Cryptographic Hash Functions

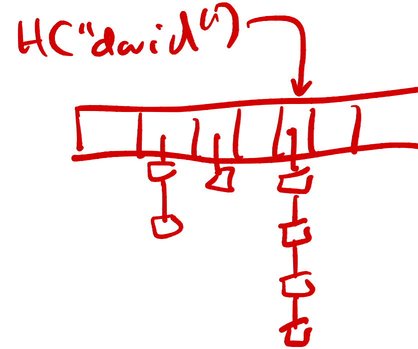
Lecture 13, CS 284, Autumn 2021

David Cash

Outline

1. Hash function basics, definitions
2. Hash function constructions
3. Small-space collision finding

Hash Functions in Computer Science



Data structures frequently use "hash functions"

$$H : \text{Labels} \to \text{Indexes}$$

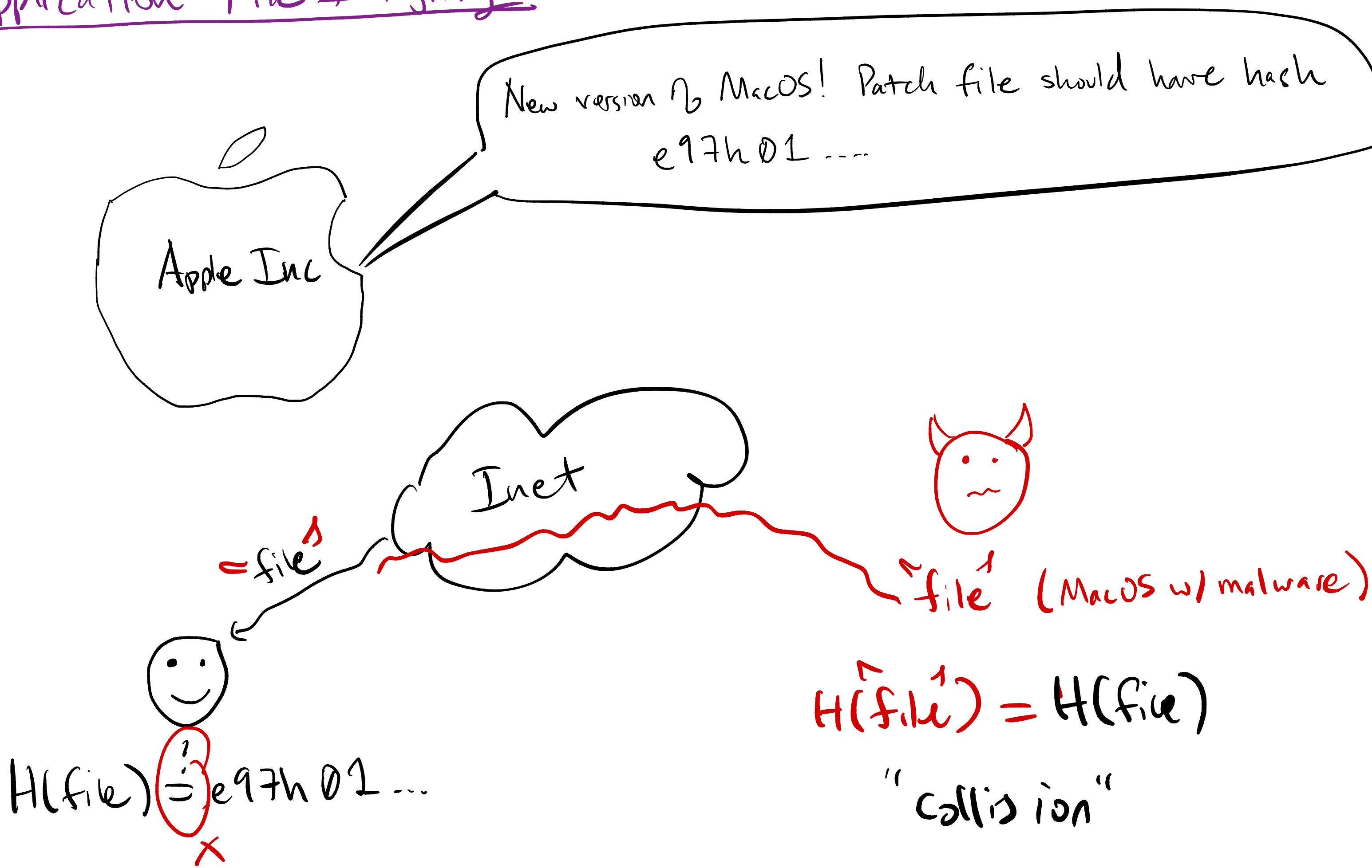
- $H(x)$ "looks random"
- H can take a "key" (like $H_k(x) = k \cdot x \bmod \ell$)
- Collisions happen and are handled ($H(x) = H(y)$)

Ex: Labels $= \{0,1\}^*$, Indexes $= \{1, \dots, 10{,}000\}$.

Cryptographic Hash Functions

- Super strong version of a hash function
- Syntax: $H : \mathcal{K} \times D \to \mathcal{R}$
  - $\mathcal{K}$ is set of keys (ex: $\{0,1\}^{128}$)
  - D is domain (ex: $\{0,1\}^*$)
  - $\mathcal{R}$ is range (ex: $\{0,1\}^{256}$)

Application: File Integrity



Application: Commitments

Before games: Publish $y = H(\text{predictions} \,\|\, \text{random})$

used to prevent finding pred early

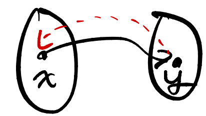
After games: Reveal predictions $\|$ random. Everyone checks y.

Security threats? → A might find input from y alone

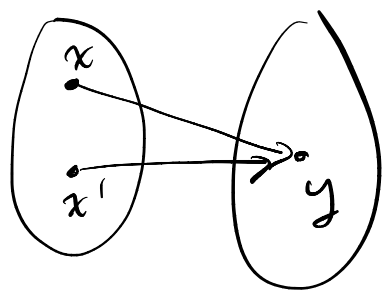
Collisions: $y = H(\text{pred} \,\|\, \text{rand}) = H(\text{pred}' \,\|\, \text{rand}')$

Hash Function Security Goals (informal)

"One-wayness": Given $y = H(x)$, find x



"Collision resistence": Find x, x' such that $x = x'$, $H(x) = H(x')$



... many possible properties.

Hash Security Goal: Collision Resistance

- Assume attacker knows key K! Key is not secret.

Goal: Design H so that it is very hard to find $x \neq x' \in D$ such that

$$H(K, x) = H(K, x').$$

Bad News For $H : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$, if $|\mathcal{D}| > |\mathcal{R}|$ ....

pigeonhole: $\exists\, x \neq x' : H(x) = H(x')$

Definition of Collision Resistance

Definition: Let $H : \mathcal{K} \times D \to \mathcal{R}$ be a hash function and $\mathcal{A}$ be an adversary. Define

$\mathrm{Expt}^{cr}_{H}(\mathcal{A})$

1. Pick $K \in \mathcal{K}$ at random
2. Give K to $\mathcal{A}$, which outputs x, x'
3. If $x, x' \in D$, $x \neq x'$, and $H(K, x) = H(K, x')$:

   Output 1

   Else Output $\emptyset$

and $\mathrm{Adv}^{cr}_{H}(\mathcal{A}) = \Pr[\mathrm{Expt}^{cr}_{H}(\mathcal{A}) = 1]$.

Obvious attack:

time $|\mathcal{R}|$

Example 1

$$H(k, x_1 \| \cdots \| x_t) = AES(k, x_1) \oplus \cdots \oplus AES(k, x_t)$$

$$H(k, x_1 \| x_2) = H(k, x_2 \| x_1)$$

$$x = x_1 \| x_2$$

$$x' = x_2 \| x_1$$

$$Adv_H^{cr}(A) = 1$$

Example 2

$$H(K, x_1 \| x_2) = K \oplus AES(x_1, x_2)$$

$x_1 = 0^{128}$

$x_2 = 0^{128}$

$z = AES(x_1, x_2)$

$x_1' = 1^{128}$

$x_2' = \underline{\quad}$

$\rightarrow AES(x_1', x_2') = z$

$\underline{x_2'} = AES^{-1}(x_1', z)$

$Adv = 1$

Recall: Collision Probabilities

Suppose we draw q independent, uniform samples from a set of size N. Let $C(N,q)$ be the probability that a value is repeated in our samples.

repeat!

$$x_1 \quad x_2 \quad x_3 \cdots x_i \cdots x_j \cdots x_q$$

$$\boxed{\begin{aligned} N &= 2^{256} \\ q &= 2^{128} \end{aligned}}$$

<u>Theorem</u> For $q \le \sqrt{2N}$,

$$0.3 \, \frac{q(q-1)}{N} \le C(N,q) \le 0.5 \, \frac{q(q-1)}{N}$$

$$\approx \quad q^2/N$$

Birthday Attack: Collision Finding Against any H

Let $H : \mathcal{K} \times D \to \mathcal{R}$

$\mathcal{A}(k)$ // Input: key k
Output: Collision x, x' ($x \neq x'$ but $H(k,x) = H(k,x')$)

Initialize hash table Y

For $x = 1, \ldots, q$:

$y \leftarrow H(k,x)$ // Treat number x as bit string input

If $Y[y] \neq \perp$:

$x' \leftarrow Y[y]$

output x, x'

Else: $Y[y] \leftarrow x$

Heuristically, $\mathrm{Adv}^{cr}_{H}(\mathcal{A}) = \mathrm{Col}(|\mathcal{R}|, q) \approx \frac{q^2}{N}$

Ex $|\mathcal{R}| = 2^{80}$

$q \approx 2^{40}$

$\mathrm{Adv} = \mathrm{Col}(2^{80}, 2^{40})$

≈ 1

$|\mathcal{R}| = 2^{256}$

$q \simeq 2^{128}$

n output bits $\Rightarrow 2^{n/2}$ security

Popular Hash Functions, Past and Present

SHA2 - SHA-256
SHA-512

Name	Date	Output Length	Security Status
MD5	1992	128	First collision in 2004. Now *very* broken.
SHA 1	1995	160	Collisions found in 2017
SHA 2 family	2001	224 or 256 or 358 or 512	Lookin' good
SHA 3 family	2015	same	Lookin' good

Outline

1. Hash function basics, definitions
2. Hash function constructions
3. Small-space collision finding

Hash Function Design Plan (SHA-256)

Two steps:

① Design a fixed-length function

$$h : \{0,1\}^{512} \times \{0,1\}^{256} \to \{0,1\}^{256}$$

② Chain h together to build

$$H : \{0,1\}^{*} \to \{0,1\}^{256}$$

Finally, reason that H has good C.R. as long as h does.

Step 1: Design compression function h

Want: $h: \{0,1\}^{512} \times \{0,1\}^{256} \to \{0,1\}^{256}$

- Build h from a block cipher! SHA-256 uses a custom block cipher

$$E: \{0,1\}^{512} \times \{0,1\}^{256} \to \{0,1\}^{256}$$

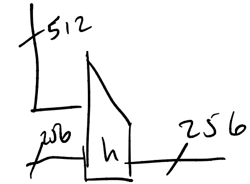
and defines

$$h(x,v) = E(x,v) \oplus v$$

Davies-Meyer

* Why not $h(x,v) = E(x,v)$? See problem set.

Step 2: Chain h together to build H



Assume we have $h: \{0,1\}^{512} \times \{0,1\}^{256} \to \{0,1\}^{256}$.

Now build $H: \{0,1\}^* \to \{0,1\}^{256}$.

$\underline{H(x)}$

$\ell \leftarrow \text{length}(x)$

$\bar{x} \leftarrow \text{pad}(x)$ // add zeros

Parse $x_1 \| \cdots \| x_t \leftarrow \bar{x}$ $(x_i \in \{0,1\}^{512})$

Set v_0 to a magic number 6a09e6...

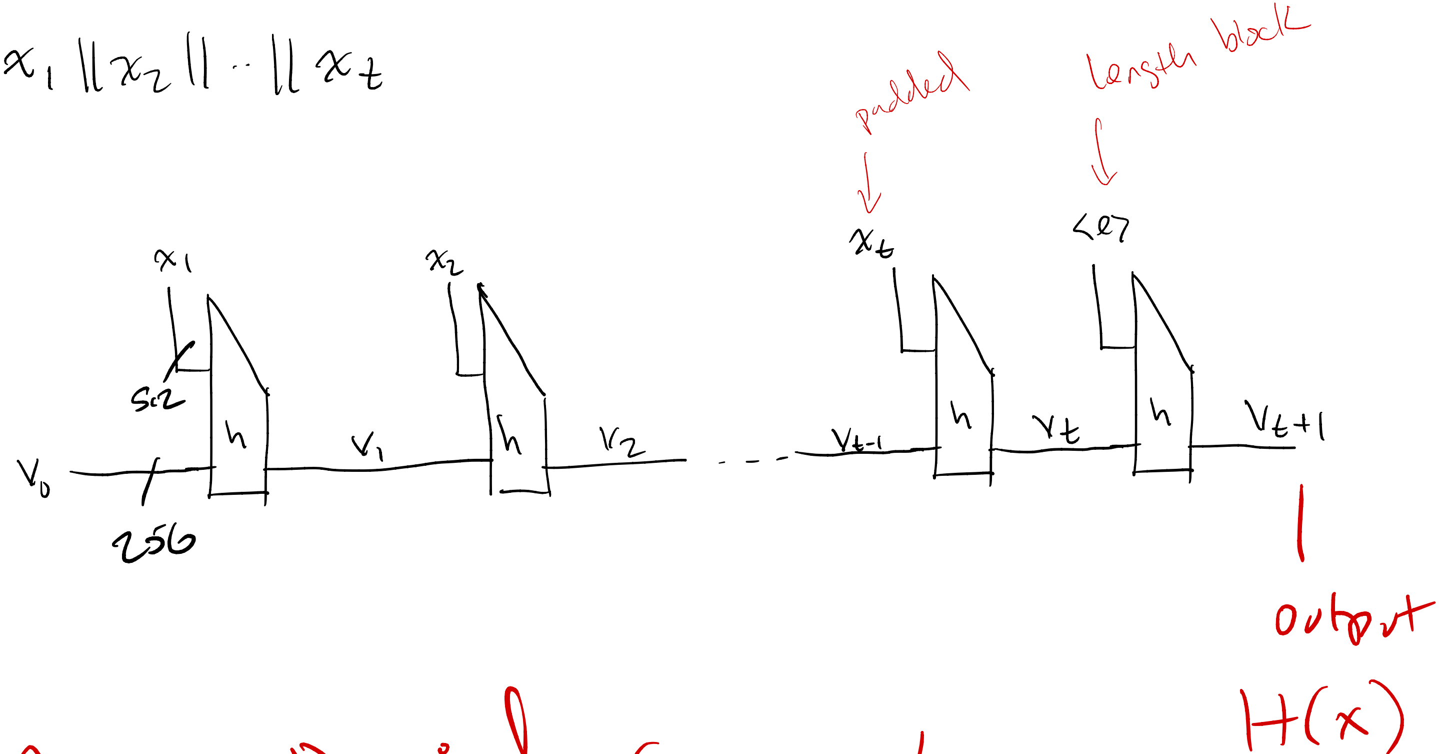
For $i=1 \ldots t$: $v_i \leftarrow h(x_i, v_{i-1})$

$v_{t+1} \leftarrow h(\langle \ell \rangle, v_t)$ 512-bit encoding of ℓ

output v_{t+1}

Step 2 in a Picture

$x_1 \| x_2 \| \cdots \| x_t$



Merkle-Damgård Chaining/Transform

Analysis / Intuition

Claim Given a collision x, x' for H, one can easily find a collision for h.

$\Rightarrow$ If attackers can't find collisions in h, then they can't find collisions in H either!

Analysis

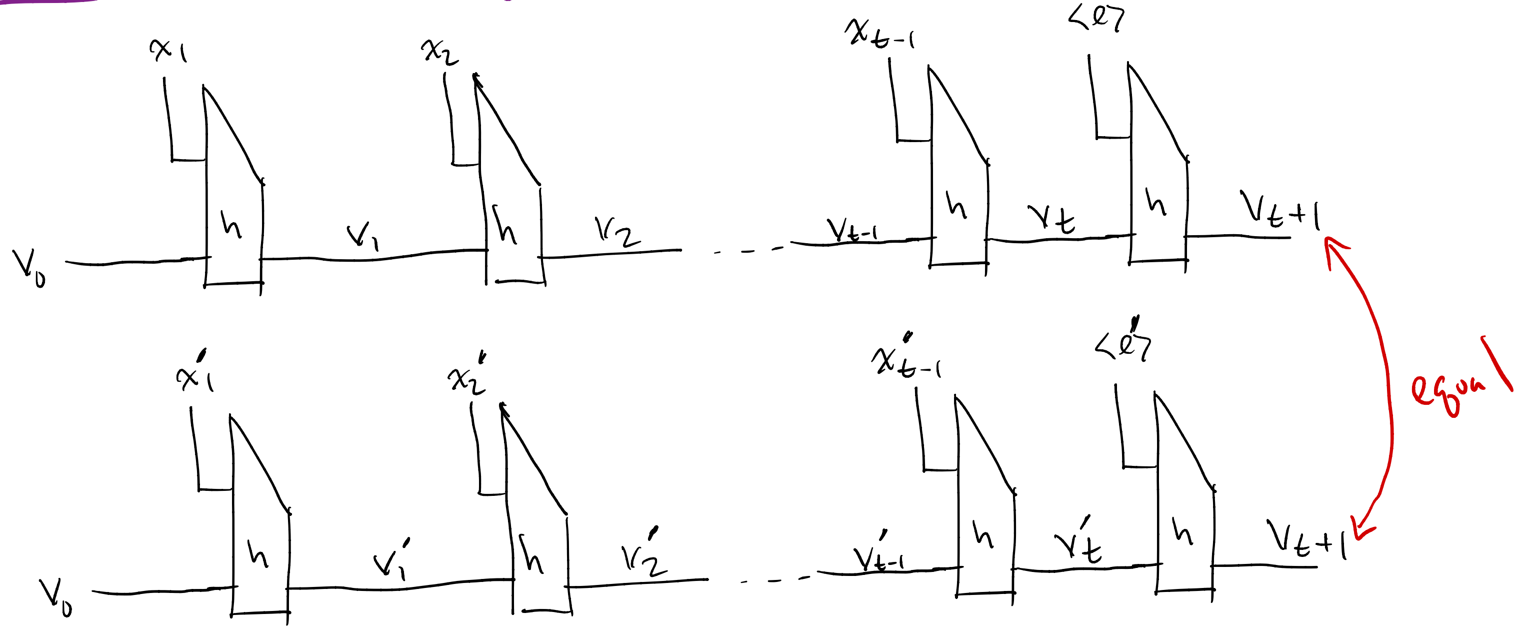
$h(\langle \ell \rangle, v_t) = h(\langle \ell' \rangle, v_t')$

$\neq$

Given a collision x, x' for H, need to find collision for h

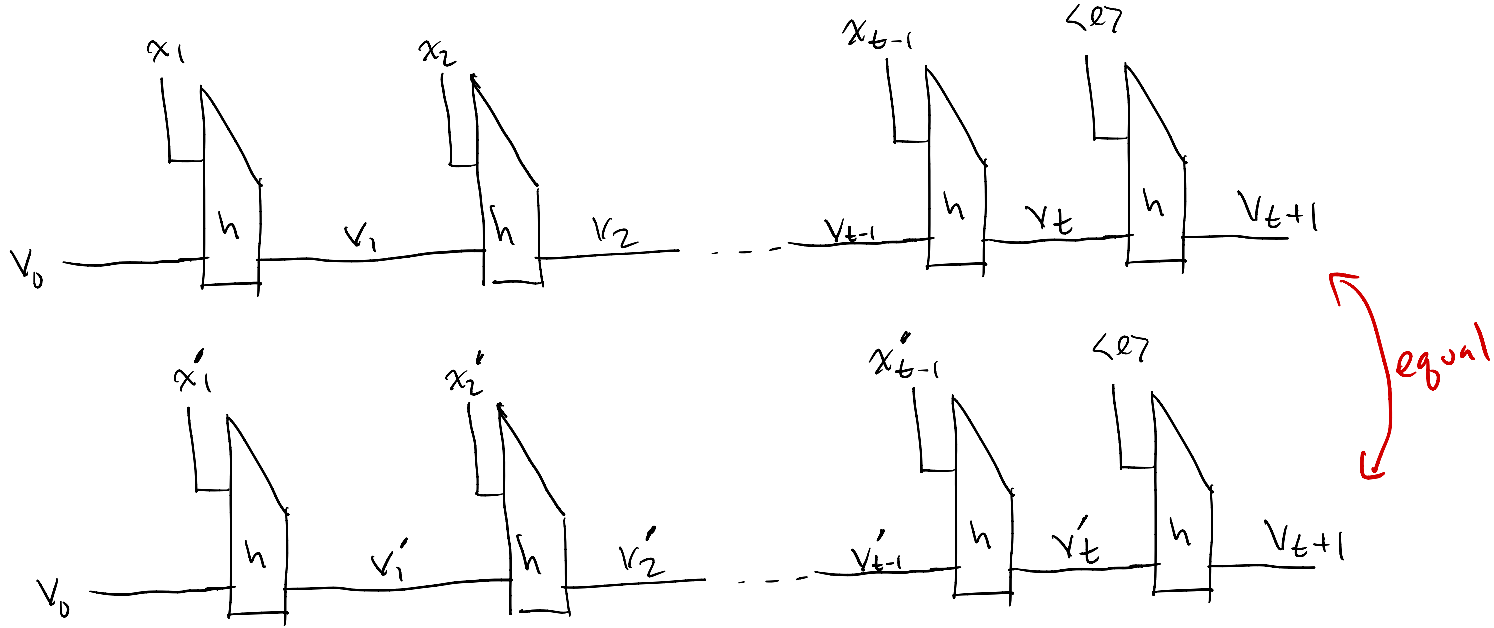
$\langle \ell \rangle \neq \langle \ell' \rangle$

Case 1 $\text{length}(x) \neq \text{length}(x')$



Analysis continued

Case 2 length(x) = length(x')



Outline

1. Hash function basics, definitions
2. Hash function constructions
3. Small-space collision finding

Birthday Attacks and Space

$\boxed{\cancel{\sqrt{R}} \quad \cancel{R^{1/3}} \quad \cancel{R^{1/100}} \quad \log R \quad \text{Const}}$

Let $H : \mathcal{K} \times D \to \mathcal{R}$

$\underline{A(K)}$ // Input: key K
Output: Collision x, x' ($x \neq x'$ but $H(K,x) = H(K,x')$)

Initialize hash table Y

For $x = 1, \ldots, q$:

$y \leftarrow H(K, x)$ // Treat number i as bit string input

If $Y[y] \neq \perp$:

$x' \leftarrow Y[y]$

output x, x'

Size of Y:

$\approx q$

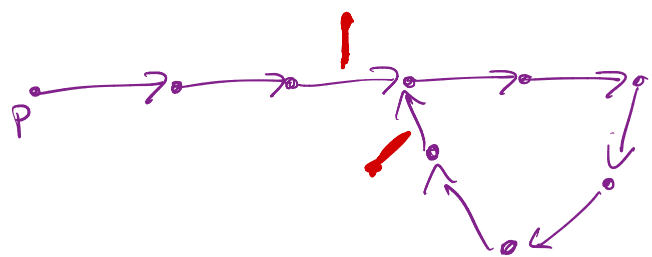
$q \approx \sqrt{|R|}$

$Y = \sqrt{|R|}$

2^{64} space ☹

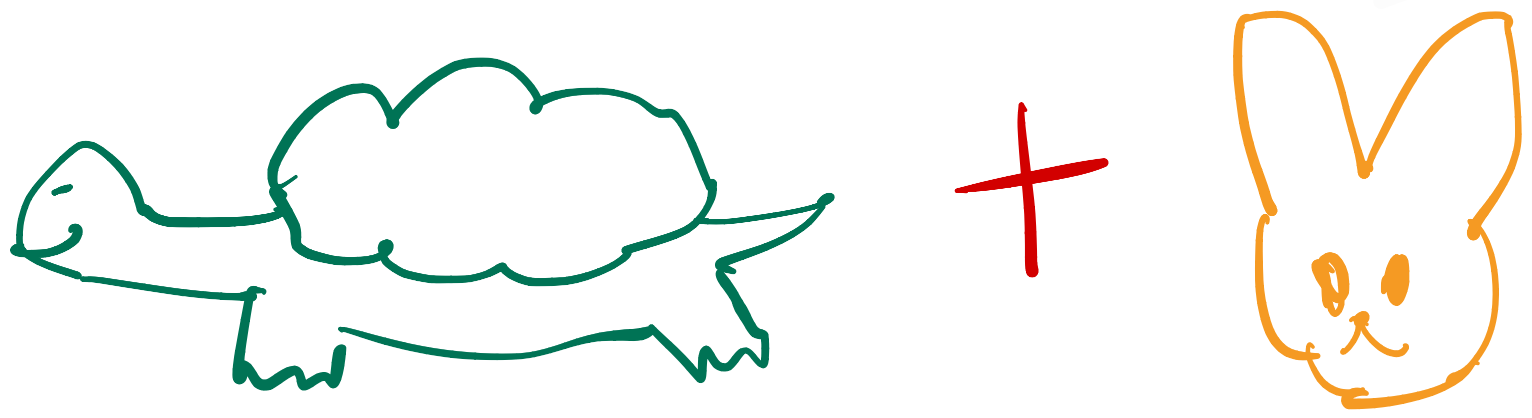
An Interview Question

Suppose you are given a pointer p to a linked list, and told that the list contains a cycle. Show how to find the "colliding pointers" using as little memory as possible.



Solving the Puzzle: Floyd's Cycle Detecting Alg

Finds the cycle in "constant space"!

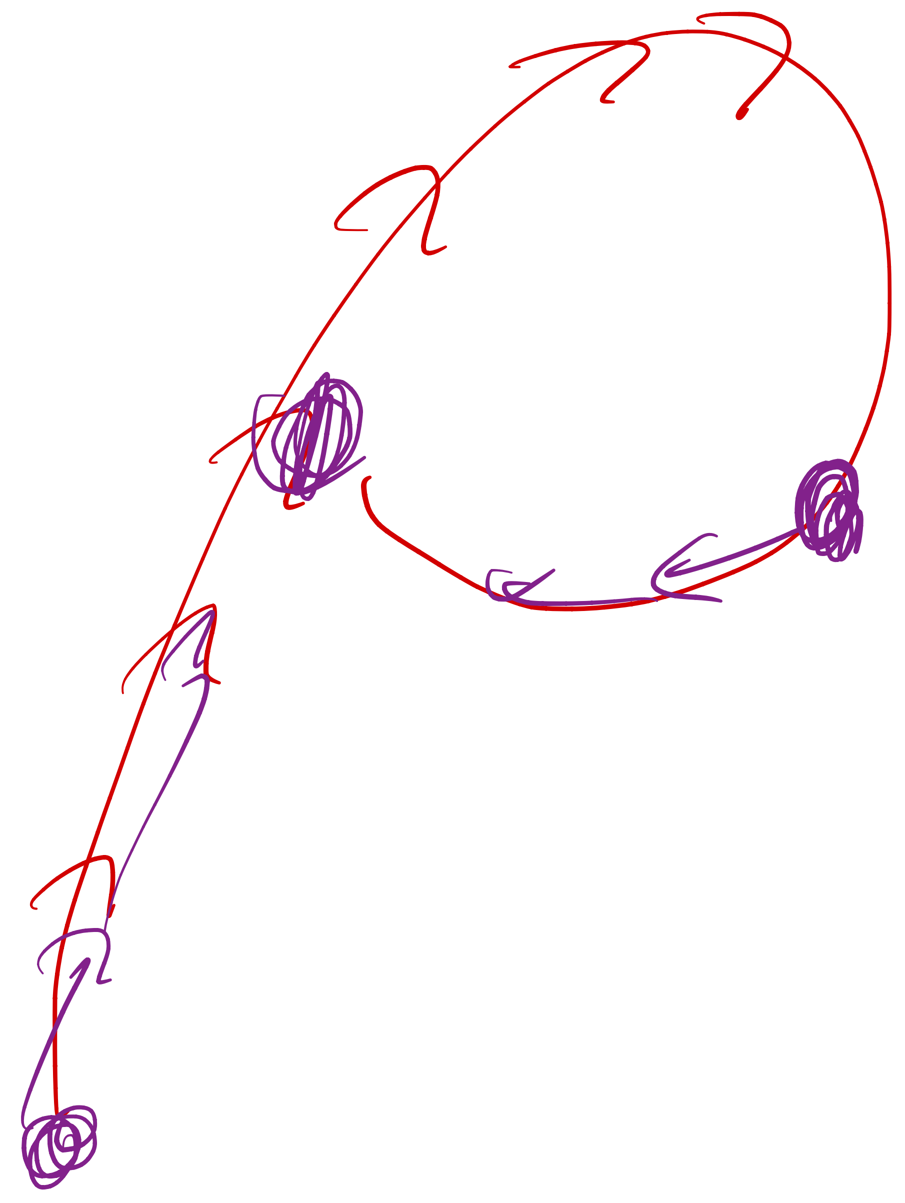


Input pointer

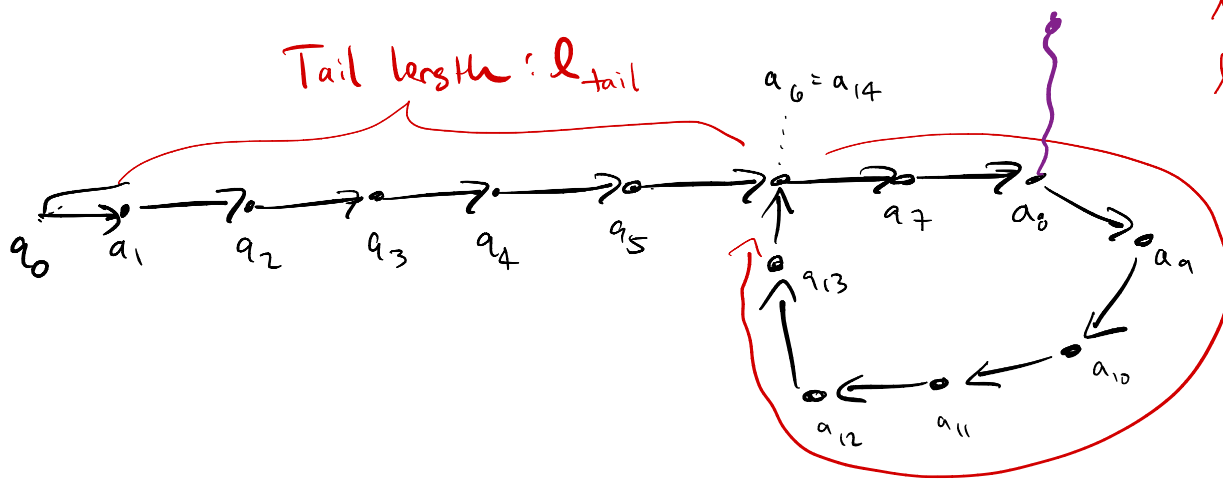
```
tortoise ← head
hare     ← head
While tortoise ≠ hare:
  tortoise ← next(tortoise)
  hare     ← next(next(hare)) # double speed

tortoise ← head     # restart
While next(tortoise) ≠ next(hare):
  tortoise ← next(tortoise)
  hare     ← next(hare)    # single speed

Output tortoise, hare
```



Analysis of Floyd's Alg



$\ell_{tail} = 6$

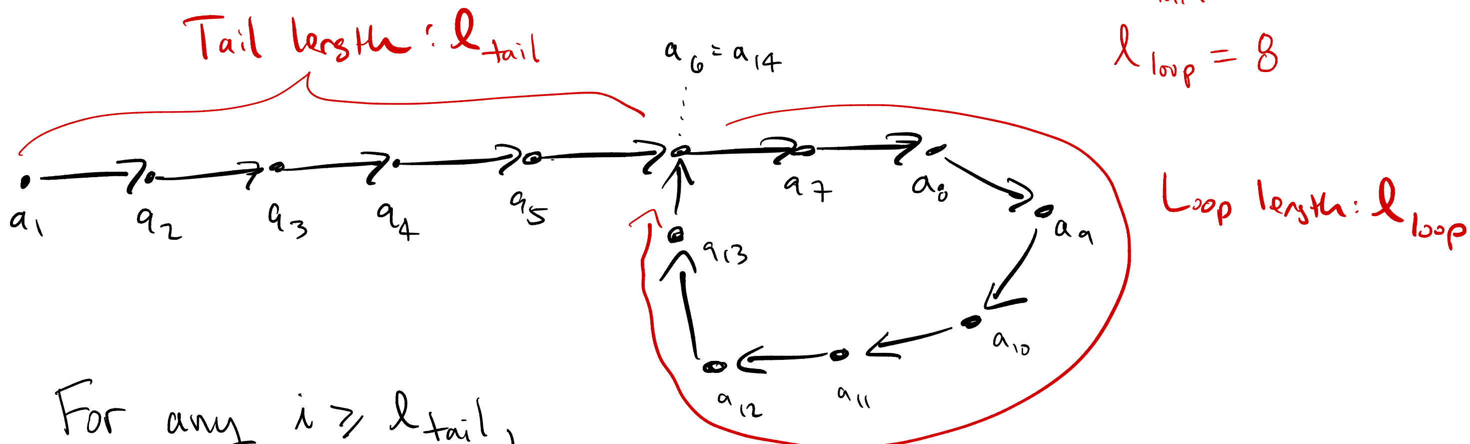
$\ell_{loop} = 8$

Loop length: ℓ_{loop}

tortoise: a_3

hare: a_{16}

Analysis of Floyd's Alg



$\ell_{tail} = 6$

$\ell_{loop} = 8$

Claim 1 For any $i \geq \ell_{tail}$,

$$a_i = a_{i+\ell_{loop}} = a_{i+2\cdot\ell_{loop}} = a_{i+3\cdot\ell_{loop}} = \cdots$$

Moreover, if $i \geq \ell_{tail}$ and $a_i = a_{i+j}$, then j is a multiple of ℓ_{loop}.

Claim 1 For any $i \geq \ell_{tail}$,

$$a_i = a_{i+\ell_{loop}} = a_{i+2\cdot\ell_{loop}} = a_{i+3\cdot\ell_{loop}} = \cdots$$

Moreover, if $i \geq \ell_{tail}$ and $a_i = a_{i+j}$, then j is a multiple of ℓ_{loop}.

Corollary $a_i = a_{2i}$ if and only if $i \geq \ell_{tail}$ and i is a multiple of ℓ_{loop}

Proof ($\Leftarrow$) If $i = k\cdot\ell_{loop}$, $a_{2i} = a_{i+i} = a_{i+k\cdot\ell_{loop}} = a_i$.

($\Rightarrow$) If $a_i = a_{2i}$, then $i \geq \ell_{tail}$ because there are not early repeats.

Moreover, since $a_i = a_{2i} = a_{i+i}$, i is a multiple of ℓ_{loop} by second part of Claim 1.

Analysis of Floyd's Alg

Claim 2 For some $1 \le i \le \ell_{tail} + \ell_{loop}$, $a_i = a_{2i}$.

Proof Take i = smallest multiple of ℓ_{loop} that is at least ℓ_{tail}.

Apply corollary: $a_{2i} = a_{i+i} = a_{i + k \cdot \ell_{loop}} = a_i$.

Claim 3 If i is a multiple of ℓ_{loop}, then $a_{i+\ell_{tail}} = a_{\ell_{tail}}$.

Proof This is just the original claim 1 (but with $i = \ell_{tail}$ there!)

Analysis of Floyd's Algorithm

Putting it together:

1. Alg will find tortoise = hare in at most $\ell_{tail} + \ell_{loop}$ iterations
   ↳ By claim 2, there is $i \le \ell_{tail} + \ell_{loop}$ such that $a_i = a_{2i}$
2. When tortoise = hare, this pointer is exactly ℓ_{tail} steps from the collision.
   ↳ By claim 3, $a_{\ell_{tail}} = a_{i+\ell_{tail}}$ for this i.
3. Restarted walk will arrive at collision in ℓ_{tail} steps as well!

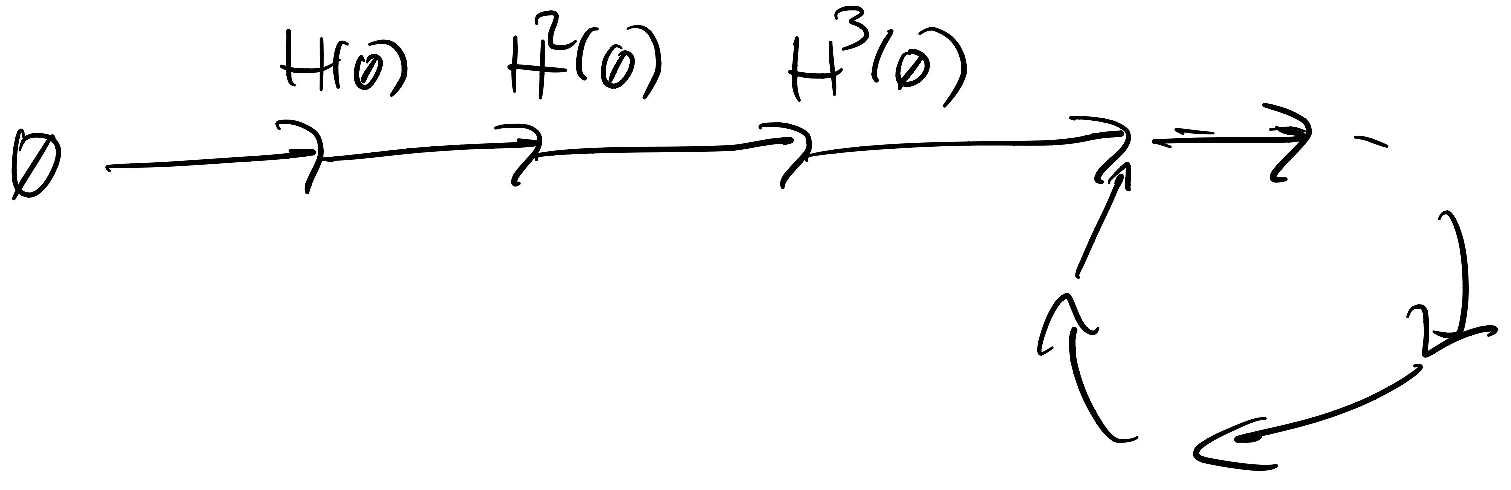
Putting it together for Collision Finding

Define a "list" $H(\emptyset)$, $H(H(\emptyset))$, $H(H(H(\emptyset)))$ ...

```
tortoise ← 0
hare     ← 0
While tortoise ≠ hare:
  tortoise ← H(tortoise)
  hare     ← H(H(hare)) # double speed

tortoise ← head     # restart
While H(tortoise) ≠ H(hare):
  tortoise ← H(tortoise)
  hare     ← H(hare)    # single speed

Output tortoise, hare
```



Heuristically, expect a collision after about $\sqrt{|R|}$ steps.

Rest of analysis is the same!

Run time: $\approx \sqrt{|R|}$

Space: $\approx \emptyset$

	wall clock time	space
BDay w/ 10,000 machines:	$\frac{\sqrt{\lvert R \rvert}}{10{,}000}$	$\sqrt{\lvert R \rvert}$
Floyd's w/ 10,000 machines:	$\sqrt{\lvert R \rvert}$	$\approx \emptyset$

$\rightarrow$ more advanced algs, time / space

"parallelism"