Note for Hashing

- ***Hash function*** => The function is chosen at random.

Given a typically large universe U of keys, and a positive integer m. A random hash function $h : U \to [m]$ is a randomly chosen function from $U \to [m]$.

My Understanding - 1

A random hash function is firstly a function that is selected from a set of hash functions randomly and it can map the keys from U to a range of numbers $0, \ldots, m-1$.

Equivalently, => For each x, the value at x is chosen at random.

It is a function h such that for each $x \in U$, $h(x) \in [m]$ is a random variable.

My Understanding - 2

A random hash function is let each key in U be the variable, and the result of hashing every time is random. For example, h_i means i^{th} hashing. $h_1(x) = a, h_2(x) = b$. a b are random variables.

Chinese Version

1. 随机哈希函数首先是一个从一个含有多个hash functions的集合里随机挑选出来的方程，使得$U \to [m]$.
2. 同样可以理解为一个哈希方程是让U里的每一个值作为哈希方程的自变量，每次对该自变量映射的结果都是随机的。

宏观上来看，每一个值在经过随机哈希后，输出的值是随机的。

Cryptographic hash functions such as MD5, SHA-1, and SHA-256 are not *random* hash functions.

- ***Three things we care***

1. Space (seed size) needed to represent h. => the size of S_h, cannot be too big
2. Time needed to calculate $h(x)$ given $x \in U$. => The inner part of a lot of algorithms is hashing.
3. Properties of the random variable.

- ***Hash function types***

Truly random

A hash function $h : U \to [m]$ is truly random if the variables $h(x)$ for $x \in U$ are ***independent*** and ***uniform***.

一个哈希方程想要 truly random，就得满足对于 $x \in U, h(x)$ 的结果每次都是 m 种可能，每次hashing的结果互不影响（独立），且概率都一样，都是$\frac{1}{m}$（统一）。

一共有$|U|$个输入，对于每一个输入，需要对应m个输出，此时一个输入需要$\log_2 m$字节在计算机里，则一共需要$|U| \log_2 m$个空间。

Universal

A random hash function $h : U \to [m]$ is ***universal*** if, for all $x \neq y \in U : \Pr[h(x) = h(y)] \leq \frac{1}{m}$. => Hash to the same value.

C-approximately universal

A random hash function $h : U \to [m]$ is ***c-approximately*** universal if, for all $x \neq y \in U : \Pr[h(x) = h(y)] \leq \frac{c}{m}$.

Strongly universal

A random hash function $h : U \to [m]$ is ***strongly universal*** (a.k.a. 2-independent) if,

1. Each key is hashed *uniformly* into $[m]$. => i.e., $\forall x \in U, q \in [m] : \Pr[h(x) = q] = \frac{1}{m}$.
2. Any two distinct keys hash *independently*.

Equivalently, if for all $x \neq y \in U$, and $q, r \in [m] : \Pr[h(x) = q \wedge h(y) = r] = \frac{1}{m^2}$.

C-approximately strongly universal

A random hash function $h : U \to [m]$ is c-approximately strongly universal if,

1. Each key is hashed c-approximately uniformly into $[m]$. => i.e., $\forall x \in U, q \in [m] : \Pr[h(x) = q] \leq \frac{c}{m}$
2. Any two distinct keys hash independently.

- ***Unordered sets / Hashing with chaining***

Maintain a set S of at most n keys from some unordered universe U, under three operations.

`INSERT(x, S)` Insert key x into S.

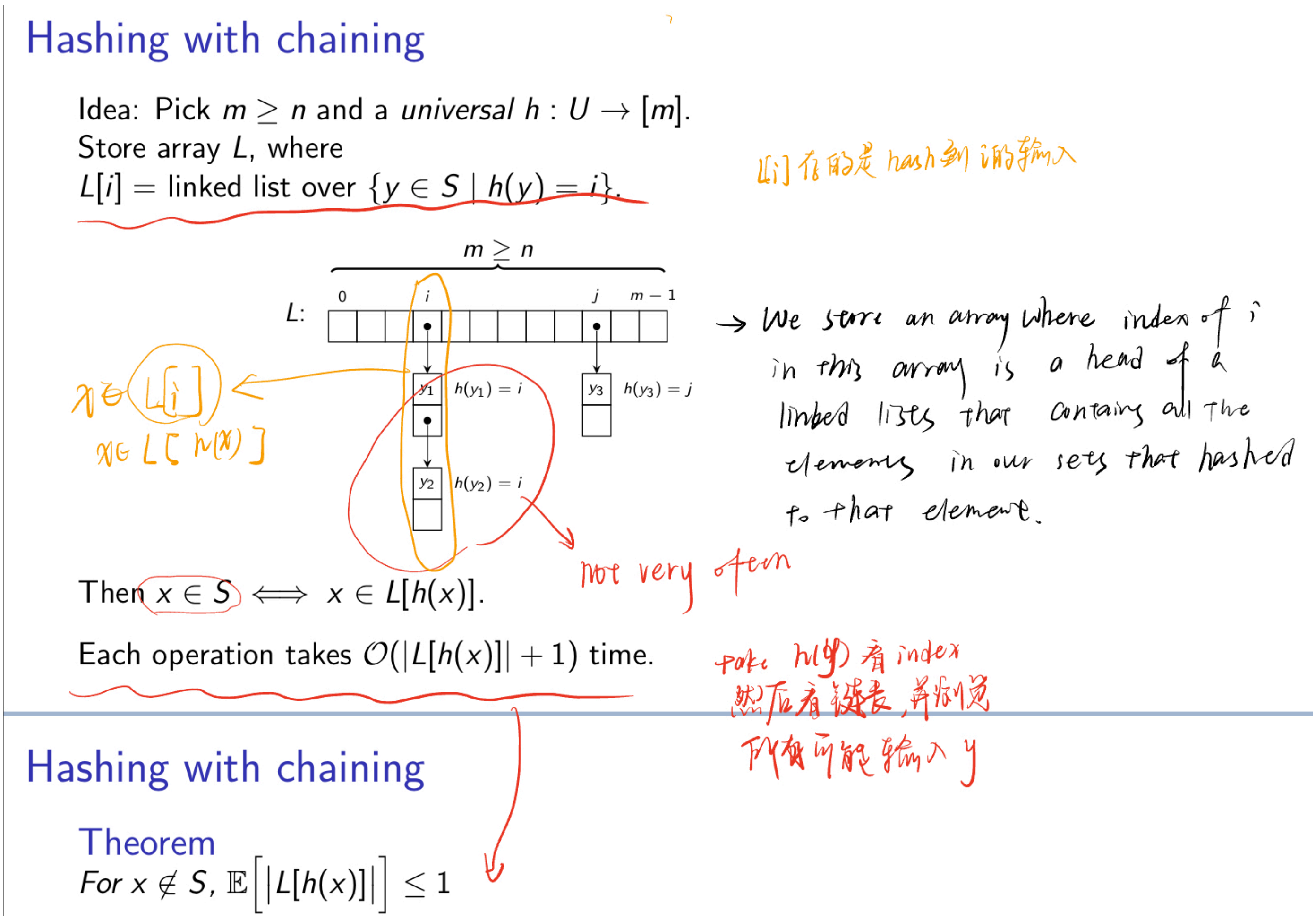
`DELETE(x, S)` Delete key x from S.

`MEMBER(x, S)` Return $x \in S$.

We could use *some form of balanced tree to store S, but they usually take $O(\log n)$ time operation*, and we want each operation to *run in expected constant time*.

The worst case for both `INSERT` and `DELETE` is rotating $\log_2 n$ times. And the worst case of `MEMBER` operation is finding the leaf node. That's the reason why these three operations are all run in $O(\log n)$, while hashing can help us run these three operations in constant time. => ***Hashing with Chaining***

- ***Hashing with Chaining*** => ***Universal Hashing***



Then, we store an array where the index of i in this array is a head of a linked list that contains all the elements in our sets that hashed to that element.

这三个方法所花费时间都和链表长度成正比。↓

Each operation take $O(|L[h(x)]| + 1)$ time. And we need to prove the former part is a constant time.

- ***Theorem - 1***

For $x \notin S, \mathbb{E}[|L[h(x)]|] \leq 1.$ => 找不存在在集合里的x所花费的时间。某种程度上算是最差情况，如果最差情况也被bound住，那一般情况肯定在bound里。

Proof.

$$
\begin{aligned}
\mathbb{E}[|L[h(x)]|] &= \mathbb{E}[|\{y \in S | h(y) = h(x)\}|] \Leftarrow By\, definition \\
&= \mathbb{E}\left[\sum_{y \in S} [h(y) = h(x)]\right] \Leftarrow Indicator\ variable \\
&= \sum_{y \in S} \mathbb{E}\left[[h(y) = h(x)]\right] \Leftarrow Linearity\, of\ expectation \\
&= \sum_{y \in S} \Pr[h(y) = h(x)] \Leftarrow Expectation\ of\ indicator\ variable \\
&\leq |S| \frac{1}{m} \Leftarrow Since\ x \neq y \Rightarrow Universal \\
&= \frac{n}{m} \leq 1
\end{aligned}
$$

This actually proves that hashing with chaining and expectation you use only constant time per operation.

- ***Signatures => Universal Hashing***

Application: Signatures

Problem: Assign a unique "signature" to each $x \in S \subseteq U$, $|S| = n$.

Solution: Use universal hash function $s : U \to [n^3]$. → The probability of getting a collision among your chosen signatures is very small.

Then by a "union bound"

$$\Pr[\exists \{x, y\} \subseteq S \mid s(x) = s(y)] \leq \sum_{\{x,y\} \subseteq S} \Pr[s(x) \overset{\frac{1}{n^3}}{=} s(y)] \leq \frac{\binom{n}{2}}{n^3} < \frac{1}{2n}$$

It's with high probability that we have no collisions.

"$\cup$" ≤ P(A) + P(B)

P(A∪B)

→ 共有 n chose 2 种pair，每种pair碰撞 $p \leq \frac{1}{n^3}$

→ universal

避免碰撞

① 如果想要更大的，n^4 …，小的概数

② 查是否有碰撞，有就再来一遍.

Thus with "high probability" we have no collisions.

universal hash function → cheaper to compute

- ***Multiply-mod-prime (2-approximately strongly universal)***

It is the most classic but not the fastest. However, it is good enough for some applications.

Multiply-mod-prime

Let $U = [u]$ and pick prime $p \geq u$. For any $a, b \in [p]$, and $m < u$, let $h^m_{a,b} : U \to [m]$ be

$$h^m_{a,b}(x) = ((ax + b) \bmod p) \bmod m$$

所以要定义为随机

Choose $a, b \in [p]$ independently and uniformly at random, and let $h(x) := h^m_{a,b}(x)$.
Then $h : U \to [m]$ is a 2-approximately strongly universal hash function.

相拟定义

必须要保证每次跑程序都得随机选的

Multiply-shift

→ love to work with

Let $U = [2^w]$ and $m = 2^\ell$. For any odd $a \in [2^w]$ define

$$h_a(x) := \left\lfloor \frac{(ax) \bmod 2^w}{2^{w-\ell}} \right\rfloor$$

向下取整

computer love to work with power of 2

Choose odd $a \in [2^w]$ uniformly at random, and let $h(x) := h_a(x)$.
Then $h : U \to [m]$ is a 2-approximately universal hash function.

足够 chaining 和 signature

$w = 64, 86, \ldots$

不要担心溢出. 因为他会自动选一个 mod

Exercise 3.4 asks you to show if there is some constant c so it is c-approximately strongly universal.

超级 cheap to compute

- ***Multiply-shift (2-approximately universal) => Universal Hashing***

Extremely cheaper to compute.

- ***Strong Multiply-shift => Strongly Universal Hashing***

It is a strongly universal hash function.

- ***Coordinated sampling => Strongly Universal Hashing***

Application: Coordinated sampling

Suppose we have a bunch of *agents* that each observe some set of events from some universe U. Let $A_i \subseteq U$ denote the set of events seen by agent i, and suppose $|A_i|$ is large so only a small sample $S_i \subseteq A_i$ is actually stored.

一堆代理人观察同一个U的一些事件
每个代理人会看到一堆事件集 A_i，但$|A_i|$太大，
所以只会取 A_i 里的样本 S_i

If each agent independently just samples a random subset of the seen events, there is very little chance that two agents that see an event make the same decision.
$\implies$ The samples are incomparable.

如果每个代理人在采样时是随机独立的，
那么两个代理人看到同一个事件并决定存在
S_i 里的概率是 $\frac{1}{n}$.

Coordinated sampling means that all agents that see an event make the same decision about whether to store it.
$\implies$ Samples can be combined, i.e.

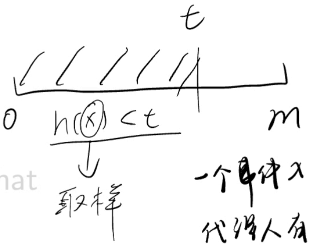
- $S_i \cup S_j$ is a sample of $A_i \cup A_j$
- $S_i \cap S_j$ is a sample of $A_i \cap A_j$

协调采样

给所有代理人设一个标准（一件事），
如果满足标准，都存在 S_i 里，不满足
都不存.

Let $h : U \to [m]$ be a strongly universal hash function, and let $t \in \{0, \ldots, m\}$. Send h and t to all the agents.

Each agent samples $x \in U$ iff $\boxed{h(x) < t}$



Thus if an agent sees the set $A \subseteq U$, the set $S_{h,t}(A) := \{x \in A \mid h(x) < t\}$ is sampled. Note that

- $S_{h,t}(A_i) \cup S_{h,t}(A_j) = S_{h,t}(A_i \cup A_j)$
- $S_{h,t}(A_i) \cap S_{h,t}(A_j) = S_{h,t}(A_i \cap A_j)$

一个事件 x 如果被多个代理人看到，
代理人有相同的 h 和 t. 如果 $h(x)<t$
则所有人都存在 S_i 里作为样本.

Application: Coordinated sampling

Let $h : U \to [m]$ be a strongly universal hash function, and let $t \in \{0, \ldots, m\}$. Send h and t to all the agents.

Each agent samples $x \in U$ iff $h(x) < t$.

Thus if an agent sees the set $A \subseteq U$, the set $S_{h,t}(A) := \{x \in A \mid h(x) < t\}$ is sampled. Note that

- $S_{h,t}(A_i) \cup S_{h,t}(A_j) = S_{h,t}(A_i \cup A_j)$
- $S_{h,t}(A_i) \cap S_{h,t}(A_j) = S_{h,t}(A_i \cap A_j)$

$\frac{t}{m}$

$[h(x) = t]$　小球命中 $<t$ 的概率为 $\frac{t}{m}$

Each $x \in A$ is sampled with probability $\Pr[h(x) < t] = \frac{t}{m}$.
Why? Strong universality $\implies$ $h(x)$ uniform in $[m]$

E[样本集的大小] = $|A| \cdot \frac{t}{m}$

均值

For any $A \subseteq U$, $\mathbb{E}[|S_{h,t}(A)|] = |A| \cdot \frac{t}{m}$.

$-|S|$

Thus we have an unbiased estimate $|A| \approx \frac{m}{t} \cdot |S_{h,t}(A)|$.

How good is this estimate?

- ***Lemma***

Lemma

Let $X = \sum_{a \in A} X_a$ where the X_a are pairwise independent 0–1 variables. Let $\mu = \mathbb{E}[X]$. Then $\mathrm{Var}[X] \leq \mu$, and for any $q > 0$,

$$\Pr[|X - \mu| \geq q\sqrt{\mu}] \leq \tfrac{1}{q^2}$$

Proof (not curriculum).

For $a \in A$ let $p_a = \Pr[X_a = 1]$. Then $p_a = \mathbb{E}[X_a]$ and

$$\begin{aligned}
\mathrm{Var}[X_a] &= \mathbb{E}[(X_a - p_a)^2] = (1 - p_a)(0 - p_a)^2 + p_a(1 - p_a)^2 \\
&= (p_a^2 + p_a(1 - p_a))(1 - p_a) = p_a(1 - p_a) \leq p_a \\
\mathrm{Var}[X] &= \mathrm{Var}\Big[\sum_{a \in A} X_a\Big] = \sum_{a \in A} \mathrm{Var}[X_a] \leq \sum_{a \in A} p_a = \mu
\end{aligned}$$

Finally, since $\sigma_X = \sqrt{\mathrm{Var}[X]} \leq \sqrt{\mu}$ we get:

$$\begin{aligned}
\Pr[|X - \mu| \geq q\sqrt{\mu}] &\leq \Pr[|X - \mu| \geq q\sigma_X] \\
&\leq \tfrac{1}{q^2} \qquad \text{(Chebyshev's ineq.)} \ \square
\end{aligned}$$

- ***How good is the unbiased estimate with Lemma?***

Application: Coordinated sampling

Let's apply this lemma to the estimate $|A| \approx \frac{m}{t}|S_{h,t}(A)|$ from our coordinated sampling.

Let $X = |S_{h,t}(A)|$ and for $a \in A$ let $X_a = [h(a) < t]$. Then $X = \sum_{a \in A} X_a$ and for any $a, b \in A$, X_a and X_b are independent. Also, let $\mu = \mathbb{E}[X] = \frac{t}{m}|A|$.

Then for any $q > 0$,

$$\begin{aligned}
&\Pr\left[\left|\tfrac{m}{t}|S_{h,t}(A)| - |A|\right| \geq q\sqrt{\tfrac{m}{t}|A|}\right] \\
&= \Pr\left[\left||S_{h,t}(A)| - \tfrac{t}{m}|A|\right| \geq q\sqrt{\tfrac{t}{m}|A|}\right] \\
&= \Pr[|X - \mu| \geq q\sqrt{\mu}] \leq \tfrac{1}{q^2}
\end{aligned}$$

We needed strong universality in two places for this to work. Where? h must be uniform to get unbiased estimate, and pairwise independent for the lemma.

Todays topic was hashing, and we have covered

- What is a random hash function, and what properties do we want.
- Two applications of universal hashing — unordered sets and signatures.
- Some concrete universal or strongly universal hash functions.
- An application of strongly universal hashing — coordinated sampling.
- Next time: An ordered set data structure that is not comparison based, and an application of hash tables.