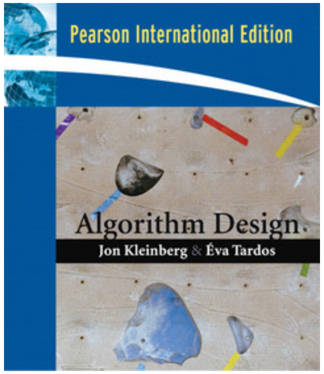
The Dictionary Problem
and
the Hash Functions

Luciano Gualà and Andrea Clementi

Hash tables

A randomized implementation of dictionaries

reference (Chapter 13.6)



+

Design and Analysis of Algorithms (MIT opencourseware) Lecture 8

https://ocw.mit.edu/courses/6-046j-design-and-analysis-of-algorithms-spring-2015/resources/lecture-8-randomization-universal-perfect-hashing/

The dictionary problem:

Given a universe U of possible elements, maintain an *arbitrary* subset $S \subseteq U$ of n elements subject to the following **operations**:

- make-dictionary(): Initialize an empty dictionary.
- insert(u): Add element $u \in U$ to S.
- delete(u): Delete u from S, if u is currently in S.
- look-up(u): Determine whether u is in S.

Challenge: Universe U can be extremely large w.r.t. n so defining an array of size $|U|$ is infeasible. Solutions should be proportional to $|S| = n$

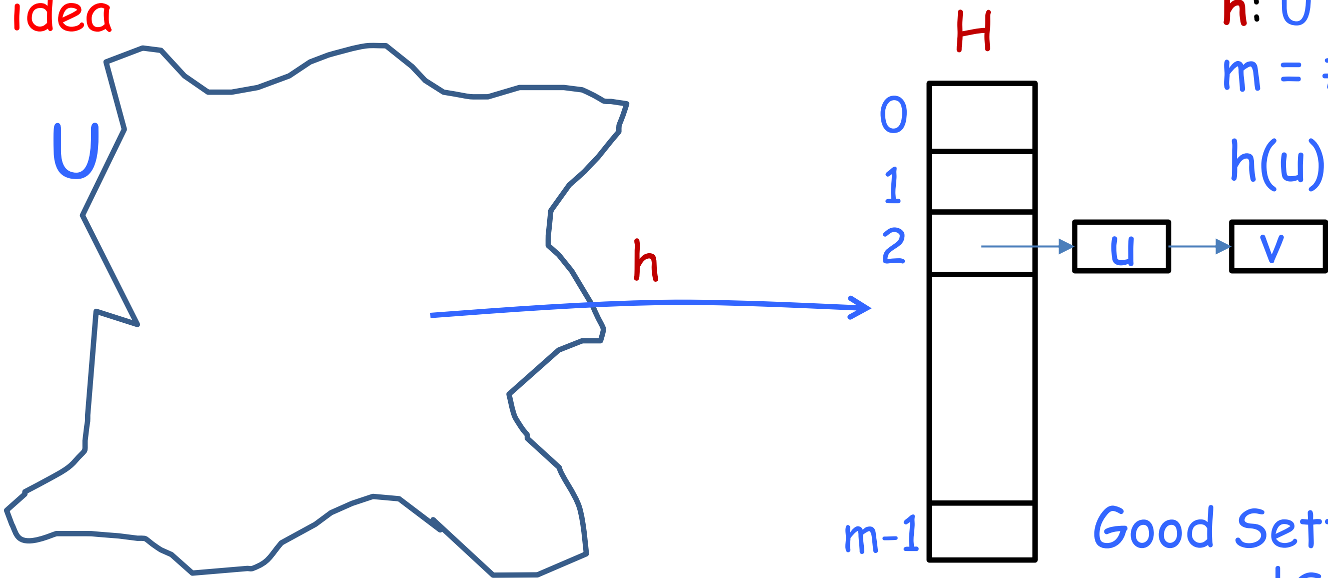
A deterministic solution: balanced (e.g. ***AVL***) trees

- **O(n) space**
- **O(log n) time per operation**

A Randomized Solution: Hash Tables

- O(n) space
- O(1) expected time per operation

idea



h: $U \longrightarrow \{0,1,\ldots,m-1\}$
m = # of slots of H

$h(u)$: slot where u is stored

Good Setting:
$m \approx n := |S|$

collision: when $h(u) = h(v)$ but $u \neq v$.

$H[i]$: linked list of all elements that h maps to slot i
(hashing with chaining)

Insert/Delete/Lookup of u:
- compute $h(u)$
- insert/delete/search u by scanning list $H[h(u)]$

Goal: Design a function h that *well-distribute* elements

DESIGNING GOOD HASH FUNCTIONS: Wrong Approach

Fact I: IF $|U| > m^2$, for ***any*** deterministic hash function **h**, there is a set **S of size n** s.t. *all* elements of S are mapped to the same slot. :(

Proof: **h** is fixed and must map every U-element to H and S can be chosen *adversarially* w.r.t. h. So there is at least one slot i of H that must store >= n elements. Then choose $S = \{u \in U : h(u) = i\}$

⇒ $\Theta(n)$ **congestion** and $\Theta(n)$ **time** per operation !!

Deterministic Hashing

- Let U be the Universe and $|U| = N \gg m \approx n := |S|$
- Represent elements u of U, as integers in [N]
- Take p any prime number s.t. $m \le p \le 2m$
- Det. Hashing: $h(u) = u \bmod p$

- Then, **Fact I** still clearly holds but the elements of U are well distributed!

It works quite well for some "*static*" applications!

RANDOMIZED HASH FUNCTIONS

Trivial non-efficient approach: for each u, choose **h(u)** *independently* and *uniformly at random (i.u.r.)* , i.e.,
for any slot $i \in H$, $Pr[h(u) = i] = 1/m$

\+ Nice distribution, no matter who is S !!!!

\- Efficiency: Terrible!!!.....

look-up(u): ...where did we put u?...

to implement one h we have to store the set of **all** pairs $\{(u,h(u)): u \in S\}$, in particular we have to store **n** independent addresses h(u)'s !!!

we back to the dictionary problem!

maybe I can use a hash table

Target Property: Universal Hashing

Main Idea: use a *family* of Hash Functions

DEF. A family $\mathcal{H}$ of hash functions is ***universal*** if

for each distinct $u, v \in U$ $\qquad \Pr_{h \in \mathcal{H}}(h(u)=h(v)) \le 1/m$

Recall: $|U| = N \gg |S| = n \approx |H| = m$

Theorem. 1

Let $\mathcal{H}$ be a family of universal hash functions. Let $S \subseteq U$ of n elements. Let $u \in S$. Pick *u.a.r.* function h from $\mathcal{H}$, and let X be the random variable counting the number of elements of S mapped to slot $h(u)$.

Then: $E[X] \leq 1+n/m$

Proof. Fix u.

for each $s \in S$, X_s r. v. $= \begin{cases} 1 & \text{if } h(s)=h(u) \\ 0 & \text{otherwise} \end{cases}$ $\qquad X = \sum_{s \in S} X_s$

$$E[X] = E\left[\sum_{s \in S} X_s\right] = \sum_{s \in S} E[X_s] = \sum_{s \in S} Pr(h(s)=h(u))$$

$$= 1 + \sum_{s \in S\setminus\{u\}} Pr(h(s)=h(u)) \leq 1 + n/m$$

■

Note: for $m=\Theta(n)$ ⟹ expected $O(1)$ time per operation

Designing a Universal Family of Hash Functions

always exists
[Chebyshev 1850]

Hash Table size: choose m as a **prime** number such that $n \le m \le 2n$

Integer encoding: Identify each element $x \in U$ with a **base-m** integer of r digits: $x = (x_1, x_2, \ldots, x_r)$. The choice of r is given below.

Hash function: for any fixed $a \in U$, $a = (a_1, a_2, \ldots, a_r)$, $a_i \in [m]$ define

$$h_a(x) = \left[\sum_{i=1}^{r} a_i x_i\right] \bmod m \quad (1)$$

Hash-Function Family: $\mathcal{H} = \{h_a : a \in U\}$

$|\mathcal{H}| = m^r = \theta(n^r)$

Parameter Tuning: If $|U| = N$, then r is s.t. $m^r \ge N$ so $r \ge \log N / \log m$

Cost Analysis: to choose and store one h, we need $r = \theta(\log N/\log m)$ digits ($a = (a_1, a_2, \ldots, a_r)$), each one of $\log m$ bits.

Operations (Example): After choosing u.a.r. $a \in U$, to insert 10 elements, computes (1), 10 times (with the same a) and get 10 slots of Table H.

Designing a Universal Family of Hash Functions

always exists
[Chebyshev 1850]

Table size: choose m as a **prime** number such that $n \le m \le 2n$

Integer encoding: Identify each element $x \in U$ with a **base-m** integer of r digits: $x = (x_1, x_2, \ldots, x_r)$.

Hash function:
given $a \in U$, $a = (a_1, a_2, \ldots, a_r)$

$$h_a(x) = \left[\sum_{i=1}^{r} a_i x_i \right] \bmod m$$

hash function family: $\mathcal{H} = \{h_a : a \in U\}$

word-RAM Computational Model:

- manipulating $O(1)$ machine words takes $O(1)$ time
- every object of interest fits in a machine word

$\Rightarrow$

- storing $h_a(x)$ requires just storing a single value, a (1 machine word)
- computing $h_a(x)$ takes $O(1)$ time

THM. $\mathcal{H} = \{h_a : a \in U\}$ is *universal*

proof

Let $x = (x_1, x_2, \dots, x_r)$ and $y = (y_1, y_2, \dots, y_r)$ be two distinct elements of U.
We need to show that $\Pr[h_a(x) = h_a(y)] \leq 1/m$.

since $x \neq y$, there exists an integer j such that $x_j \neq y_j$.

we have $h_a(x) = h_a(y)$ iff

$$a_j \underbrace{(y_j - x_j)}_{z \neq 0} = \underbrace{\sum_{i \neq j} a_i(x_i - y_i)}_{\alpha} \mod m$$

$*$

$a_j \cdot z \cdot z^{-1} = \alpha \cdot z^{-1} \mod m$

$\underbrace{a_j = \alpha \cdot z^{-1}}$ $\mod m$

$\Rightarrow$ unique sol. in $\mathbb{Z}_m$

we can assume a was chosen *u.a.r.* by first selecting all coordinates a_i where $i \neq j$, then selecting a_j at random. Thus, we can assume a_i is fixed for all coordinates $i \neq j$.

since m is prime AND $z \neq 0$, z has a *unique* multiplicative inverse z^{-1}, i.e.

$$z z^{-1} = 1 \mod m$$

$\overset{*}{\Rightarrow} \Pr[a_j =_m \alpha \cdot z^{-1}] \leq 1/m$

$\square$

Another Universal Hash Family (rivedere e correggere bene!!!)

- choose a prime $p \geq |U|$ (once) (elements in U are repres. by numbers in Z_p)
- Hash function: choose $a,b \in Z_p$, and define:

  $h_{ab}(x) = [(ax+b) \bmod p] \bmod m$ (recall m is prime)

  p is a large prime

- Hash function family = $\mathcal{H} = \{ h_{ab} : a,b \in U \}$

- Costs: $p \approx N = |U|$ → basic operation costs = $\Theta(\log N)$

Lemma. $\mathcal{H}$ is *2-wise independent* and *universal.*

$X \neq Y$

Proof. Let $X = (ax+b) \bmod p$ and $Y = (ay+b) \bmod p$ for any $x \neq y$. Since $a \neq 0$ AND $p > N$ → $\underbrace{X \neq Y}_{\to \bmod p}$, so $h_{ab}(x) = h_{ab}(y)$ iff $X = Y \bmod m$.

Claim 1: X and Y are uniformly distributed over Z_p

Proof: a,b are uniformly distributed and h_{ab} is a *linear* (*injective*) function

Claim 2: X,Y are (almost) pairwise independent, i.e., $\Pr[X=i \wedge Y = j] = 1/(p-1)p$ (*) $\forall i, j$

Proof: $\Pr[(ax+b) =_p i \wedge (ay+b) = j] = \Pr[\mathbf{a} = f(x,y,i,j) \wedge \mathbf{b} = g(x,y,i,j)] =$ (**a**,**b** *i.u.r*) *

$\Pr[\mathbf{a} = f(x,y,i,j)] \cdot \Pr[\mathbf{b} = g(x,y,i,j)]$, where **f**, **g** are the <u>unique</u> solutions for the linear system since $x \neq y$. Then, from **Claim 1**, we get (*) $\underbrace{(a \neq 0)}$. * →

$$\to \quad \frac{1}{(p-1) \cdot p}$$

$p \to b \in Z_p$

$(p-1)$: $a \neq 0$, $a \in Z_p$

$X \equiv_p Y$ iff:

$$(1)\ \begin{cases} \underset{\sim}{a}\, x + \underset{\sim}{b} = i \\ \underset{\sim}{a}\, y + \underset{\sim}{b} = j \end{cases}$$

with a, b are unkn. rnd var.

x, y, i, j

Sono i termini noti

Q: How many solutions $(a,b) \in \mathbb{Z}_p^2$ for (1)?

A: Only 1! Indeed the rank of (1) is $\underline{\underline{2}}$ since $x \neq y$.

Another Universal Hash Family

- choose a prime $p \geq |U|$ (once) (elements in U are repres. by numbers in Z_p)
- Hash function: choose $a,b \in Z_p$, and define:

$$h_{ab}(x) = [(ax+b) \bmod p\ ] \bmod m \qquad \text{(recall } m \ll p \text{ is a prime)}$$

Claim 1: X and Y are uniformly distributed over Z_p

Claim 2: X,Y are (almost) pairwise independent, i.e., $\Pr[X=i \wedge Y = j] = 1/(p-1)p$ (*)

$\rightarrow \Pr[Y = j \mid X = i] = \Pr[X=i \wedge Y = j] / \Pr[X=i] = 1 / (p-1)$ (***) ✓ (from (*))

For a fixed i, in Z_p there are <u>at most</u> $\lceil p/m \rceil - 1 \leq (p-1)/m$ values for Y s.t. (*)

$Y = i \bmod m$ (all integers in Z_p whose distance from i is a multiple of m) $\therefore\ i(m+1), i(m+2), \ldots$

From **Claim 2**, $\Pr[Y=j|X=i] = 1/(p-1)$, by Union Bound over all possible values, we get:

$$\Pr[Y = i \bmod m \mid X = i] \leq \underbrace{(p-1)/m}_{*} \cdot \underbrace{1/(p-1)}_{***} = 1/m \quad (**)$$

The universality property then follows by:

$\Pr[h_{ab}(x) = h_{ab}(y)] = \sum_{i=0,\ldots,p-1} \Pr[Y = i \bmod m \mid X = i\ ] \Pr[X = i] \leq$
$\sum_{i=0,\ldots,p-1} (1/p) \cdot 1/m$ (from (**)) $\leq 1/m$

▣ (of Lemma)

<u>CLAIM</u> $\mathcal{H} = \{ h_{a,b}(x) \mid a \in \mathbb{Z}_p - \emptyset ; b \in \mathbb{Z}_p \}$ is a universal Hash family.

<u>Proof</u>: $h_{a,b}(x) = ([ax+b] \bmod p) \bmod m$

Set $X = ax+b \bmod p$ and $Y = ay+b \bmod p$. Then:

CLAIM 1: X and Y are uniform over $\mathbb{Z}_p$. **(1)**

CLAIM 2: $\forall i, j$ PROB$[X \equiv_p i \wedge Y \equiv_p j] = 1/(p-1) \cdot p$ **(2)**

FROM (1), (2) $\Rightarrow$ $\Pr[Y \equiv_p j \mid X \equiv_p i] = \frac{p}{p(p-1)} = 1/p-1$ **(3)**
$\forall$ fixed i, j

Now, we work on $\mathbb{Z}_m$: for fix $i \in \mathbb{Z}_p$, there are at most $(p-1)/m$ values in $\mathbb{Z}_p$ for Y s.t. $Y \equiv_m i$. So; FROM **(3)** and UNION B on all values:

$$\Pr[Y \equiv_m i \mid X \equiv_m i] \leq \boxed{\frac{(p-1)}{m}} \cdot \frac{1}{p-1} = 1/m \quad (4) \quad \forall i \in \mathbb{Z}_p$$

Now, $\forall x \neq y$: $\Pr[h_{a,b}(x) = h_{a,b}(y)] =$

$$= \sum_{i \in \mathbb{Z}_p} \underbrace{\Pr[Y \equiv_m i \mid X \equiv_m i]}_{(4)} \cdot \underbrace{\Pr[X \equiv_m i]}_{(1)} \leq \sum_{i \in \mathbb{Z}_p} \frac{1}{m} \cdot \frac{1}{p}$$

$$= \frac{1}{m} \Rightarrow$$

$\Rightarrow \mathcal{H}$ is UNIVERSAL

$\square$

how to (dynamically) choose the table size

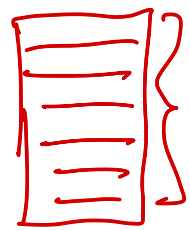
notice: S changes over time and we want to use $O(|S|)$ space

parameters:

- n: # of elements currently in the table, i.e. $n=|S|$;
- N: virtual size of the table/UNIVERSE
- m: actual size of the table (a prime number between N and 2N)

doubling/halving technique:

- init n=N=1;
- whenever n>N:
  - N:=2N
  - choose a new m prime s.t. $m \sim \Theta(n)$
  - re-hash all items (in O(n) time)
- whenever n<N/4:
  - N:=N/2
  - choose a new m
  - re-hash all items (in O(n) time)



m prime

O(1) amortized time per insertion/deletion

Perfect (Randomized) Hashing

(giovedì 18)

optimal static dictionary

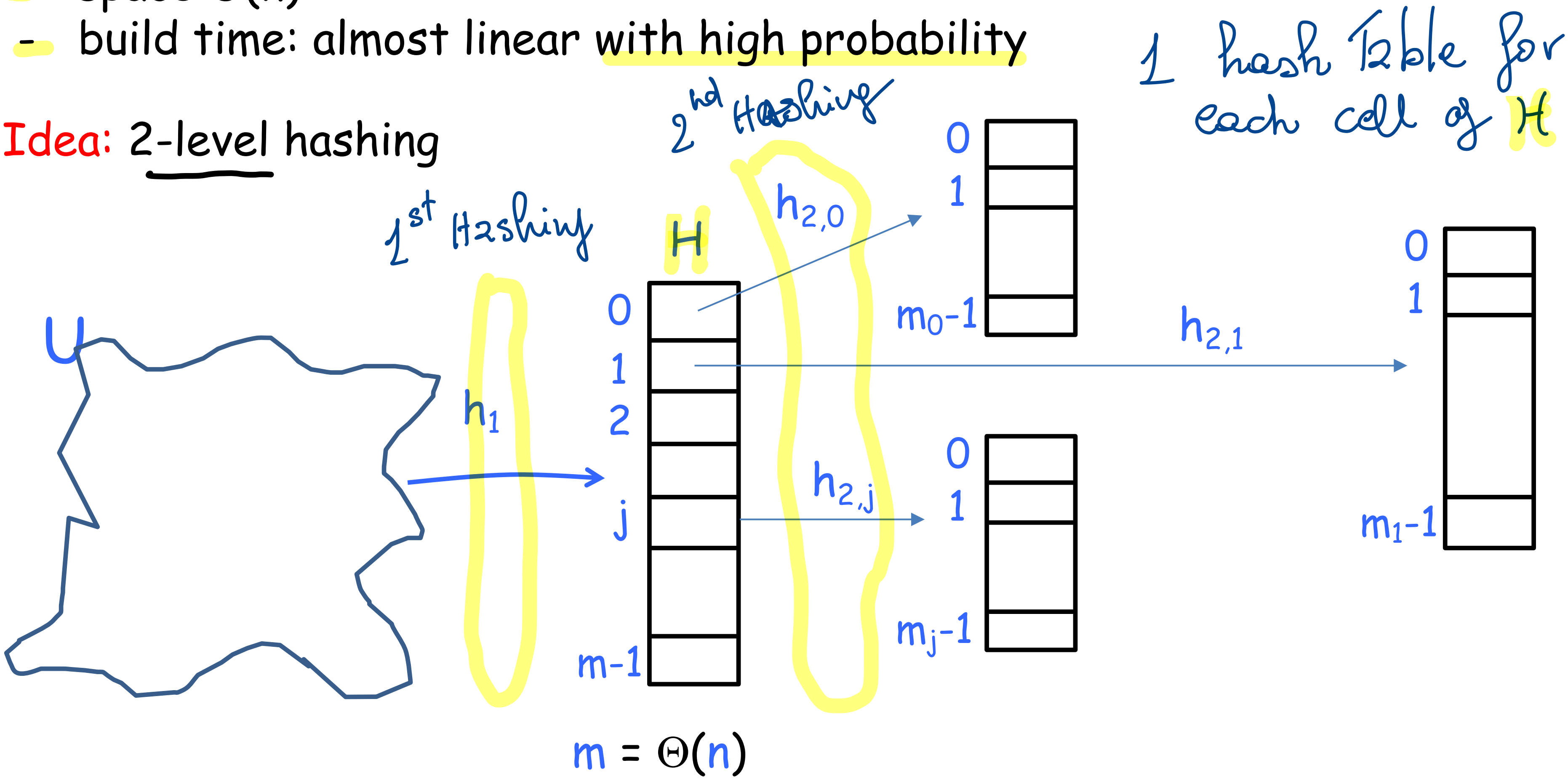
The static dictionary problem:

given a set S of n elements (keys), build a data structure supporting search operations.

Perfect Hashing:

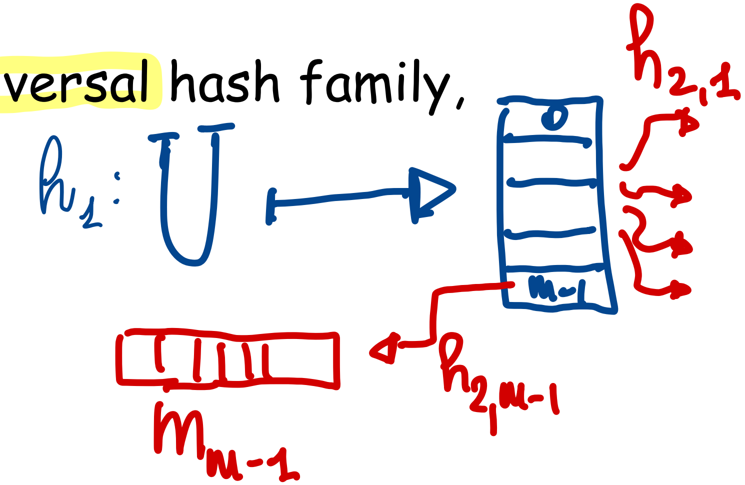
- O(1) worst-case time per search
- space O(n)
- build time: almost linear with high probability

Idea: 2-level hashing



Building the dictionary

- Step 1:
  - pick h_1: $U \longrightarrow \{0,1,\ldots,m-1\}$ u.a.r. from a universal hash family, with $m=\Theta(n)$ (e.g. nearby prime)
  - hash all items with chaining using h_1



Step 2: $h_{2,5}$

for each $j \in \{0,1,\ldots,m-1\}$

- n_j: # of elements mapped to j by h_1
- pick $h_{2,j}$:$U \longrightarrow \{0,1,\ldots, m_j -1\}$ u.a.r. from a universal hash family, with $n_j^2 \le m_j \le O(n_j^2)$
- replace linked list for slot j with a hash table of size m_j using $h_{2,j}$.

Building the dictionary

$n_j \equiv$ # of elements mapped into slot j of the first H.TABLE

Step 1:

- pick $h_1: U \longrightarrow \{0,1,\dots,m-1\}$ u.a.r. from a universal hash family, with $m = \Theta(n)$ (e.g. nearby prime)
- hash all items with chaining using h_1

$m_j \equiv$ size of H.TABLE of cell j

check of "SIZE"

Step 1.5: if $\sum_{j=0}^{m-1} n_j^2 > c\,n$ * for some c (chosen later) redo Step 1

* →

Step 2:

for each $j \in \{0,1,\dots,m-1\}$

- n_j: # of elements mapped to j by h_1
- pick $h_{2,j} : U \longrightarrow \{0,1,\dots, m_j - 1\}$ u.a.r. from a universal hash family, with $n_j^2 \le m_j \le O(n_j^2)$
- replace linked list for slot j with a hash table of size m_j using $h_{2,j}$.

Step 2.5:

while $h_{2,j}(u) = h_{2,j}(v)$ for some $u \neq v$ with $h_1(u) = h_1(v)$
- repick $h_{2,j}$ and re-hash all those n_j elements

collision at 2nd Hash level

[n_j elements into $\Theta(n_j^2)$ cells ?!]

↳ UNLIKELY.

⇒ no collision at second level & linear size

NOTE: if $|S| = n$ then

$\sum_{j=0}^{m-1} n_j = n$ but since we set:

$m_j = \Theta(n_j^2)$ we might have

$$\underbrace{\sum m_j = \sum \Theta(n_j^2) = \underbrace{\omega(n)}_{\text{superlinear}}}_{\to \text{ BAD error}} \; !!$$

Building time

ANALYSIS OF STEP 2.5

Step 1&2 take $O(n)$ time

Step 2.5

$$\Pr_{h_{2,j}} \{ h_{2,j}(u) = h_{2,j}(v), \text{ for some } u \neq v \} \leq$$

$$\sum_{\substack{u,v \in S \\ u \neq v \\ \text{with } h_1(u) = h_1(v)}} \underbrace{\overset{\mathcal{E}_{i,j}}{\Pr \{ h_{2,j}(u) = h_{2,j}(v) \}}}_{\leq 1/n_j^2} \leq \tfrac{1}{2} n_j (n_j - 1) \; \overbrace{1/n_j^2}^{\Pr[\mathcal{E}_{i,j}]} < \tfrac{1}{2}$$

(Universal Hashing on $m = n^2$!!!!!)

$\Rightarrow \Pr[\text{Error}] \leq \sum_{i<j} \Pr[\mathcal{E}_{i,j}] \leq 1/2$

$\Downarrow$

1 TRIAL

- for each j:
  - $E[\#\text{ trials}] \leq 2$
  - $O(\log n)$ trials w.h.p.
  - each trial takes $O(n_j)$ time

fixed H_1

slot j

time for Step 2.5: all j's!

$$\sum_j (\#\text{ trials for } j)\, O(n_j) \Rightarrow \boxed{\begin{array}{c} O(n \log n) \\ \text{with high probability} \end{array}}$$

Building time (Step 1.5)

Expectation !!

Idea: we show that $E\left[\sum_{j=0}^{m-1} n_j^2\right] = \Theta(n)$ and then we use Markov's inequality

collision

$$X_{u,v} \text{ r. v.} = \begin{cases} 1 & \text{if } h_1(u)=h_1(v) \\ 0 & \text{otherwise} \end{cases}$$

FACT A

$$\sum_{j=0}^{m-1} n_j^2 = \sum_{u\in S} \sum_{v\in S} X_{u,v}$$

Proof

$* \Rightarrow$

$m \geq n$

$$E\left[\sum_{j=0}^{m-1} n_j^2\right] = \sum_{u\in S} \sum_{v\in S} E[X_{u,v}] = \sum_{u\in S} \sum_{v\in S} \left(\Pr\{h_1(u)=h_1(v)\}\right) \leq n + n^2/m \leq 2n$$

UNIV. H.S

THM 1 $\rightarrow$ $\boxed{1 + \frac{n}{m}}$

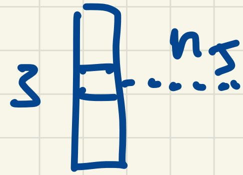
by suitably choosing c $\rightarrow c \geq 4$

$$\Pr\left\{\sum_{j=0}^{m-1} n_j^2 > c\, n\right\} \leq \frac{E\left[\sum_{j=0}^{m-1} n_j^2\right]}{c\, n} \leq \frac{2n}{c\, n} \leq 1/2$$

- $E[\#\text{ trials}] \leq 2$
- $O(\log n)$ trials w.h.p.
- each trial takes $O(n)$

$O(n \log n)$
with high probability

$$X_{u,v} = \begin{cases} 1 & \text{se } h_1(u) = h_1(v) \\ 0 & \text{o.w} \end{cases}$$



$n_j \equiv$ # elements in slot j

$$\sum_{u \in S} \sum_{v \in S} X_{u,v} = \underbrace{n_1 \times n_1}_{\text{slot } 1} + \underbrace{n_2 \times n_2}_{\text{SLOT } 2} + \dots + \underbrace{n_m \times n_m}_{\text{SLOT } m-1}$$

i.e. :

Consider all elements $z_1 \dots z_{n_j}$ mapped to slot j

Consider all pairs $z_e, z_m \Rightarrow$

all r.v. $X_{e,m}$ are equal 1 $\quad e = 1 \dots n_j$, $m = 1 \dots n_j$

How many are? n_j^2

Building time (Step 1.5)

Idea: we show that $E\left[\sum_{j=0}^{m-1} n_j^2\right] = \Theta(n)$ and then we use Markov's inequality

$$X_{u,v} \text{ r. v.} = \begin{cases} 1 & \text{if } h_1(u)=h_1(v) \\ 0 & \text{otherwise} \end{cases} \qquad \sum_{j=0}^{m-1} n_j^2 = \sum_{u\in S}\sum_{v\in S} X_{u,v}$$

$$E\left[\sum_{j=0}^{m-1} n_j^2\right] = \sum_{u\in S}\sum_{v\in S} E[X_{u,v}] = \sum_{u\in S}\sum_{v\in S} \Pr\{h_1(u)=h_1(v)\} \le n + n^2/m \le 2n$$

$$\Pr\left\{\sum_{j=0}^{m-1} n_j^2 > c\,n\right\} \le \frac{E\left[\sum_{j=0}^{m-1} n_j^2\right]}{c\,n} \le \frac{2n}{c\,n} \le 1/2$$

by suitably choosing c

- E[# trials]≤2
- O(log n) trials w.h.p.
- each trial takes O(n)

$\Rightarrow$ $\boxed{\text{O(n log n) with high probability}}$