Randomness in Computing

CS 537

Lecture 18

Last time

- Finding Hamiltonian cycles in random graphs

Today

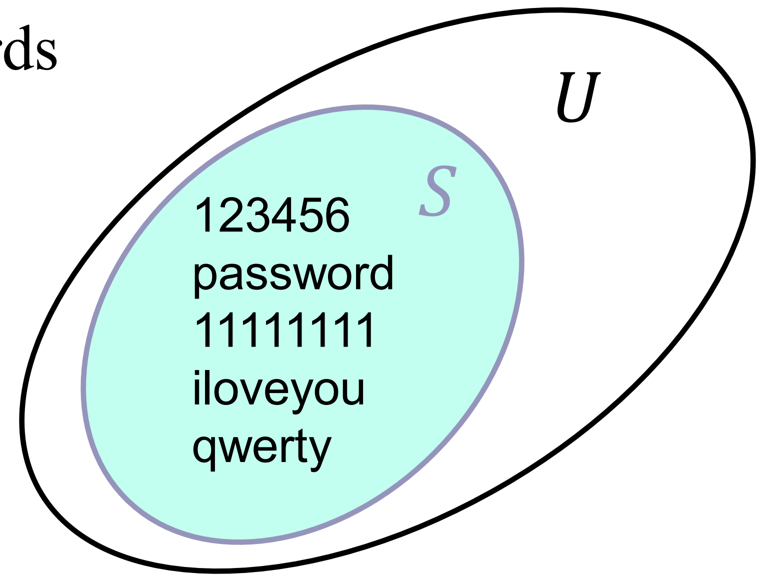
- Hashing

Static dictionary problem

Motivating example

Password checker to prevent people from using common passwords.

- S is the set of common passwords



- **Universe:** set U
- $S \subseteq U$ and $m = |S|$
- $m \ll |U|$

Goal: A data structure for storing $\boldsymbol{S}$ that supports the search query

"*Does* $w \in \boldsymbol{S}$?" for all words $w \in \boldsymbol{U}$.

Solutions

Deterministic solutions

- Store $\boldsymbol{S}$ as a sorted array (or as a binary search tree)

Search time: **O**($\log m$), **Space**: **O**(m)

- Store an array that for each $w \in \boldsymbol{U}$ has 1 if $w \in \boldsymbol{S}$ and 0 otherwise.

Search time: O(1), **Space**: **O**($|\boldsymbol{U}|$)

A randomized solution

- Hashing

Chain Hashing

- **Hash table:** n bins, words that fall in the same bin are chained into a linked list.
- **Hash function:** $h : U \rightarrow [n]$

To construct the table

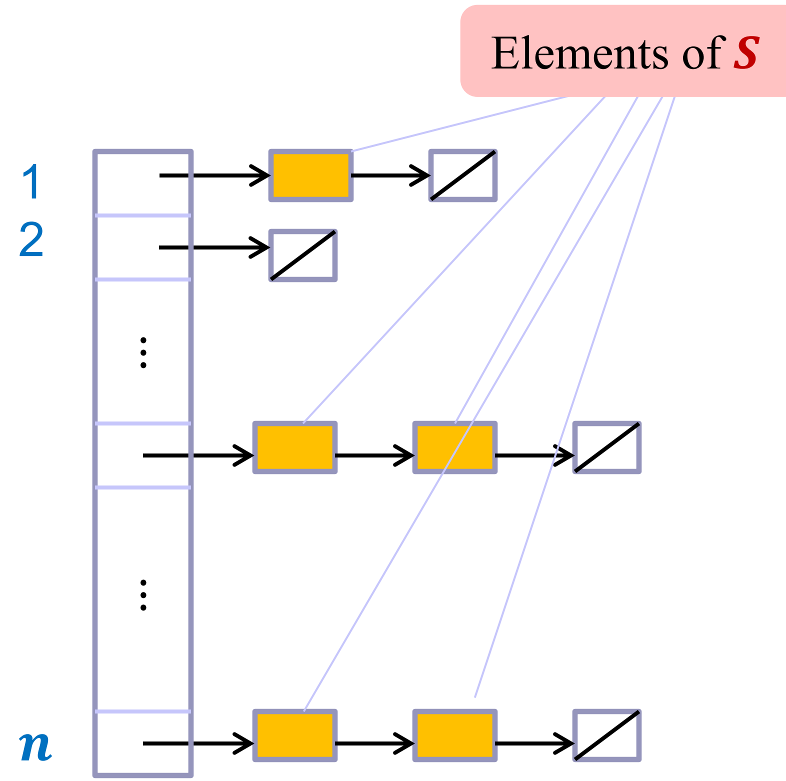
hash all elements of S

To search for word w

check if w is in bin $h(w)$

Desiderata for h:

- O(1) evaluation time.
- O(1) space to store h.



A random hash function

- Simplifying assumption: hash function h is selected at random:

$$\Pr[h(w) = j] = \frac{1}{n} \text{ for all } w \in U, j \in [n]$$

- Once h is chosen, every evaluation of h yields the same answer.

Search time:

- If $w \notin S$, expected number of words in bin $h(w)$ is
- If $w \in S$, expected number of words in bin $h(w)$ is

If we set $n = m$, then

- the expected search time is O(1)
- max time to search is max load: w.p. close to 1, it is $\Theta\left(\frac{\ln m}{\ln \ln m}\right)$

Faster than a search tree, with space still $\Theta(m)$.

Are we done?

- How many hash functions are there?
- How many bits do we need to store a description of a hash function?

This is prohibitively expensive!

- Idea: Choose from a smaller family of hash functions.

Universal hash family

- A set $\mathcal{H}$ of hash functions is **universal** if for every pair $w_1, w_2 \in U$ and for h chosen uniformly from $\mathcal{H}$

$$\Pr[h(w_1) = h(w_2)] \leq \frac{1}{n}$$

Constructing a universal hash family

$\boldsymbol{U = \{0, 1, \dots, u-1\}}$

- Fix a prime $p \geq |U|$ and think of the range as $\{0,1,\dots,n-1\}$.
- Define $\boldsymbol{h_{a,b}(x)} = \big((ax+b) \bmod p\big) \bmod n$

$$\mathcal{H} = \left\{h_{a,b} \,\middle|\, a \in [p-1], 0 \leq b \leq p-1\right\}$$

Theorem

$\mathcal{H}$ is universal.

Proof that $\mathcal{H}$ is universal

- Define $\boldsymbol{h_{a,b}(x)} = \big((ax+b)\ mod\ p\big)\ mod\ n$

$$\mathcal{H} = \{h_{a,b} \mid a \in [p-1], 0 \le b \le p-1\}$$

Proof: Fix $x_1 \neq x_2$ from U.

- **Idea:** count # of $h_{a,b}$ in $\mathcal{H}$ for which x_1, x_2 collide.
- We will show that
  - They can't collide after performing mod p.
  - So, they must map to different values v_1, v_2 at this point
  - Each (v_1, v_2) corresponds to a unique pair (a, b).
  - So, it suffices to count the number of pairs (v_1, v_2) with $v_1 \neq v_2$, but $v_1 \equiv v_2\ (mod\ n)$

Proof that $\mathcal{H}$ is universal

- Define $\boldsymbol{h_{a,b}(x)} = \big((ax + b)\ mod\ p\big)\ mod\ n$

$$\mathcal{H} = \{h_{a,b} \mid a \in [p-1], 0 \le b \le p-1\}$$

Claim 1. If $x_1 \neq x_2$ then $ax_1 + b \neq ax_2 + b \pmod{p}$.

Proof that $\mathcal{H}$ is universal

- Define $\boldsymbol{h_{a,b}(x)} = \big((ax+b) \; mod \; p\big) \; mod \; n$

$$\mathcal{H} = \{h_{a,b} \mid a \in [p-1], 0 \le b \le p-1\}$$

Claim 1. If $x_1 \neq x_2$ then $ax_1 + b \neq ax_2 + b \pmod p$.

Claim 2. For every pair (v_1, v_2), where $v_1 \neq v_2$ and $0 \le v_1, v_2 \le p-1$,
$\exists$ exactly one pair (a, b):
$$ax_1 + b \equiv v_1 \pmod p;$$
$$ax_2 + b \equiv v_2 \pmod p.$$

Proof that $\mathcal{H}$ is universal

- Define $\boldsymbol{h_{a,b}(x)} = \big((ax+b) \bmod p\big) \bmod n$

$$\mathcal{H} = \{h_{a,b} \mid a \in [p-1], 0 \le b \le p-1\}$$

Using a universal family

As before:

- If $w \notin S$, expected number of words in bin $h(w)$ is $\leq \frac{m}{n}$
- If $w \in S$, expected number of words in bin $h(w)$ is $\leq 1 + \frac{m-1}{n}$

The previous guarantee on max load no longer holds!

Goal: Given S, find a hash function with no collisions for words in S.

Recall: Two elements $w_1, w_2 \in \boldsymbol{U}$ *collide* under a hash function $\boldsymbol{h}$ if

$$\boldsymbol{h}(w_1) = \boldsymbol{h}(w_2).$$

A hash function $\boldsymbol{h}$ is *perfect* for set S
if no elements of S collide under $\boldsymbol{h}$.

Perfect hashing: no collisions

Theorem

If $h: U \to \{0,1,\dots,n-1\}$ is chosen uniformly at random from a universal hash family, then $\forall S$ of size m, such that $n \geq m^2$,

$$\Pr[h \text{ is perfect for } S] \geq 1/2.$$

Proof: Let $s_1, \dots, s_m$ be elements of S.

- Let $X_{ij} = \begin{cases} 1 & \text{if } h(s_i) = h(s_j) \\ 0 & \text{otherwise} \end{cases}$

$X = $ # of collisions $= \sum_{i,j \in [m], i<j} X_{ij}$

- $\mathbb{E}[X] \overset{\text{Linearity of expectation}}{=} \sum_{i,j \in [m], i<j} \mathbb{E}[X_{ij}] \overset{\text{symmetry}}{=} \binom{m}{2} \mathbb{E}[X_{12}] \overset{X_{12} \text{ is an indicator}}{=} \binom{m}{2} \Pr[h(s_1) = h(s_2)]$

$\overset{h \text{ is universal}}{\leq} \binom{m}{2} \frac{1}{n} < \frac{m^2}{2n}$

- $\Pr[X \geq 1] \overset{\text{since } n \geq m^2}{\leq} \Pr\left[X \geq \frac{m^2}{n}\right] \overset{\text{by Markov}}{\leq} \frac{\mathbb{E}[X]}{m^2/n}$

Perfect hashing

Theorem

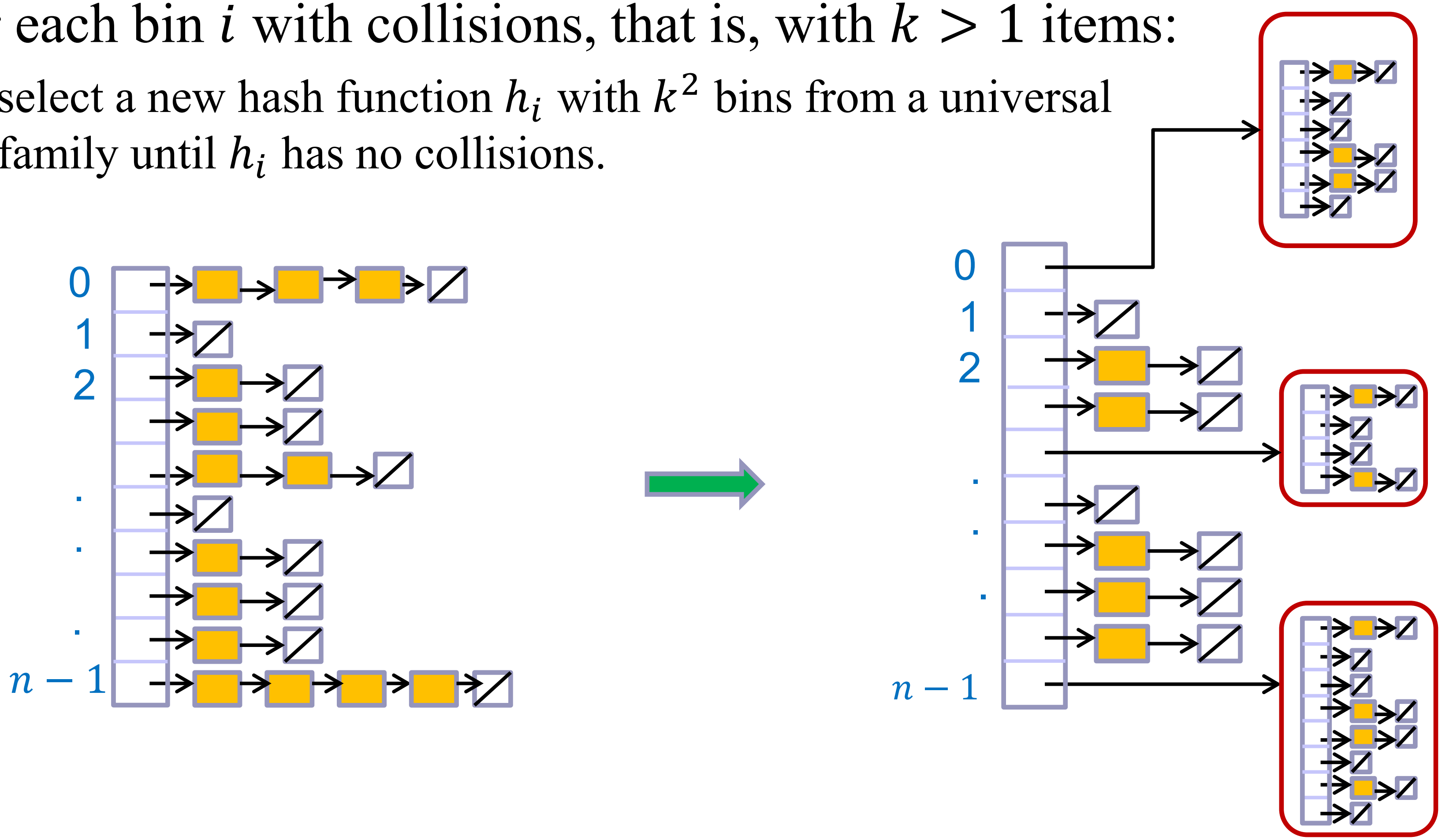
If $h: U \to \{0,1,\ldots,n-1\}$ is chosen uniformly at random from a universal hash family, then $\forall S$ of size m, such that $n \geq m^2$,

$$\Pr[h \text{ is perfect for } S] \geq 1/2.$$

- Select $h \in \mathcal{H}$ until a perfect h for a given S is found.
- Expected number of tries is at most 2.
- Each try takes $O(m)$ time.
- **Drawback:** $\Omega(m^2)$ space.

2-level scheme for perfect hashing

- Set $n = m$.
- Select $h \in \mathcal{H}$ until h with at most m collisions is found.
- For each bin i with collisions, that is, with $k > 1$ items:
  - select a new hash function h_i with k^2 bins from a universal family until h_i has no collisions.



2-level scheme for perfect hashing

- Set $n = m$.
- Select $h \in \mathcal{H}$ until h with at most m collisions is found.
- For each bin i with collisions, that is, with $k > 1$ items:
  - select a new hash function h_i with k^2 bins from a universal family until h_i has no collisions.

Theorem

2-level scheme achieves perfect hashing with $O(m)$ space.

A solution for static dictionary problem with:

- O(1) worst case guarantee on search time.
- O($\boldsymbol{m}$) space.
- Expected O($\boldsymbol{m}$) preprocessing time.

Analysis of 2-level scheme

Theorem

2-level scheme achieves perfect hashing with $O(m)$ space.

Proof:

- Let $X = \#$ of collisions in Stage 1.
- We showed before: $\Pr\left[X \geq \frac{m^2}{n}\right] \leq \frac{1}{2}$.
- Now $n = m$: $\Pr[X \geq m] \leq \frac{1}{2}$.
- So at least half of $h \in \mathcal{H}$ have $\leq m$ collisions.
- Assume we found such h.

Analysis of 2-level scheme

Theorem

2-level scheme achieves perfect hashing with $O(m)$ space.

Proof (continued): Assume we found $h \in \mathcal{H}$ with $\leq m$ collisions.

- Let $k_i =$ number of items in bin i.
- Then # of collisions between items in bin i is

Conclusion: 2-level hashing

A solution for static dictionary problem with:

- O(**1**) worst case guarantee on search time.
- O($\boldsymbol{m}$) space.
- Expected O($\boldsymbol{m}$) preprocessing time.