# COMPSCI 514: ALGORITHMS FOR DATA SCIENCE

Andrew McGregor

Lecture 3

**Today:**

- Continue random hash functions and hash tables.
- See an application of random hashing to load balancing in distributed systems.
- Through this application learn about:
  - Chebyshev's inequality, which strengthens Markov's inequality.
  - The union bound, for understanding the probabilities of correlated random events.

# HASH TABLES

Want to store a set of items from some finite but massive universe of items (e.g., images of a certain size, text documents, 128-bit IP addresses).

Want to store a set of items from some finite but massive universe of items (e.g., images of a certain size, text documents, 128-bit IP addresses).

**Goal:** support *query*$(x)$ to check if $x$ is in the set in $O(1)$ time.

Want to store a set of items from some finite but massive universe of items (e.g., images of a certain size, text documents, 128-bit IP addresses).

**Goal:** support $query(x)$ to check if $x$ is in the set in $O(1)$ time.

**Classic Solution:**

Want to store a set of items from some finite but massive universe of items (e.g., images of a certain size, text documents, 128-bit IP addresses).

**Goal:** support $query(x)$ to check if $x$ is in the set in $O(1)$ time.

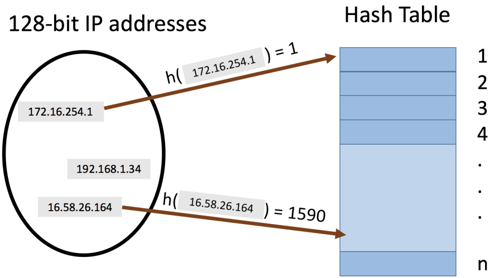
**Classic Solution:** Hash tables

Want to store a set of items from some finite but massive universe of items (e.g., images of a certain size, text documents, 128-bit IP addresses).

**Goal:** support $query(x)$ to check if $x$ is in the set in $O(1)$ time.

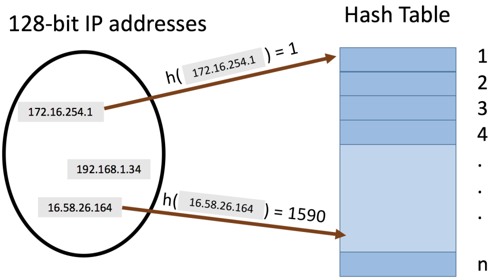
**Classic Solution:** Hash tables

- *Static hashing* since we won't worry about insertion and deletion today.



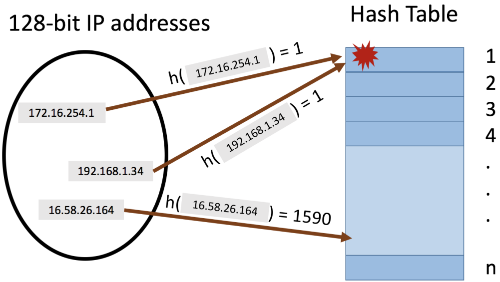

- **hash function** $h : U \to [n]$ maps elements in universe $U = \{x_1, x_2, \ldots\}$ to indices of an array. Assume **h** is fully independent, i.e.,
  a) $\Pr(\mathbf{h}(x_i) = j) = \frac{1}{n}$ for all $x_i \in U$ and $j \in [n]$ and
  b) all $\mathbf{h}(x_1), \mathbf{h}(x_2), \mathbf{h}(x_3) \ldots$ are all independent.

  It is *very expensive* to represent and compute fully independent random functions. Later, we will see how efficient hash functions are sufficient.




- **hash function** $h : U \to [n]$ maps elements in universe $U = \{x_1, x_2, \ldots\}$ to indices of an array. Assume **h** is fully independent, i.e.,
  a) $\Pr(\mathbf{h}(x_i) = j) = \frac{1}{n}$ for all $x_i \in U$ and $j \in [n]$ and
  b) all $\mathbf{h}(x_1), \mathbf{h}(x_2), \mathbf{h}(x_3) \ldots$ are all independent.

  It is *very expensive* to represent and compute fully independent random functions. Later, we will see how efficient hash functions are sufficient.
- **Collisions:** when we insert $m$ items into the hash table we may have to store multiple items in the same location (typically as a linked list).




- **hash function** $h : U \to [n]$ maps elements in universe $U = \{x_1, x_2, \ldots\}$ to indices of an array. Assume $\mathbf{h}$ is fully independent, i.e.,
  a) $\Pr(\mathbf{h}(x_i) = j) = \frac{1}{n}$ for all $x_i \in U$ and $j \in [n]$ and
  b) all $\mathbf{h}(x_1), \mathbf{h}(x_2), \mathbf{h}(x_3) \ldots$ are all independent.

  It is *very expensive* to represent and compute fully independent random functions. Later, we will see how efficient hash functions are sufficient.
- **Collisions:** when we insert $m$ items into the hash table we may have to store multiple items in the same location (typically as a linked list).

Let $\mathbf{C}_{i,j} = 1$ if items $i$ and $j$ collide ($\mathbf{h}(x_i) = \mathbf{h}(x_j)$), and 0 otherwise. The number of pairwise collisions is:

$$\mathbf{C} = \sum_{i,j \in [m], i<j} \mathbf{C}_{i,j}.$$

$x_i, x_j$: pair of stored items, $m$: total number of stored items, $n$: hash table size, $\mathbf{C}$: total pairwise collisions in table, $\mathbf{h}$: random hash function.

# LINEARITY OF EXPECTATION

Let $\mathbf{C}_{i,j} = 1$ if items $i$ and $j$ collide ($\mathbf{h}(x_i) = \mathbf{h}(x_j)$), and 0 otherwise. The number of pairwise collisions is:

$$\mathbb{E}[\mathbf{C}] = \sum_{i,j \in [m], i<j} \mathbb{E}[\mathbf{C}_{i,j}]. \qquad \text{(linearity of expectation)}$$

$x_i, x_j$: pair of stored items, $m$: total number of stored items, $n$: hash table size, $\mathbf{C}$: total pairwise collisions in table, $\mathbf{h}$: random hash function.

# LINEARITY OF EXPECTATION

Let $\mathbf{C}_{i,j} = 1$ if items $i$ and $j$ collide ($\mathbf{h}(x_i) = \mathbf{h}(x_j)$), and 0 otherwise. The number of pairwise collisions is:

$$\mathbb{E}[\mathbf{C}] = \sum_{i,j \in [m], i<j} \mathbb{E}[\mathbf{C}_{i,j}]. \qquad \text{(linearity of expectation)}$$

For any pair $i, j$, $i < j$:

$$\mathbb{E}[\mathbf{C}_{i,j}] = \Pr[\mathbf{C}_{i,j} = 1] = \Pr[\mathbf{h}(x_i) = \mathbf{h}(x_j)]$$

$x_i, x_j$: pair of stored items, $m$: total number of stored items, $n$: hash table size, $\mathbf{C}$: total pairwise collisions in table, $\mathbf{h}$: random hash function.

Let $\mathbf{C}_{i,j} = 1$ if items $i$ and $j$ collide ($\mathbf{h}(x_i) = \mathbf{h}(x_j)$), and 0 otherwise. The number of pairwise collisions is:

$$\mathbb{E}[\mathbf{C}] = \sum_{i,j \in [m], i<j} \mathbb{E}[\mathbf{C}_{i,j}]. \qquad \text{(linearity of expectation)}$$

For any pair $i, j$, $i < j$:
$\mathbb{E}[\mathbf{C}_{i,j}] = \Pr[\mathbf{C}_{i,j} = 1] = \Pr[\mathbf{h}(x_i) = \mathbf{h}(x_j)] = \frac{1}{n}$.

$x_i, x_j$: pair of stored items, $m$: total number of stored items, $n$: hash table size, $\mathbf{C}$: total pairwise collisions in table, $\mathbf{h}$: random hash function.

Let $\mathbf{C}_{i,j} = 1$ if items $i$ and $j$ collide ($\mathbf{h}(x_i) = \mathbf{h}(x_j)$), and 0 otherwise. The number of pairwise collisions is:

$$\mathbb{E}[\mathbf{C}] = \sum_{i,j \in [m], i<j} \mathbb{E}[\mathbf{C}_{i,j}]. \qquad \text{(linearity of expectation)}$$

For any pair $i, j$, $i < j$:
$\mathbb{E}[\mathbf{C}_{i,j}] = \Pr[\mathbf{C}_{i,j} = 1] = \Pr[\mathbf{h}(x_i) = \mathbf{h}(x_j)] = \frac{1}{n}$.

$$\mathbb{E}[\mathbf{C}] = \sum_{i,j \in [m], i<j} \frac{1}{n} = \frac{\binom{m}{2}}{n} = \frac{m(m-1)}{2n}.$$

$x_i, x_j$: pair of stored items, $m$: total number of stored items, $n$: hash table size, $\mathbf{C}$: total pairwise collisions in table, $\mathbf{h}$: random hash function.

Let $\mathbf{C}_{i,j} = 1$ if items $i$ and $j$ collide ($\mathbf{h}(x_i) = \mathbf{h}(x_j)$), and 0 otherwise. The number of pairwise collisions is:

$$\mathbb{E}[\mathbf{C}] = \sum_{i,j \in [m], i<j} \mathbb{E}[\mathbf{C}_{i,j}]. \qquad \text{(linearity of expectation)}$$

For any pair $i, j$, $i < j$:
$\mathbb{E}[\mathbf{C}_{i,j}] = \Pr[\mathbf{C}_{i,j} = 1] = \Pr[\mathbf{h}(x_i) = \mathbf{h}(x_j)] = \frac{1}{n}$.

$$\mathbb{E}[\mathbf{C}] = \sum_{i,j \in [m], i<j} \frac{1}{n} = \frac{\binom{m}{2}}{n} = \frac{m(m-1)}{2n}.$$

Identical to the CAPTCHA analysis!

$x_i, x_j$: pair of stored items, $m$: total number of stored items, $n$: hash table size, $\mathbf{C}$: total pairwise collisions in table, $\mathbf{h}$: random hash function.

# COLLISION FREE HASHING

$$\mathbb{E}[\mathbf{C}] = \frac{m(m-1)}{2n}.$$

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

# COLLISION FREE HASHING

$$\mathbb{E}[\mathbf{C}] = \frac{m(m-1)}{2n}.$$

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

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

# COLLISION FREE HASHING

$$\mathbb{E}[\mathbf{C}] = \frac{m(m-1)}{2n}.$$

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

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

$$\mathbb{E}[\mathbf{C}] = \frac{m(m-1)}{2n}.$$

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

**Apply Markov's Inequality:**

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

$$\mathbb{E}[\mathbf{C}] = \frac{m(m-1)}{2n}.$$

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

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

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

$$\mathbb{E}[\mathbf{C}] = \frac{m(m-1)}{2n}.$$

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

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

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

$$\mathbb{E}[\mathbf{C}] = \frac{m(m-1)}{2n}.$$

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

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

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

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

# COLLISION FREE HASHING

$$\mathbb{E}[\mathbf{C}] = \frac{m(m-1)}{2n}.$$

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

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

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

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

# COLLISION FREE HASHING

$$\mathbb{E}[\mathbf{C}] = \frac{m(m-1)}{2n}.$$

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

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

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

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

$$\mathbb{E}[\mathbf{C}] = \frac{m(m-1)}{2n}.$$

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

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

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

Pretty good but we are using $O(m^2)$ space to store $m$ items.

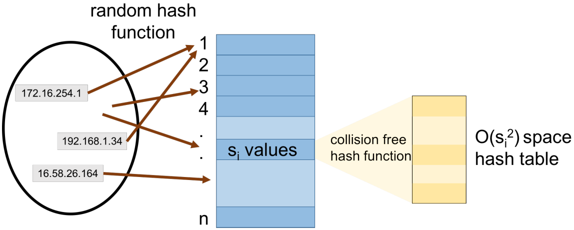
$m$: total number of stored items, $n$: hash table size, $\mathbf{C}$: total pairwise collisions in table.

# TWO LEVEL HASHING

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

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

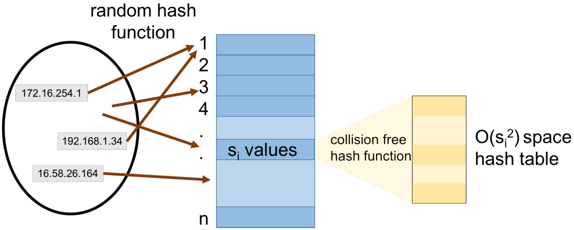
**Two-Level Hashing:**



# TWO LEVEL HASHING

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

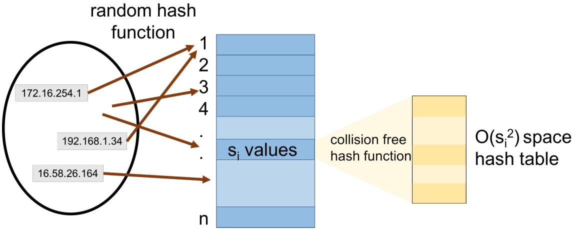
**Two-Level Hashing:**




- For each bucket with $s_i$ values, pick a collision free hash function mapping $[s_i] \rightarrow [4s_i^2]$.

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 $[s_i] \to [4s_i^2]$.
- **Previously:** Showed that 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: $\mathbf{S} = n + 4\sum_{i=1}^{n} \mathbf{s}_i^2$

$x_j, x_k$: 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: $\mathbb{E}[\mathbf{S}] = n + 4\sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$x_j, x_k$: 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: $\mathbb{E}[\mathbf{S}] = n + 4\sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

$x_j, x_k$: 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: $\mathbb{E}[\mathbf{S}] = n + 4\sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

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

$x_j, x_k$: 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: $\mathbb{E}[\mathbf{S}] = n + 4\sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

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

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

**Collisions again!**

$x_j, x_k$: 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: $\mathbb{E}[\mathbf{S}] = n + 4\sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

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

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

$x_j, x_k$: 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: $\mathbb{E}[\mathbf{S}] = n + 4\sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

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

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

- 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: $\mathbb{E}[\mathbf{S}] = n + 4\sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

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

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

- 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: $\mathbb{E}[\mathbf{S}] = n + 4\sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

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

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

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

$x_j, x_k$: 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: $\mathbb{E}[\mathbf{S}] = n + 4\sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

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

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

- 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: $\mathbb{E}[\mathbf{S}] = n + 4\sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

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

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

- 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: $\mathbb{E}[\mathbf{S}] = n + 4\sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

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

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

- 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: $\mathbb{E}[\mathbf{S}] = n + 4\sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

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

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

- 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: $\mathbb{E}[\mathbf{S}] = n + 4\sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$

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

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

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

$x_j, x_k$: 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

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

- 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

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

- 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

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

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

- 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

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

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

- 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

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

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

$x_j, x_k$: 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 $n = m$)

$$\mathbb{E}[\mathbf{S}] = n + 4 \sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2]$$

$x_j, x_k$: 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 $n = m$)

$$\mathbb{E}[\mathbf{S}] = n + 4\sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2] \leq n + 4n \cdot 2 = 9n = 9m.$$

$x_j, x_k$: 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 $n = m$)

$$\mathbb{E}[\mathbf{S}] = n + 4\sum_{i=1}^{n} \mathbb{E}[\mathbf{s}_i^2] \leq n + 4n \cdot 2 = 9n = 9m.$$

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 FUNCTIONS

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

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

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

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

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

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

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

**Exercise:** Rework the two level hashing proof to show that this property is really all that is needed.

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

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

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

**Exercise:** Rework the two level hashing proof to show that this property is really all that is needed.

When $\mathbf{h}(x)$ and $\mathbf{h}(y)$ are chosen independently at random from $[n]$, $\Pr[\mathbf{h}(x) = \mathbf{h}(y)] = \frac{1}{n}$ (so a fully random hash function is 2-universal)

# EFFICIENTLY COMPUTABLE HASH FUNCTIONS

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

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

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

**Exercise:** Rework the two level hashing proof to show that this property is really all that is needed.

When $\mathbf{h}(x)$ and $\mathbf{h}(y)$ are chosen independently at random from $[n]$, $\Pr[\mathbf{h}(x) = \mathbf{h}(y)] = \frac{1}{n}$ (so a fully random hash function is 2-universal)

**Efficient Alternative:** Let $p$ be a prime with $p \geq |U|$. Choose random $\mathbf{a}, \mathbf{b} \in [p]$ with $\mathbf{a} \neq 0$. Let:

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

# PAIRWISE INDEPENDENCE

Another common requirement for a hash function:

# PAIRWISE INDEPENDENCE

Another common requirement for a hash function:

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

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

Another common requirement for a hash function:

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

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

Which is a more stringent requirement? 2-universal or pairwise independent?

Another common requirement for a hash function:

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

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

Which is a more stringent requirement? 2-universal or pairwise independent?

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

Another common requirement for a hash function:

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

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

Which is a more stringent requirement? 2-universal or pairwise independent?

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

A closely related $(\mathbf{a}x + \mathbf{b}) \mod p$ construction gives pairwise independence on top of 2-universality.

Another common requirement for a hash function:

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

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

Which is a more stringent requirement? 2-universal or pairwise independent?

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

A closely related $(\mathbf{a}x + \mathbf{b}) \mod p$ construction gives pairwise independence on top of 2-universality.

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

# NEXT STEP

1. We'll consider an application where our toolkit of linearity of expectation + Markov's inequality doesn't give much.

# NEXT STEP

1. We'll consider an application where our toolkit of linearity of expectation + Markov's inequality doesn't give much.

2. Then we'll show how a simple twist on Markov's can give a much stronger result.

## CHEBYSHEV'S INEQUALITY

With a very simple twist Markov's Inequality can be made much more powerful.

# CHEBYSHEV'S INEQUALITY

With a very simple twist Markov's Inequality can be made much more powerful.

For any random variable $\mathbf{X}$ and any value $t > 0$:

$$\Pr(|\mathbf{X}| \geq t) = \Pr(\mathbf{X}^2 \geq t^2).$$

With a very simple twist Markov's Inequality can be made much more powerful.

For any random variable $\mathbf{X}$ and any value $t > 0$:

$$\Pr(|\mathbf{X}| \geq t) = \Pr(\mathbf{X}^2 \geq t^2).$$

$\mathbf{X}^2$ is a nonnegative random variable. So can apply Markov's inequality:

# CHEBYSHEV'S INEQUALITY

With a very simple twist Markov's Inequality can be made much more powerful.

For any random variable $\mathbf{X}$ and any value $t > 0$:

$$\Pr(|\mathbf{X}| \geq t) = \Pr(\mathbf{X}^2 \geq t^2).$$

$\mathbf{X}^2$ is a nonnegative random variable. So can apply Markov's inequality:

$$\Pr(\mathbf{X}^2 \geq t^2) \leq \frac{\mathbb{E}[\mathbf{X}^2]}{t^2}.$$

# CHEBYSHEV'S INEQUALITY

With a very simple twist Markov's Inequality can be made much more powerful.

For any random variable $\mathbf{X}$ and any value $t > 0$:

$$\Pr(|\mathbf{X}| \geq t) = \Pr(\mathbf{X}^2 \geq t^2).$$

$\mathbf{X}^2$ is a nonnegative random variable. So can apply Markov's inequality:

$$\Pr(|\mathbf{X}| \geq t) = \Pr(\mathbf{X}^2 \geq t^2) \leq \frac{\mathbb{E}[\mathbf{X}^2]}{t^2}.$$

With a very simple twist Markov's Inequality can be made much more powerful.

For any random variable $\mathbf{X}$ and any value $t > 0$:

$$\Pr(|\mathbf{X}| \geq t) = \Pr(\mathbf{X}^2 \geq t^2).$$

$\mathbf{X}^2$ is a nonnegative random variable. So can apply Markov's inequality:

**Chebyshev's inequality:**

$$\Pr(|\mathbf{X}| \geq t) = \Pr(\mathbf{X}^2 \geq t^2) \leq \frac{\mathbb{E}[\mathbf{X}^2]}{t^2}.$$

With a very simple twist Markov's Inequality can be made much more powerful.

For any random variable $\mathbf{X}$ and any value $t > 0$:

$$\Pr(|\mathbf{X}| \geq t) = \Pr(\mathbf{X}^2 \geq t^2).$$

$\mathbf{X}^2$ is a nonnegative random variable. So can apply Markov's inequality:

**Chebyshev's inequality:**

$$\Pr(|\mathbf{X} - \mathbb{E}[\mathbf{X}]| \geq t) \leq \frac{\text{Var}[\mathbf{X}]}{t^2}.$$

(by plugging in the random variable $\mathbf{X} - \mathbb{E}[\mathbf{X}]$)