdot z) \;\bmod p\big) \;\bmod n$$

Inner Hash Function:

$$z \mapsto \big((r_0 \cdot z) \;\bmod p\big) \;\bmod n_i^2$$

where, n_i = size of $T[i]$

- $r,\ r_0$ = random integers from $[1,\ p-1]$



New Table

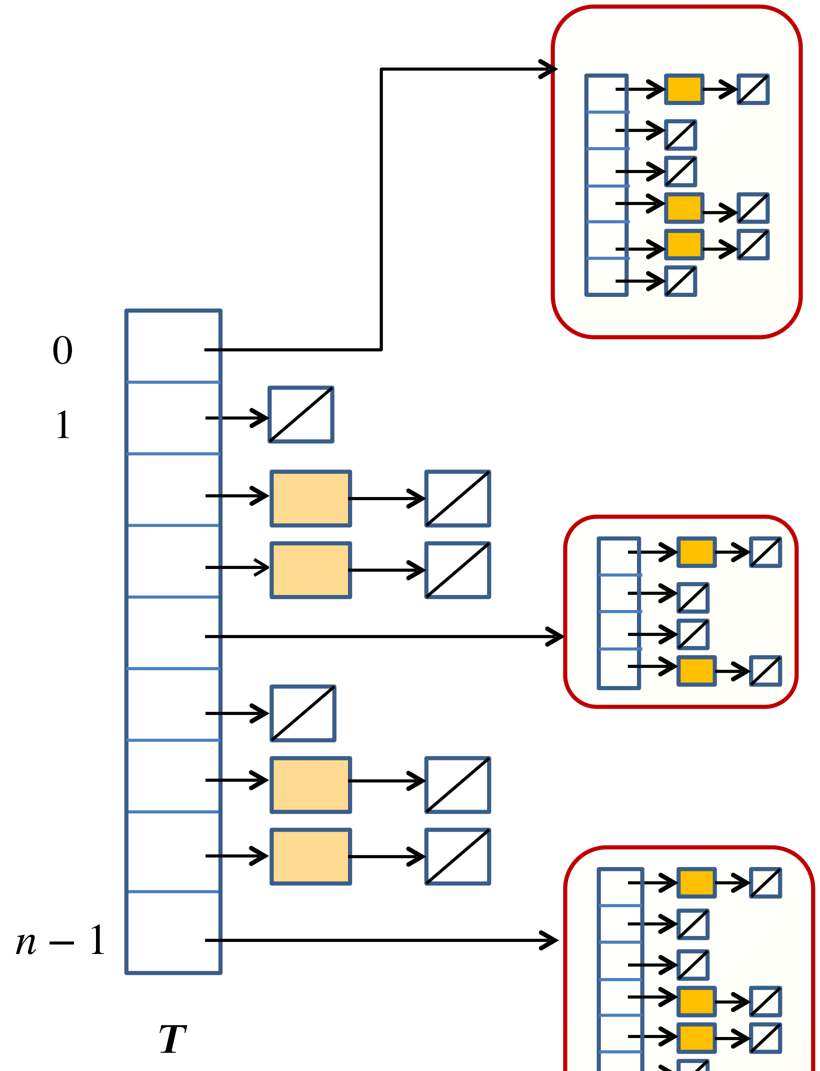
Expected Size of Data-structure

Size is: $n + \sum_{i=0}^{n-1} n_i^2$

$n_i \geq 1$

$$\leq n + \sum_{\substack{0 \leq i \leq n-1 \\ n_i \geq 1}} (n_i^2 - n_i) + \sum_{0 \leq i \leq n-1} n_i$$

$$\leq 2\Big(n + \underbrace{\sum_{\substack{0 \leq i \leq n-1 \\ n_i \geq 1}} (n_i^2 - n_i)}_{\text{Total no. of collisions from outer Hash fn}}\Big)$$



New Table

Exp size $= n +$ Expected no. of total collisions from outer hash $= O(n)$

Number of Collisions in a Secondary table

Hash Function:

$$z \mapsto \big((r_0 \cdot z) \mod p\big) \mod n_i^2$$

***Question*:** For a set of size n_i, what is expected number of **total** collisions?

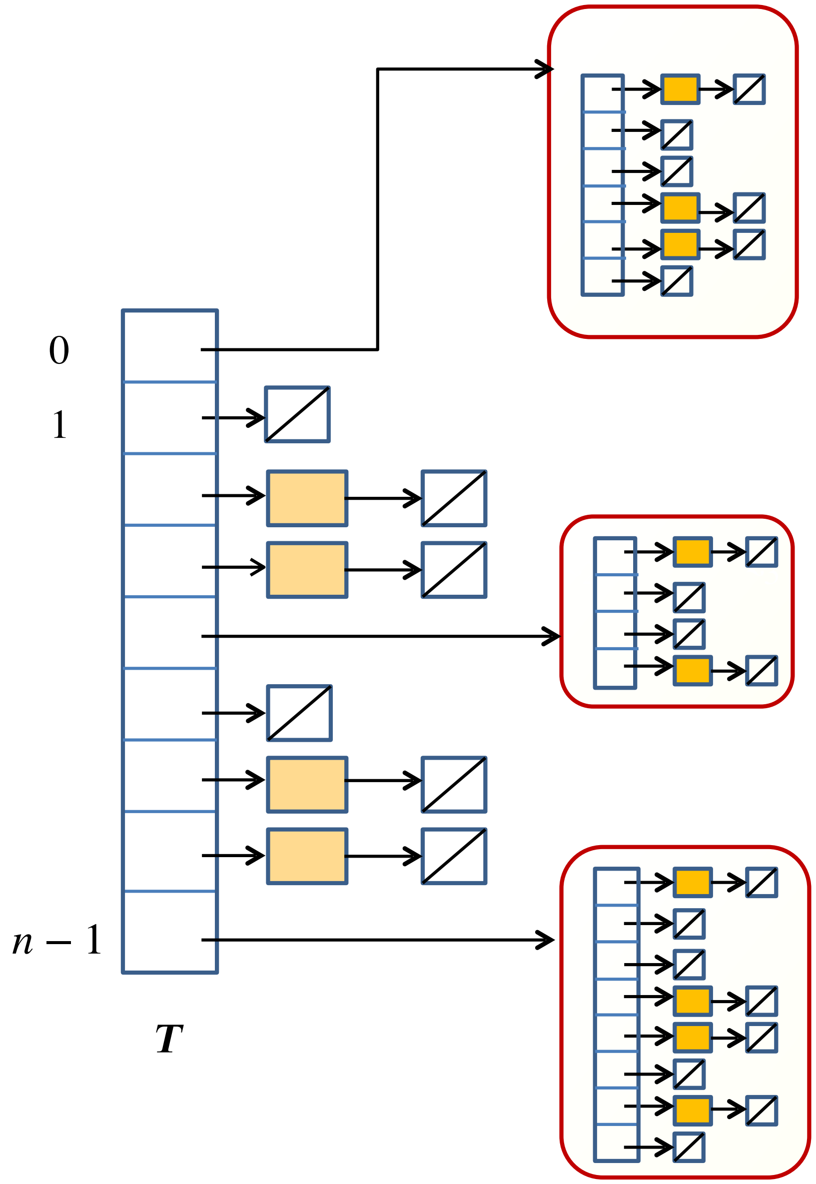
Answer: $$\sum_{\substack{x, y \in \text{set} \\ x \neq y}} \text{Prob}\big(H_r(y) = H_r(x)\big) \leq {}^{n_i}C_2 \cdot \frac{2}{n_i^2} = O(1)$$

Perfect Hashing

Goals:

1. Expected size = $O(n)$.

2. Expected number of total collisions is $O(1)$, for each secondary table



Pattern Matching

Pattern Matching

Given: String T = $(t_{n-1}, \ldots, t_1, t_0)$ and a pattern X = $(x_{k-1}, \ldots, x_1, x_0)$, both binary.

Find: If there exists a sub-string of T that is identical to X.

Know algorithms:

- Brute-force comparison
- KMP algorithm

Can we have a simpler hashing based algorithm?

Numeric/Decimal Representation

X = $(x_{k-1}, \ldots, x_1, x_0)$

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

(decimal form of X)

$X = 0101$

$N_X = 5$

T = $(t_{n-1}, \ldots, t_1, t_0)$

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

(decimal form of $(t_{j+k-1}, \cdots, t_{j+1}, t_j)$)

							6	5	4	3	2	1	0
P	H	A	S	E	D		H	A	S	H	I	N	G

-

H	A	S	H
3	2	1	0

	1	1	0	1	
				j	

$N_T(j) = 13$

Algorithm

Flag= False

For $j = 0$ to $(n - k)$:

If $N_X = N_T(j)$ **then**

Flag = True

Return Flag

Time = $O(n \cdot k)$

Time to compute N_X, $N_T(j)$

check if $N_X = N_T(j)$

Algorithm

- If $N_X = N_T(j)$, then $H(N_X) = H(N_T(j))$
- If $N_X \neq N_T(j)$, then we want with high prob. $H(N_X) \neq H(N_T(j))$

Hash Function:

$H : z \to z \mod p$

p = random prime in range $[2, n^4]$.

Flag= False

For $j = 0$ to $(n - k)$:

If $H(N_X) = H(N_T(j))$ **then**

Flag = True

Return Flag

Show:

- Answer returned is correct with probability $(1 - 1/n)$.
- Implementation in $O(n)$ time.

Hints:

Claim 1: For any integer $z \leqslant 2^k$, the number of distinct prime factors of z is at most k.

Claim 2: For any $j \leqslant n - k$, the number of distinct prime factors of $(N_T(j) - N_X)$ is at most n.

How to compute $\text{Prob}(H(N_x) = H(N_T(j))$ for $N_x \neq N_T(j))$?

Prime Number Theorem: Number of primes in the range [2,L] is $\Theta\left(\frac{L}{\log L}\right)$.

$$H(N_x) = H(N_T(j)) \implies p \text{ divides } N_T(j) - N_x$$

- No of Prime factors of $N_T(j) - N_x$ is $\leq n$.
- No of choices for $p = \Theta\left(\frac{n^4}{\log n^4}\right)$

So,

$$\text{Prob}\Big(H(N_x) = H(N_T(j))\Big) \leq \frac{n}{\Theta(n^4/\log n^4)} \leq \frac{c \cdot \log n}{n^3}$$