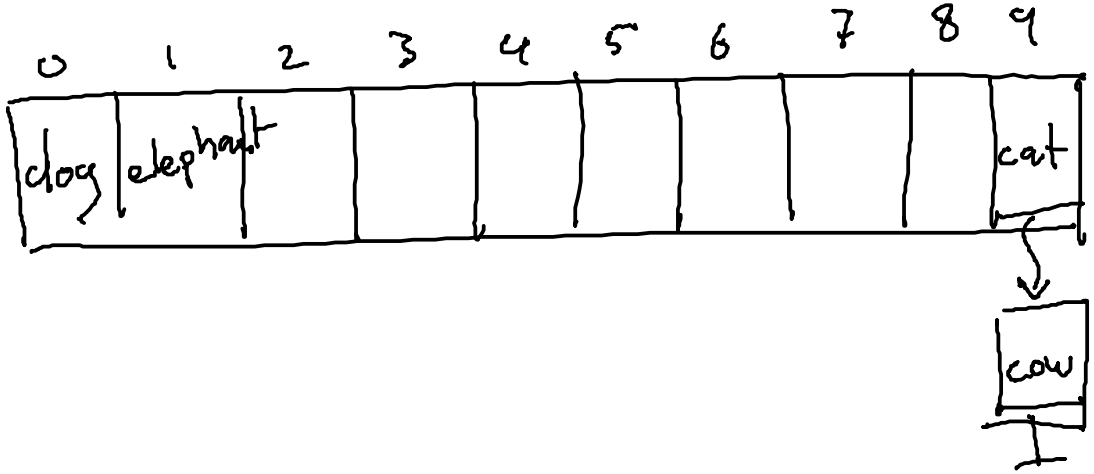
EE 200 Lecture 21: Hash tables

Steven Bell

1 December 2022

Tufts UNIVERSITY

What is a hash table?



dog => 100 % 10 = 0

cat => 99 % 10 = 9

elephant 101 => 1

cow => 99 => 9

Trying to write hash functions

```
int hash(const char* data, int len)
{



}
```

Compare: hash table vs binary tree

	b-tree	hash table
Lookup	log(N)	N
Insertion	log(N)	N
	ordered traversal	no order

Compare: hash table vs array

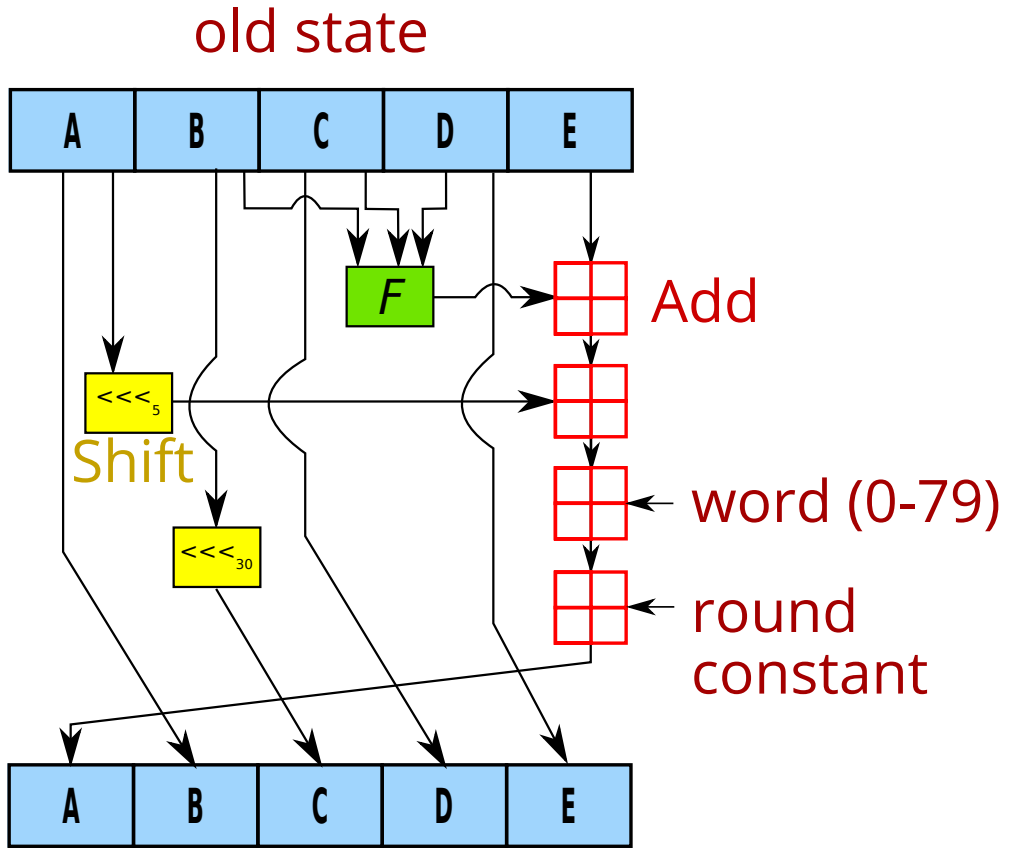
An example hash: SHA-1

Break a message into 512-bit blocks (16x32 bits)

Expand the 16 words into 80 by
xor-ing combinations

For each of the 80 words, compute

Add the state (ABCDE) to the result



https://en.wikipedia.org/wiki/SHA-1

Proof you've got something

Securing passwords

Choose a password:

`mypassword`

hash `3c27714c69a55d106d59fd5c9...`

Enter your password:

`password?`

`de3e8de1b4256df106818fc8e...`

But can be circumvented with a rainbow table

So salt passwords before hashing!

Proof of work

Bitcoin requires "miners" to find a hash with a certain number of 0s.

Git uses SHA-1

Objects and commits all are referenced by their SHA-1 hashes

But SHA-1 is broken

See shattered.io

Crafting a collision is still hard, but it's been done.

Upcoming deadlines:

Classwork 14 and 15 are posted

Classwork 12 and 13: resubmissions due next week (11/8)