COMPSCI 514: Algorithms for Data Science

Cameron Musco

University of Massachusetts Amherst. Fall 2024.

Lecture 3

Logistics

- Problem Set 1 was posted this morning on the course website and is due **Friday 9/20 at 11:59pm**.
- Three core problems and two challenge problems, pick one.
- On the quiz a large number of people cited concerns about their linear algebra and/or probability background. For linear algebra you have time to review – we will mostly use it after the midterm. For probability, come to my office hours today and next week if you would like to more review.
- I also highly recommend looking at the exercises in *Foundations of Data Science* and *Probability and Computing*. Feel free to ask for solutions to these on Piazza.
- It is common to not catch everything in lecture. I strongly encourage going back to the slides to review/check your understanding after class. Also come to office hours for more in-depth discussion/examples.

Content Overview

Last Class:

- Linearity Expectation
- Linearity of variance.
- Markov's inequality: the most fundamental concentration bound. $\Pr(X \geq t \cdot \mathbb{E}[X]) \leq 1/t$.
- Algorithmic applications of Markov's inequality, linearity of expectation, and indicator random variables:
    - Counting collisions to estimate CAPTCHA database size.
    - Start on analyzing hash tables with random hash functions.

Content Overview

Today:

- Finish up random hash functions and hash tables.
- Collision free hashing using a table with $O(m^2)$ slots to store m items.
- 2-level hashing, 2-universal and pairwise independent hash functions.
- Maybe start on application of random hashing to distributed load balancing.
- Through this application learn about Chebyshev's inequality, which strengthens Markov's inequality.

Quiz Questions

linearity of expectation

5 1 point

The expected number of inches of rain on Saturday is 5.8 and the expected number of inches on Sunday is 6.9. There is a 50% chance of rain on Saturday. If it rains on Saturday, there is a 75% chance of rain on Sunday. If it does not rain on Saturday, there is only a 25% chance of rain on Sunday. What is the expected number of inches of rainfall total over the weekend?

Type your answer...

S_a = # inches on Sat.

S_u = # inches on Sun

$$\mathbb{E}[W] = \mathbb{E}[S_a + S_u] = \mathbb{E}[S_a] + \mathbb{E}[S_u]$$
$$= 5.8 + 6.9 = 12.7$$

15 1 point

$\text{Bin}(n, p)$, $n = 1000$, $p = \frac{1}{3921}$ — $n \cdot p = \frac{1000}{3921}$

You store 1000 items in a hash table with 3,921 buckets. You use a fully random hash function to implement the table. What is the expected number of items stored in bucket i? Enter your answer to two decimal places.

Type your answer...

$\frac{1000}{3921}$

S_i = # items in bucket i

$$\mathbb{E}[S_i] = \mathbb{E}\sum_{j=1}^{1000} S_{ij}$$

$$\mathbb{E}[S_i] = \sum_{j=1}^{1000} \mathbb{E}[S_{ij}] = \sum_{j=1}^{1000} \frac{1}{3921} = \frac{1000}{3921}$$

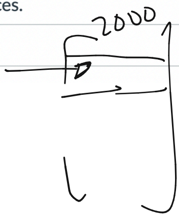
item j lands in bucket i

16 1 point

As above, you store 1000 items in a hash table, this time with 2,374 buckets. You use a fully random hash function to implement the table. What is the total expected number of pairwise collisions across the hash table? Enter your answer to two decimal places.

Type your answer...

$$\binom{1000}{2} \cdot \frac{1}{2372} = ?$$



$\frac{1}{2000^2} \cdot 2000 = \frac{1}{2000}$

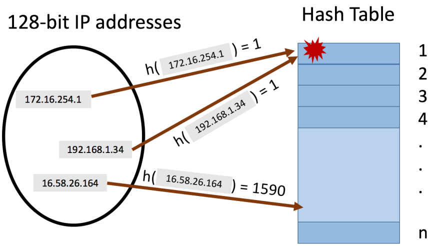
$$\mathbb{E}C = \sum_{i<j} D_{ij}$$

items i and j collide

Quiz Questions

Hash Tables

We store m items from a large universe in a hash table with n positions.



- Want to show that when $\mathsf{h} : U \to [n]$ is a fully random hash function, query time is $O(1)$ with good probability.
- Equivalently: want to show that there are few collisions between hashed items.

Collision Free Hashing

Let $\mathsf{C} = \sum_{i,j \in [m], i<j} \mathsf{C}_{i,j}$ be the number of pairwise collisions between items.

$$\mathbb{E}[\mathsf{C}] = \frac{m(m-1)}{2n} \quad \text{(via the Captcha analysis)}$$

m: total number of stored items, n: hash table size, C: total pairwise collisions in table.

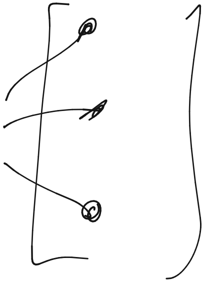
Collision Free Hashing

Let $C = \sum_{i,j \in [m], i<j} C_{i,j}$ be the number of pairwise collisions between items.

$$\mathbb{E}[C] = \frac{m(m-1)}{2n} \quad \text{(via the Captcha analysis)}$$

- For $n = 4m^2$ we have: $\mathbb{E}[C] = \frac{m(m-1)}{8m^2} \leq \frac{1}{8}$.

$\Pr(\text{have a collision}) \leq \Pr(C \geq 1) \leq \frac{\mathbb{E}[C]}{1} = \frac{1}{8}$



> m: total number of stored items, n: hash table size, C: total pairwise collisions in table.

Collision Free Hashing

Let $C = \sum_{i,j \in [m], i<j} C_{i,j}$ be the number of pairwise collisions between items.

$$\mathbb{E}[C] = \frac{m(m-1)}{2n} \quad \text{(via the Captcha analysis)}$$

- For $n = 4m^2$ we have: $\mathbb{E}[C] = \frac{m(m-1)}{8m^2} \leq \frac{1}{8}$.

Apply Markov's Inequality: $\Pr[C \geq 1] \leq \frac{\mathbb{E}[C]}{1} = \frac{1}{8}$.

m: total number of stored items, *n*: hash table size, C: total pairwise collisions in table.

Collision Free Hashing

Let $\mathsf{C} = \sum_{i,j \in [m], i<j} \mathsf{C}_{i,j}$ be the number of pairwise collisions between items.

$$\mathbb{E}[\mathsf{C}] = \frac{m(m-1)}{2n} \quad \text{(via the Captcha analysis)}$$

- For $n = 4m^2$ we have: $\mathbb{E}[\mathsf{C}] = \frac{m(m-1)}{8m^2} \leq \frac{1}{8}$.

Apply Markov's Inequality: $\Pr[\mathsf{C} \geq 1] \leq \frac{\mathbb{E}[\mathsf{C}]}{1} = \frac{1}{8}$.

$$\Pr[\mathsf{C} = 0] = 1 - \Pr[\mathsf{C} \geq 1]$$

m: total number of stored items, *n*: hash table size, C: total pairwise collisions in table.

Collision Free Hashing

Let $C = \sum_{i,j \in [m], i<j} C_{i,j}$ be the number of pairwise collisions between items.

$$\mathbb{E}[C] = \frac{m(m-1)}{2n} \quad \text{(via the Captcha analysis)}$$

- For $n = 4m^2$ we have: $\mathbb{E}[C] = \frac{m(m-1)}{8m^2} \leq \frac{1}{8}$.

Apply Markov's Inequality: $\Pr[C \geq 1] \leq \frac{\mathbb{E}[C]}{1} = \frac{1}{8}$.

$$\Pr[C = 0] = 1 - \Pr[C \geq 1] \geq 1 - \frac{1}{8} = \frac{7}{8}$$

m: total number of stored items, n: hash table size, C: total pairwise collisions in table.

Collision Free Hashing

Let $\mathbf{C} = \sum_{i,j \in [m], i<j} \mathbf{C}_{i,j}$ be the number of pairwise collisions between items.

$$\mathbb{E}[\mathbf{C}] = \frac{m(m-1)}{2n} \quad \text{(via the Captcha analysis)}$$

$\frac{m(m-1)}{m^2} < 1$

- For $n = 4m^2$ we have: $\mathbb{E}[\mathbf{C}] = \frac{m(m-1)}{8m^2} \leq \frac{1}{8}$.

Apply Markov's Inequality: $\Pr[\mathbf{C} \geq 1] \leq \frac{\mathbb{E}[\mathbf{C}]}{1} = \frac{1}{8}$.

$$\Pr[\mathbf{C} = 0] = 1 - \Pr[\mathbf{C} \geq 1] \geq 1 - \frac{1}{8} = \frac{7}{8}$$

I.e., with probability at least 7/8 we have no collisions and thus $O(1)$ query time. But we are using $O(m^2)$ space to store m items...

> m: total number of stored items, n: hash table size, $\mathbf{C}$: total pairwise collisions in table.

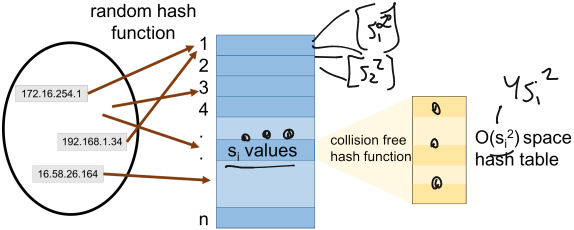
Two Level Hashing

Want to preserve $O(1)$ query time while using $O(m)$ space.

Two Level Hashing

Want to preserve $O(1)$ query time while using $O(m)$ space.

Two-Level Hashing:

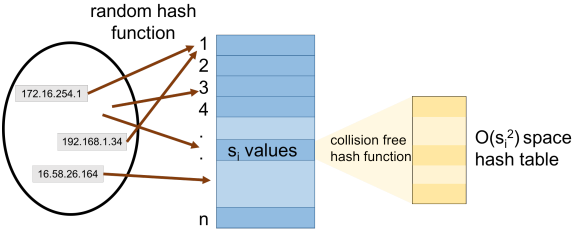


Space used by this data structure:

$$O\left(n + \sum_{i=1}^{n} s_i^2\right)$$

Want to preserve $O(1)$ query time while using $O(m)$ space.

Two-Level Hashing:

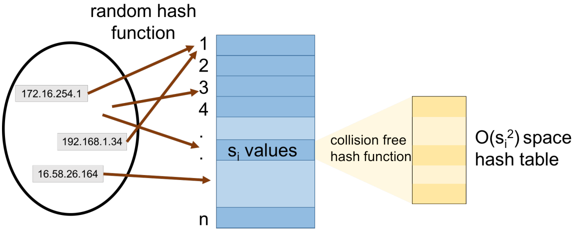


- For each bucket with s_i values, pick a collision free hash function mapping $[s_i] \to [s_i^2]$.

Two Level Hashing

Want to preserve $O(1)$ query time while using $O(m)$ space.

Two-Level Hashing:



- For each bucket with s_i values, pick a collision free hash function mapping $[s_i] \to [s_i^2]$.
- **Just Showed:** A random function is collision free with probability $\geq \frac{7}{8}$ so can just generate a random hash function and check if it is collision free.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions.

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbf{S} = n + \sum_{i=1}^{n} \mathbf{s}_i^2$ $O(n)$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\underbrace{\sum_{j=1}^{m} \underbrace{\mathbb{I}_{\mathbf{h}(x_j)=i}}_{s_{ij}}}_{s_i}\right)^2\right]$$

$$\mathbb{E}[s_i^2] = \sum_{c=0}^{n} \Pr(s_i = c) \cdot c^2$$

$$s_i = \sum_{j=1}^{m} \mathbb{I}_{h(x_j)=i}$$

$\mathbb{I}_{h(x_j)=i}$ is 1 if item j is in bucket i, 0 otherwise

s_i: # items in bucket i

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right]$$

Collisions again!

$$(I_1 + I_2 + I_3)(I_1 + I_2 + I_3) = I_1 I_1 + I_1 I_2 \ldots$$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right].$$

indicator for collision

0 1

$\mathbb{I}_{h(x_j)=i \cap h(x_k)=i}$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right].$$

- For $j = k$,

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right].$$

- For $j = k$,

$$\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \mathbb{E}\left[\left(\mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right].$$

- For $j = k$,

$$\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \mathbb{E}\left[\left(\mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right] = \Pr[\mathbf{h}(x_j) = i]$$

$$= \mathbb{E}\left(\mathbb{I}_{h(x_j)=i}\right)$$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right].$$

- For $j = k$,

$$\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \mathbb{E}\left[\left(\mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right] = \Pr[\mathbf{h}(x_j) = i] = \frac{1}{n}.$$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right].$$

- For $j = k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \mathbb{E}\left[\left(\mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right] = \Pr[\mathbf{h}(x_j) = i] = \frac{1}{n}$.
- For $j \neq k$,

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right].$$

- For $j = k$,
  $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \mathbb{E}\left[\left(\mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right] = \Pr[\mathbf{h}(x_j) = i] = \frac{1}{n}.$
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right]$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right].$$

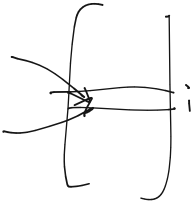
- For $j = k$,
  $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \mathbb{E}\left[\left(\mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right] = \Pr[\mathbf{h}(x_j) = i] = \frac{1}{n}.$
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \Pr[\mathbf{h}(x_j) = i \cap \mathbf{h}(x_k) = i]$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

Query time for two level hashing is $O(1)$: requires evaluating two hash functions. What is the expected space usage?

Up to constants, space used is: $\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$



$$\mathbb{E}[\mathbf{s}_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^{m} \mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j,k\in[m]} \mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right].$$

$\mathbb{E}[\ldots] = Pr[I_j \cdot I_k = 1] \cdot 1 + Pr[I_j \cdot I_k = 0] \cdot 0$

- For $j = k$,
  $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \mathbb{E}\left[\left(\mathbb{I}_{\mathbf{h}(x_j)=i}\right)^2\right] = \Pr[\mathbf{h}(x_j) = i] = \frac{1}{n}.$
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \Pr[\mathbf{h}(x_j) = i \cap \mathbf{h}(x_k) = i] = \frac{1}{n^2}.$
  $= Pr[h(x_j) = i] \cdot Pr(h(x_k) = i) = \frac{1}{n} \cdot \frac{1}{n} = \frac{1}{n^2}$

x_j, x_k: stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored in hash table at position i.

Space Usage

$$\mathbb{E}[\mathsf{s}_i^2] = \sum_{j,k \in [m]} \mathbb{E}\left[\mathbb{I}_{\mathsf{h}(x_j)=i} \cdot \mathbb{I}_{\mathsf{h}(x_k)=i}\right]$$

- For $j = k$, $\mathbb{E}\left[\mathbb{I}_{\mathsf{h}(x_j)=i} \cdot \mathbb{I}_{\mathsf{h}(x_k)=i}\right] = \frac{1}{n}$.
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathsf{h}(x_j)=i} \cdot \mathbb{I}_{\mathsf{h}(x_k)=i}\right] = \frac{1}{n^2}$.

x_j, x_k: stored items, m: # stored items, n: hash table size, h: random hash function, S: space usage of two level hashing, s_i: # items stored at pos i.

Space Usage

$$\mathbb{E}[\mathsf{s}_i^2] = \sum_{j,k\in[m]} \mathbb{E}\left[\mathbb{I}_{\mathsf{h}(x_j)=i} \cdot \mathbb{I}_{\mathsf{h}(x_k)=i}\right]$$
$$= m \cdot \frac{1}{n} + 2 \cdot \binom{m}{2} \cdot \frac{1}{n^2}$$

- For $j = k$, $\mathbb{E}\left[\mathbb{I}_{\mathsf{h}(x_j)=i} \cdot \mathbb{I}_{\mathsf{h}(x_k)=i}\right] = \frac{1}{n}$.
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathsf{h}(x_j)=i} \cdot \mathbb{I}_{\mathsf{h}(x_k)=i}\right] = \frac{1}{n^2}$.

x_j, x_k: stored items, m: # stored items, n: hash table size, h: random hash function, S: space usage of two level hashing, s_i: # items stored at pos i.

Space Usage

$$\mathbb{E}[\mathbf{s}_i^2] = \sum_{j,k \in [m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right]$$

$$= m \cdot \frac{1}{n} + 2 \cdot \binom{m}{2} \cdot \frac{1}{n^2}$$

m(m-1)

- For $j = k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n}$.
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n^2}$.

x_j, x_k: stored items, m: # stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored at pos i.

Space Usage

$$\mathbb{E}[\mathbf{s}_i^2] = \sum_{j,k \in [m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right]$$

$$= m \cdot \frac{1}{n} + 2 \cdot \binom{m}{2} \cdot \frac{1}{n^2}$$

m^2 terms

$m + 2\binom{m}{2} = m^2$

$m + m(m-1) = m^2$

$m = 3$

$(I_1 + I_2 + I_3)^2$

$I_1 I_2 + \ldots + I_2 I_1$

- For $j = k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n}$.
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n^2}$.

x_j, x_k: stored items, m: # stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored at pos i.

Space Usage

$$\mathbb{E}[\mathsf{s}_i^2] = \sum_{j,k \in [m]} \mathbb{E}\left[\mathbb{I}_{\mathsf{h}(x_j)=i} \cdot \mathbb{I}_{\mathsf{h}(x_k)=i}\right]$$
$$= m \cdot \frac{1}{n} + 2 \cdot \binom{m}{2} \cdot \frac{1}{n^2}$$
$$= \frac{m}{n} + \frac{m(m-1)}{n^2}$$

- For $j = k$, $\mathbb{E}\left[\mathbb{I}_{\mathsf{h}(x_j)=i} \cdot \mathbb{I}_{\mathsf{h}(x_k)=i}\right] = \frac{1}{n}$.
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathsf{h}(x_j)=i} \cdot \mathbb{I}_{\mathsf{h}(x_k)=i}\right] = \frac{1}{n^2}$.

x_j, x_k: stored items, m: # stored items, n: hash table size, h: random hash function, S: space usage of two level hashing, s_i: # items stored at pos i.

Space Usage

$$\mathbb{E}[\mathsf{s}_i^2] = \sum_{j,k \in [m]} \mathbb{E}\left[\mathbb{I}_{\mathsf{h}(x_j)=i} \cdot \mathbb{I}_{\mathsf{h}(x_k)=i}\right]$$

$$= m \cdot \frac{1}{n} + 2 \cdot \binom{m}{2} \cdot \frac{1}{n^2}$$

$$= \frac{m}{n} + \frac{m(m-1)}{n^2} \leq 2 \text{ if we set } n = m.$$

O(m)

- For $j = k$, $\mathbb{E}\left[\mathbb{I}_{\mathsf{h}(x_j)=i} \cdot \mathbb{I}_{\mathsf{h}(x_k)=i}\right] = \frac{1}{n}$.
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathsf{h}(x_j)=i} \cdot \mathbb{I}_{\mathsf{h}(x_k)=i}\right] = \frac{1}{n^2}$.

x_j, x_k: stored items, m: # stored items, n: hash table size, h: random hash function, S: space usage of two level hashing, s_i: # items stored at pos i.

Space Usage

$$\mathbb{E}[\mathsf{s}_i^2] = \sum_{j,k \in [m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right]$$

$$= m \cdot \frac{1}{n} + 2 \cdot \binom{m}{2} \cdot \frac{1}{n^2}$$

$$= \frac{m}{n} + \frac{m(m-1)}{n^2} \leq 2 \text{ if we set } n = m.$$

- For $j = k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n}$.
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n^2}$.

Total Expected Space Usage: (if we set $n = m$)

$$\mathbb{E}[\mathsf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathsf{s}_i^2]$$

x_j, x_k: stored items, m: # stored items, n: hash table size, $\mathbf{h}$: random hash function, S: space usage of two level hashing, s_i: # items stored at pos i.

Space Usage

$$\mathbb{E}[\mathbf{s}_i^2] = \sum_{j,k \in [m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right]$$

$$= m \cdot \frac{1}{n} + 2 \cdot \binom{m}{2} \cdot \frac{1}{n^2}$$

$$= \frac{m}{n} + \frac{m(m-1)}{n^2} \leq 2 \text{ if we set } n = m.$$

- For $j = k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n}$.
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n^2}$.

Total Expected Space Usage: (if we set $n = m$)

$$\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2] \leq n + n \cdot 2 = 3n = 3m.$$

m^2 terms total

m terms w/ $j=k$

m^2-m terms w/ $j \neq k$

$= m(m-1) = 2 \cdot \binom{m}{2}$

$\sum_{j \in [m]} \sum_{k \in [m]} \mathbb{E}[\mathbb{I}_j \cdot \mathbb{I}_k]$

how many $j,k \in [m]$ s.t. $j \neq k$

$\Pr[S > 6m] \leq \frac{1}{2}$

x_j, x_k: stored items, m: # stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored at pos i.

Space Usage

$$\mathbb{E}[\mathbf{s}_i^2] = \sum_{j,k \in [m]} \mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right]$$

$$= m \cdot \frac{1}{n} + 2 \cdot \binom{m}{2} \cdot \frac{1}{n^2}$$

$$= \frac{m}{n} + \frac{m(m-1)}{n^2} \leq 2 \text{ if we set } n = m.$$

- For $j = k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n}$.
- For $j \neq k$, $\mathbb{E}\left[\mathbb{I}_{\mathbf{h}(x_j)=i} \cdot \mathbb{I}_{\mathbf{h}(x_k)=i}\right] = \frac{1}{n^2}$.

Total Expected Space Usage: (if we set $n = m$)

$$\mathbb{E}[\mathbf{S}] = n + \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2] \leq n + n \cdot 2 = 3n = 3m.$$

Near optimal space with $O(1)$ query time!

x_j, x_k: stored items, m: # stored items, n: hash table size, $\mathbf{h}$: random hash function, $\mathbf{S}$: space usage of two level hashing, $\mathbf{s}_i$: # items stored at pos i.

Efficiently Computable Hash Function

So Far: we have assumed a **fully random hash function** $\mathbf{h}(x)$ with $\Pr[\mathbf{h}(x) = i] = \frac{1}{n}$ for $i \in 1, \ldots, n$ and $\mathbf{h}(x), \mathbf{h}(y)$ independent for $x \neq y$.

Efficiently Computable Hash Function

So Far: we have assumed a **fully random hash function** $\mathbf{h}(x)$ with $\Pr[\mathbf{h}(x) = i] = \frac{1}{n}$ for $i \in 1, \ldots, n$ and $\mathbf{h}(x), \mathbf{h}(y)$ independent for $x \neq y$.

- To compute a random hash function we have to store a table of x values and their hash values. Would take at least $O(m)$ space and $O(m)$ query time to look up $\mathbf{h}(x)$ if we hash m values. Making our whole quest for $O(1)$ query time pointless!

x	h(x)
x_1	45
x_2	1004
x_3	10
⋮	⋮
x_m	12

Efficiently Computable Hash Functions

What properties did we use of the randomly chosen hash function?

Efficiently Computable Hash Functions

What properties did we use of the randomly chosen hash function?

Pairwise Independent Hash Function. A random hash function from $\mathbf{h} : U \to [n]$ is pairwise independent if for all $i, j \in [n]$:

$$\Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = j] = \frac{1}{n^2}.$$

$$\Pr(h(x) = i \cap h(y) = j \cap h(z) = k) = \frac{1}{n^3}$$

"spec for a hash function"

Efficiently Computable Hash Functions

What properties did we use of the randomly chosen hash function?

Pairwise Independent Hash Function. A random hash function from $\mathbf{h} : U \to [n]$ is pairwise independent if for all $i, j \in [n]$:

$$\Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = j] = \frac{1}{n^2}.$$

Exercise 1: Check the two-level hashing proof to confirm that this property is all that was needed.

Efficiently Computable Hash Functions

What properties did we use of the randomly chosen hash function?

Pairwise Independent Hash Function. A random hash function from $\mathbf{h} : U \to [n]$ is pairwise independent if for all $i, j \in [n]$:

$$\Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = j] = \frac{1}{n^2}.$$

Exercise 1: Check the two-level hashing proof to confirm that this property is all that was needed.

When $\mathbf{h}(x)$ and $\mathbf{h}(y)$ are chosen independently at random from $[n]$, $\Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = j] = \frac{1}{n^2}$ (so a fully random hash function is pairwise independent).

Efficiently Computable Hash Functions

What properties did we use of the randomly chosen hash function?

Pairwise Independent Hash Function. A random hash function from $\mathbf{h} : U \rightarrow [n]$ is pairwise independent if for all $i, j \in [n]$:

$$\Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = j] = \frac{1}{n^2}.$$

Exercise 1: Check the two-level hashing proof to confirm that this property is all that was needed.

When $\mathbf{h}(x)$ and $\mathbf{h}(y)$ are chosen independently at random from $[n]$, $\Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = j] = \frac{1}{n^2}$ (so a fully random hash function is pairwise independent).

Efficient Implementation: Let p be a prime with $p \geq |U|$. Choose random $\mathbf{a}, \mathbf{b} \in [p]$ with $\mathbf{a} \neq 0$. Represent x as an integer and let

$$\mathbf{h}(x) = (\mathbf{a}x + \mathbf{b} \mod p) \mod n.$$

Universal Hashing

Another common requirement for a hash function:

2-Universal Hash Function (low collision probability). A random hash function from $\mathbf{h} : U \to [n]$ is two universal if:

$$\Pr[\mathbf{h}(x) = \mathbf{h}(y)] \leq \frac{1}{n}.$$

Universal Hashing

Another common requirement for a hash function:

2-Universal Hash Function (low collision probability). A random hash function from $\mathbf{h} : U \to [n]$ is two universal if:

$$\Pr[\mathbf{h}(x) = \mathbf{h}(y)] \leq \frac{1}{n}.$$

Think-Pair-Shair: Which is a more stringent requirement? 2-universal or pairwise independent?

Universal Hashing

Another common requirement for a hash function:

2-Universal Hash Function (low collision probability). A random hash function from $\mathbf{h} : U \to [n]$ is two universal if:

$$\Pr[\mathbf{h}(x) = \mathbf{h}(y)] \leq \frac{1}{n}.$$

Think-Pair-Shair: Which is a more stringent requirement? 2-universal or pairwise independent?

Pairwise Independent Hash Function. A random hash function from $\mathbf{h} : U \to [n]$ is pairwise independent if for all $i, j \in [n]$:

$$\Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = j] = \frac{1}{n^2}.$$

Another common requirement for a hash function:

2-Universal Hash Function (low collision probability). A random hash function from $\mathbf{h} : U \rightarrow [n]$ is two universal if:

$$\Pr[\mathbf{h}(x) = \mathbf{h}(y)] \leq \frac{1}{n}.$$

Think-Pair-Shair: Which is a more stringent requirement? 2-universal or pairwise independent?

$$\Pr[\mathbf{h}(x) = \mathbf{h}(y)] = \sum_{i=1}^{n} \Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = i] = n \cdot \frac{1}{n^2} = \frac{1}{n}.$$

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

Universal Hashing

Another common requirement for a hash function:

2-Universal Hash Function (low collision probability). A random hash function from $\mathbf{h} : U \to [n]$ is two universal if:

$$\Pr[\mathbf{h}(x) = \mathbf{h}(y)] \leq \frac{1}{n}.$$

Think-Pair-Shair: Which is a more stringent requirement? 2-universal or pairwise independent?

$$\Pr[\mathbf{h}(x) = \mathbf{h}(y)] = \sum_{i=1}^{n} \Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = i] = n \cdot \frac{1}{n^2} = \frac{1}{n}.$$

Remember: A fully random hash function is both 2-universal and pairwise independent. But it is not efficiently implementable.

Universal Hashing

Another common requirement for a hash function:

2-Universal Hash Function (low collision probability). A random hash function from $\mathbf{h} : U \to [n]$ is two universal if:

$$\Pr[\mathbf{h}(x) = \mathbf{h}(y)] \leq \frac{1}{n}.$$

Think-Pair-Shair: Which is a more stringent requirement? 2-universal or pairwise independent?

$$\Pr[\mathbf{h}(x) = \mathbf{h}(y)] = \sum_{i=1}^{n} \Pr[\mathbf{h}(x) = i \cap \mathbf{h}(y) = i] = n \cdot \frac{1}{n^2} = \frac{1}{n}.$$

Remember: A fully random hash function is both 2-universal and pairwise independent. But it is not efficiently implementable.

Exercise 2: Rework the two-level hashing proof to show that 2-universality is in fact all that is needed.

Questions on Hash Tables?