

First, a cool program

Shazam



Source: Shazam

CS106B



By the end of today I am going
to show you how this works.

We've Gotten Ahead of Ourselves



Source: The Hobbit

Start at the Beginning



Source: The Hobbit

Hashing

CS 106B

Programming Abstractions

Fall 2016

Stanford University

Computer Science Department



Hashing: The Heart of the Hash Map



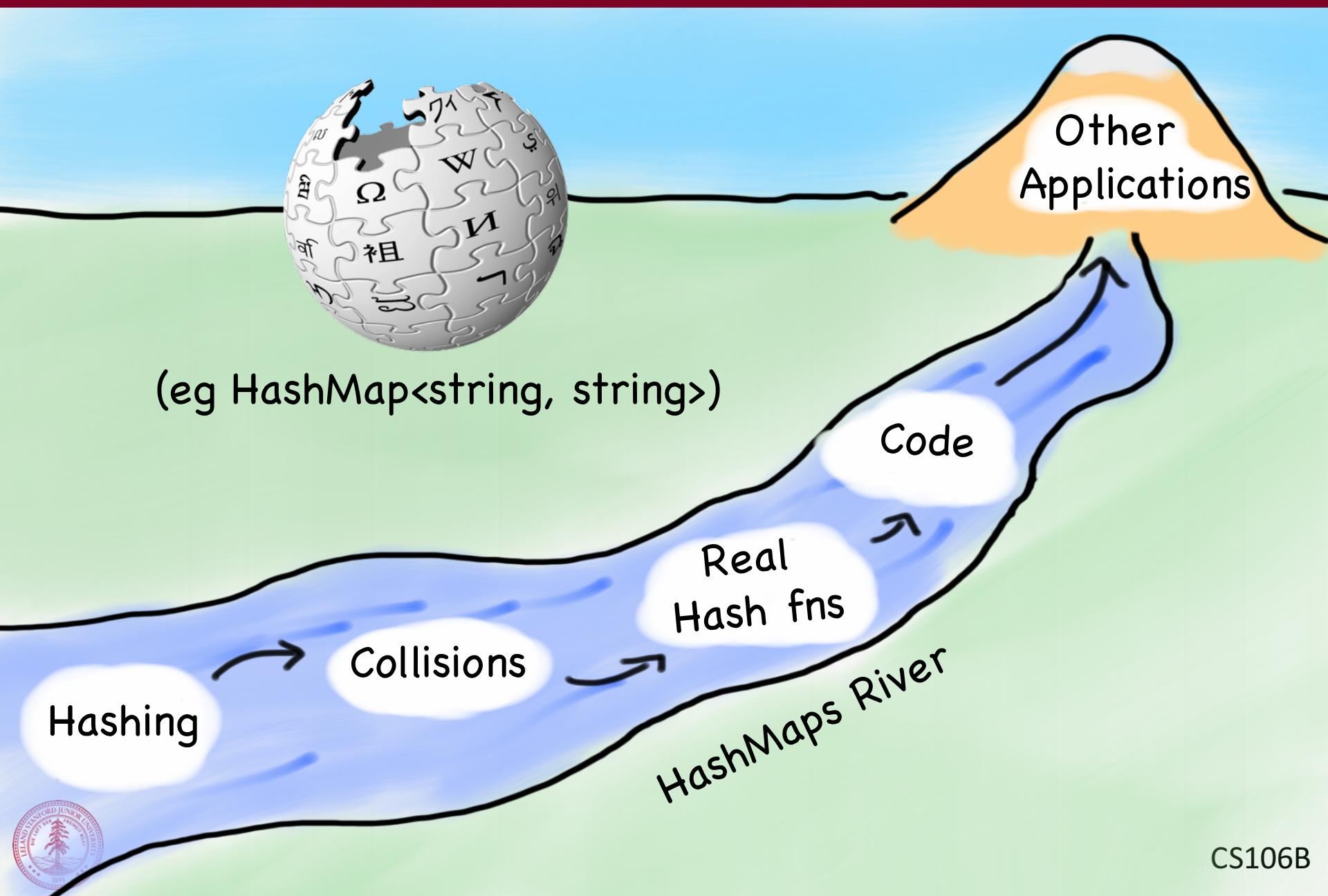
CS106B

Today's Goals

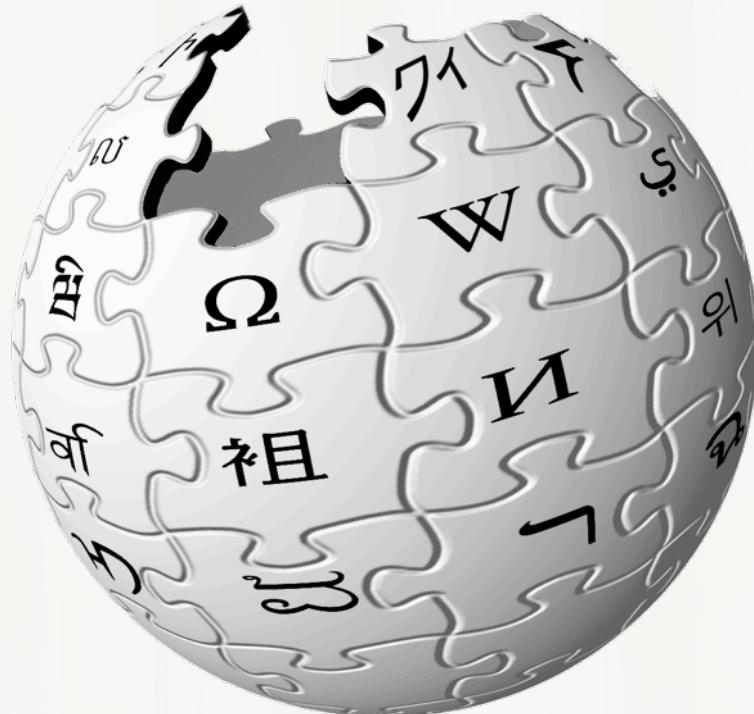
1. Understand how the
HashMap works
2. Add hashing to your
algorithmic toolbox
3. Practice linked lists



Today's Route



First Objective



(eg `HashMap<string, string>`)



Wikipedia API

```
// adds name / text pair to dataset  
put(string articleName, string articleHTML)  
  
// returns corresponding articleHTML  
get(string articleName)  
  
// removes the article  
remove(string articleName)
```



Wikipedia is Big!

5 million articles and growing



Source: xkcd

CS106B



Key Value Pairs

Key = Title, Value = Article

Key:

“Mantis Shrimp”

Value:

Mantis shrimp

From Wikipedia, the free encyclopedia

Mantis shrimp or stomatopods are marine crustaceans, the members of the order Stomatopoda. Most species can grow to around 10 centimetres (3.9 in) in length, though a few species reach up to 38 cm (15 in).^[2] The largest ever caught has a length of 46 cm (18 in) in the ocean near Fort Pierce, Florida of USA.^[3] The carapace of mantis shrimp covers only the rear part of the head and the first four segments of the thorax. There are more than 400 species of Mantis shrimp. Varieties range from shades of brown to vivid colours, and are among the most important predators in many shallow, tropical and sub-tropical marine habitats. Despite being common, they are poorly understood as many species spend most of their life tucked away in burrows and holes.^[4]



Key:

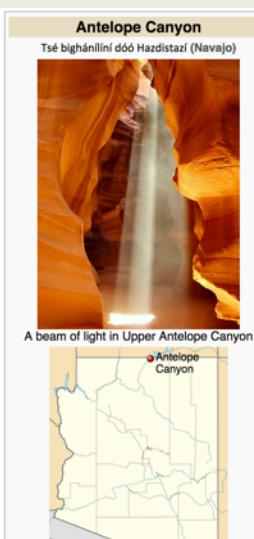
“Antelope Canyon”

Value:

Antelope Canyon is a slot canyon in the American Southwest. It is located on Navajo land east of Page, Arizona. Antelope Canyon includes two separate, photogenic slot canyon sections, referred to individually as Upper Antelope Canyon or The Crack; and Antelope Canyon or The Corkscrew.^[2]

The Navajo name for Upper Antelope Canyon is Tsé bighánílíní, which means "the place where water runs through rocks." Lower Antelope Canyon is Hazdistazí (advertised as "Hasdestwazi" by the Navajo Parks and Recreation Department), or "spiral rock arches." Both are located within the LeChee Chapter of the Navajo Nation.^[4]

Contents [hide]
1 Geology
2 Tourism and photography
2.1 Upper Antelope Canyon



Key Value Pairs

Key = Title, Value = Article

Key:

“Mantis Shrimp”

Value:

```
<p><b>Mantis shrimp</b>
or <b>stomatopods</b> are
marine <a
href="/wiki/Crustacean"
title="Crustacean">crusta
ceans</a>, the members of
the <a
href="/wiki/Order_(biolog
y)" title="Order
(biology)">order</a>
<b>Stomatopoda</b>.
though a few species
reach up to 38&#160;..."
```

Key:

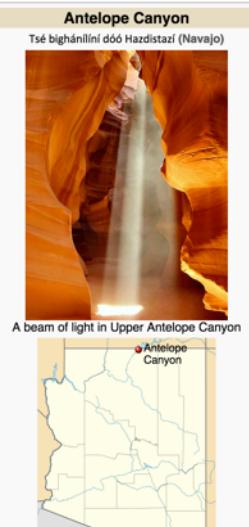
“Antelope Canyon”

Value:

Antelope Canyon is a slot canyon in the American Southwest. It is located on Navajo land east of Page, Arizona. Antelope Canyon includes two separate, photogenic slot canyon sections, referred to individually as *Upper Antelope Canyon* or *The Crack*; and *Antelope Canyon* or *The Corkscrew*.^[2]

The Navajo name for Upper Antelope Canyon is Tsé bighánílíní, which means "the place where water runs through rocks." Lower Antelope Canyon is Hazdistazí (advertised as "Hasdestwazi" by the Navajo Parks and Recreation Department), or "spiral rock arches." Both are located within the LeChee Chapter of the Navajo Nation.^[4]

Contents [hide]
1 Geology
2 Tourism and photography
2.1 Upper Antelope Canyon



Review

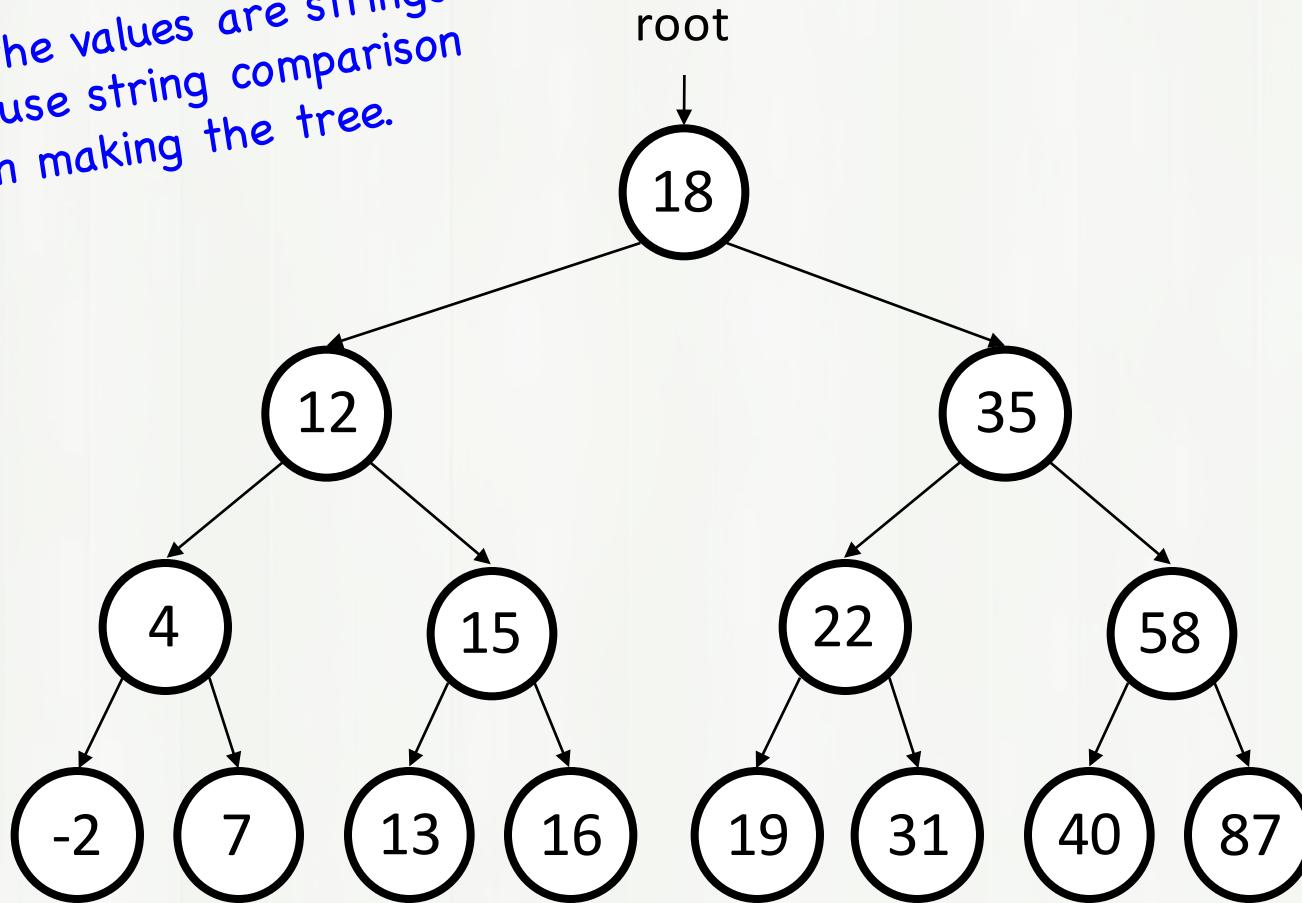
Binary Tree

Search

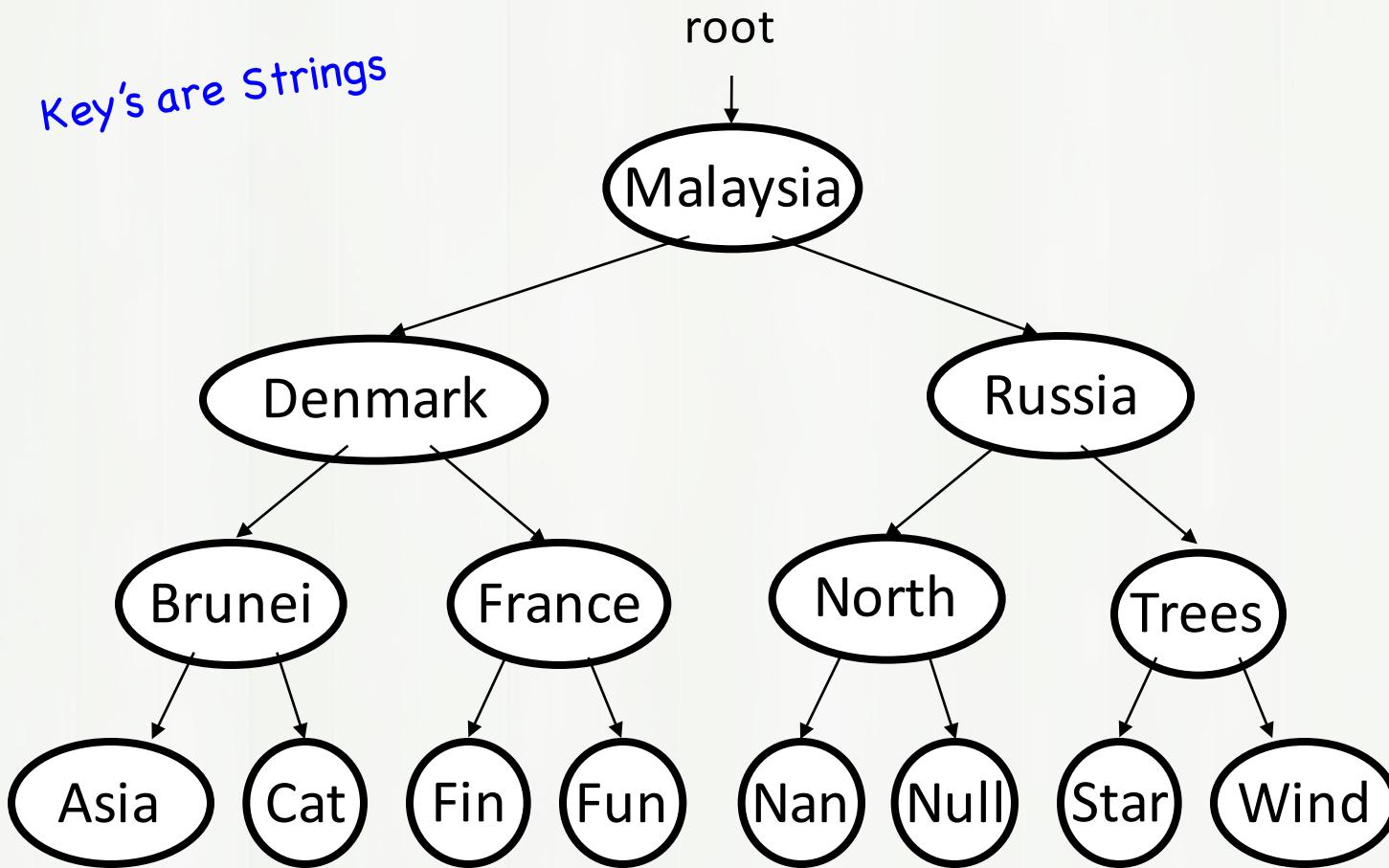


Could Use Binary Search Tree

Except the values are strings
and we use string comparison
when making the tree.



Could Use Binary Search Tree



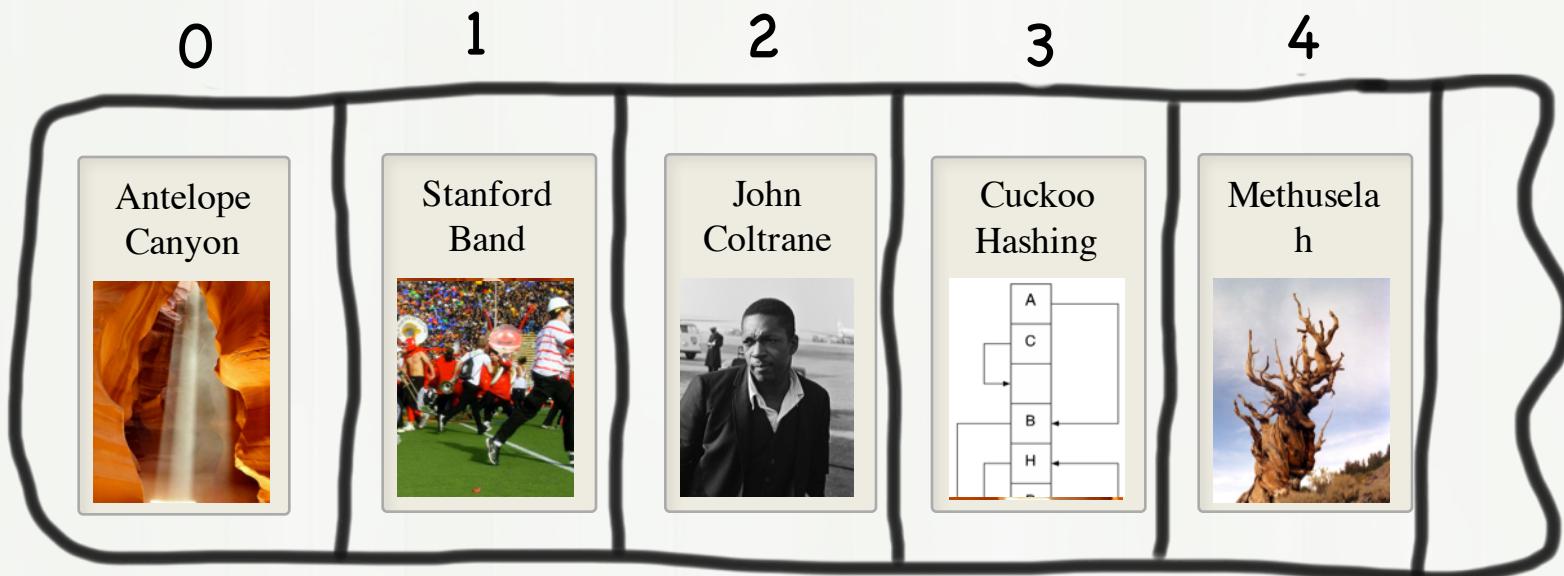
$$\log_2 5 \text{ million} = 22$$

Today we will see something even
better.

Thought Experiment

What if: we store articles in one gigantic vector

Wikipedia:



Thought Experiment

What if: we want to get ("Mantis shrimp")?

Wikipedia:

0

1

2

3

4

Antelope
Canyon



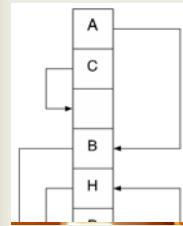
Stanford
Band



John
Coltrane



Cuckoo
Hashing



Methusela
h



Thought Experiment

What if: we want to get ("Mantis shrimp")?

Wikipedia:

0

1

2

3

4

Antelope
Canyon



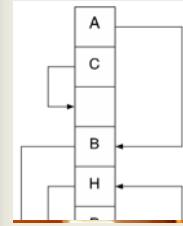
Stanford
Band



John
Coltrane



Cuckoo
Hashing



Methusela
h



Thought Experiment

What if: we want to get ("Mantis shrimp")?

Wikipedia:

0

1

2

3

4

Antelope
Canyon



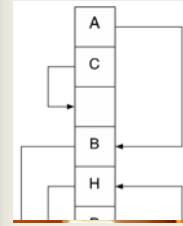
Stanford
Band



John
Coltrane



Cuckoo
Hashing



Methusela
h



Thought Experiment

What if: we want to get ("Mantis shrimp")?

Wikipedia:

0

1

2

3

4

Antelope
Canyon



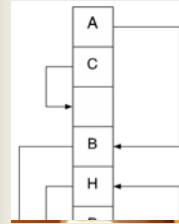
Stanford
Band



John
Coltrane



Cuckoo
Hashing



Methusela
h



Thought Experiment

What if: we want to get ("Mantis shrimp")?

Wikipedia:

0

1

2

3

4

Antelope
Canyon



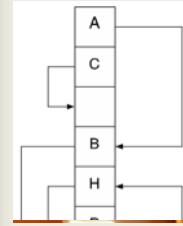
Stanford
Band



John
Coltrane



Cuckoo
Hashing



Methusela
h



Has to be a better way!

What if: there was a way to know
where our article belonged just by
looking at the title?

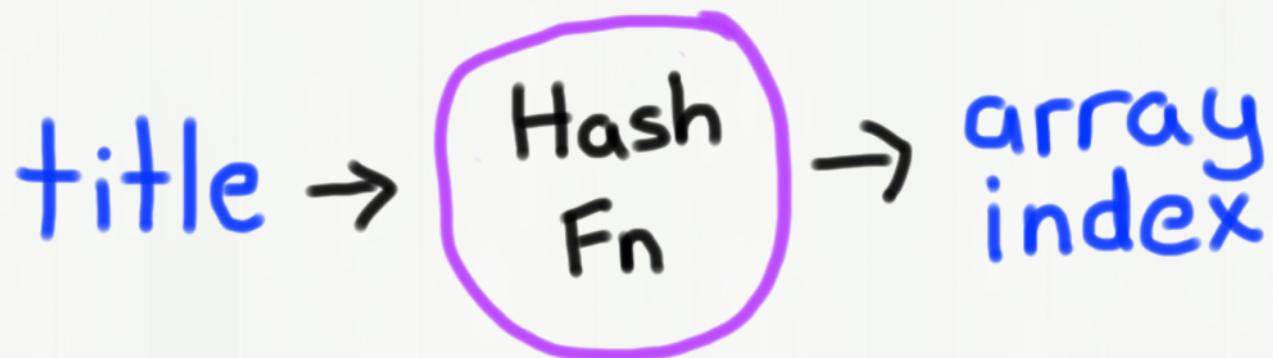
Hash Function



```
int hash(string key);
```



Hash Function

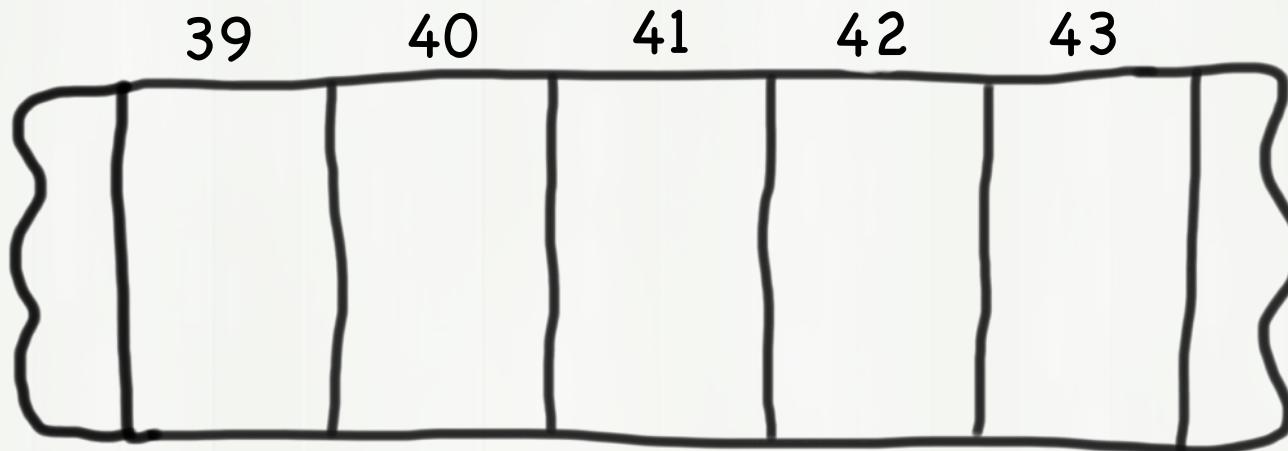


```
int hash(string title);
```



Wikipedia with Hashing

Wikipedia:

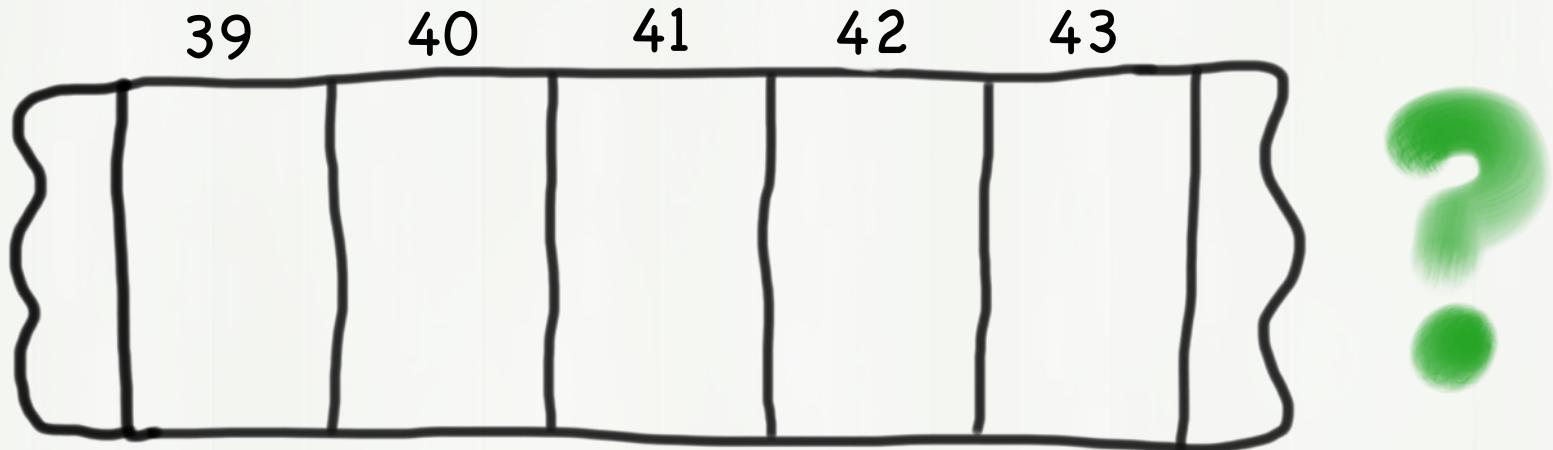


Hash
Fn



```
put("mantis shrimp", html)
```

Wikipedia:

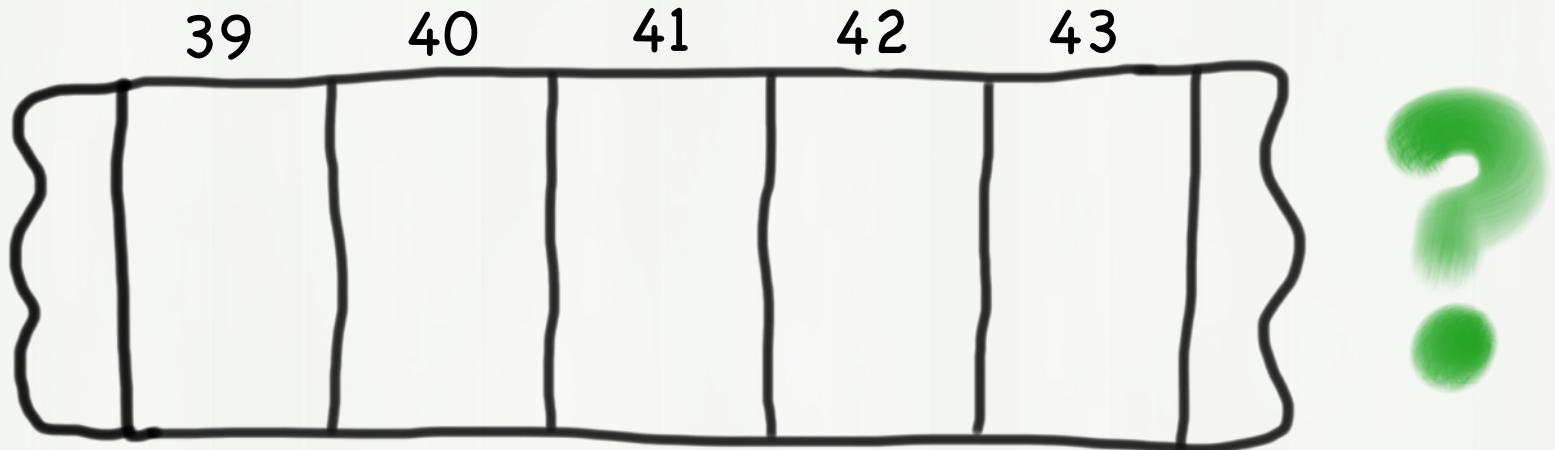


Hash
Fn



```
put("mantis shrimp", html)
```

Wikipedia:

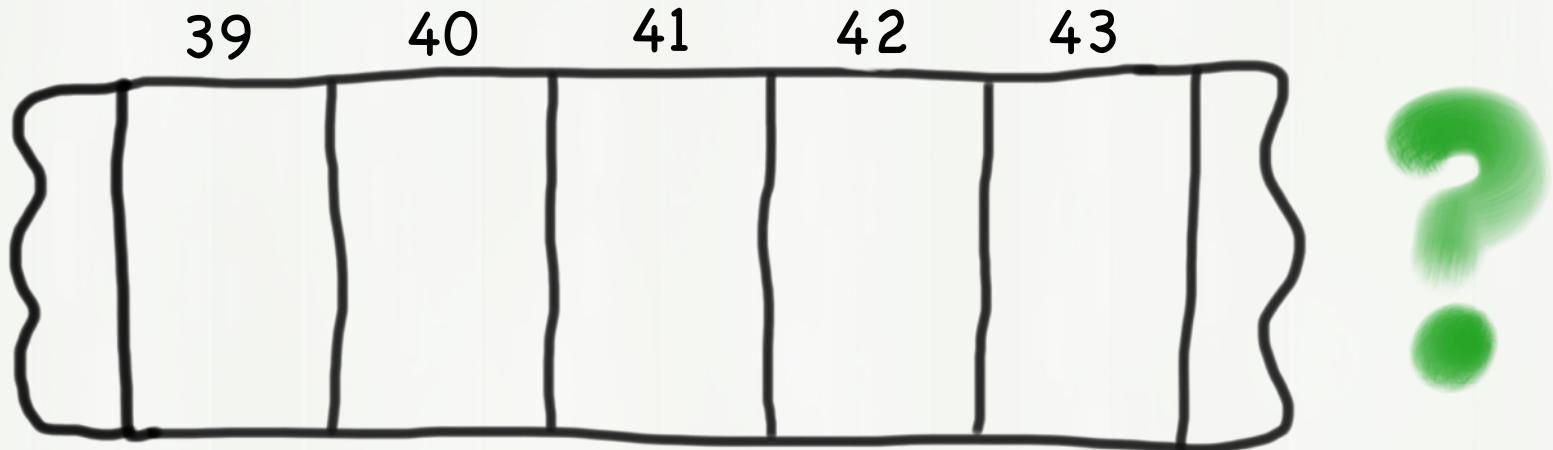


"mantis
shrimp" → Hash
Fn



```
put("mantis shrimp", html)
```

Wikipedia:

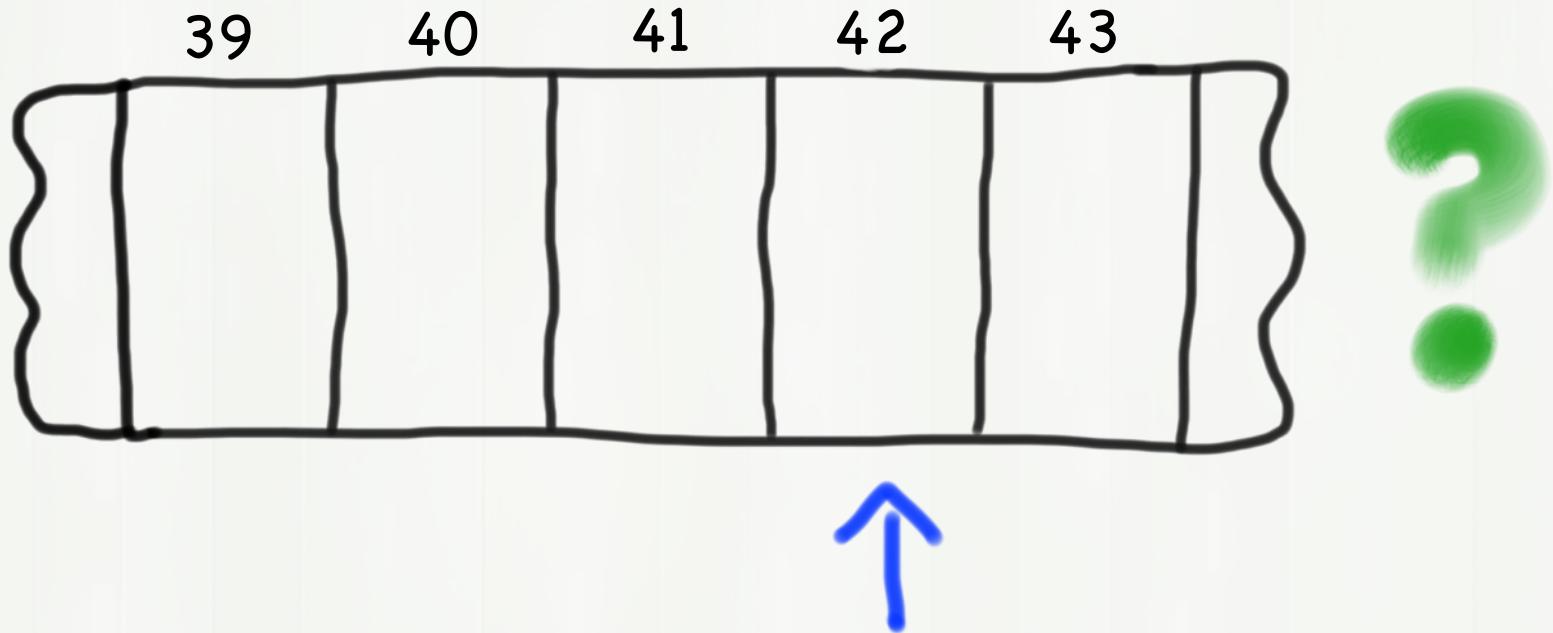


"mantis
shrimp" → Hash
Fn → 42



```
put("mantis shrimp", html)
```

Wikipedia:

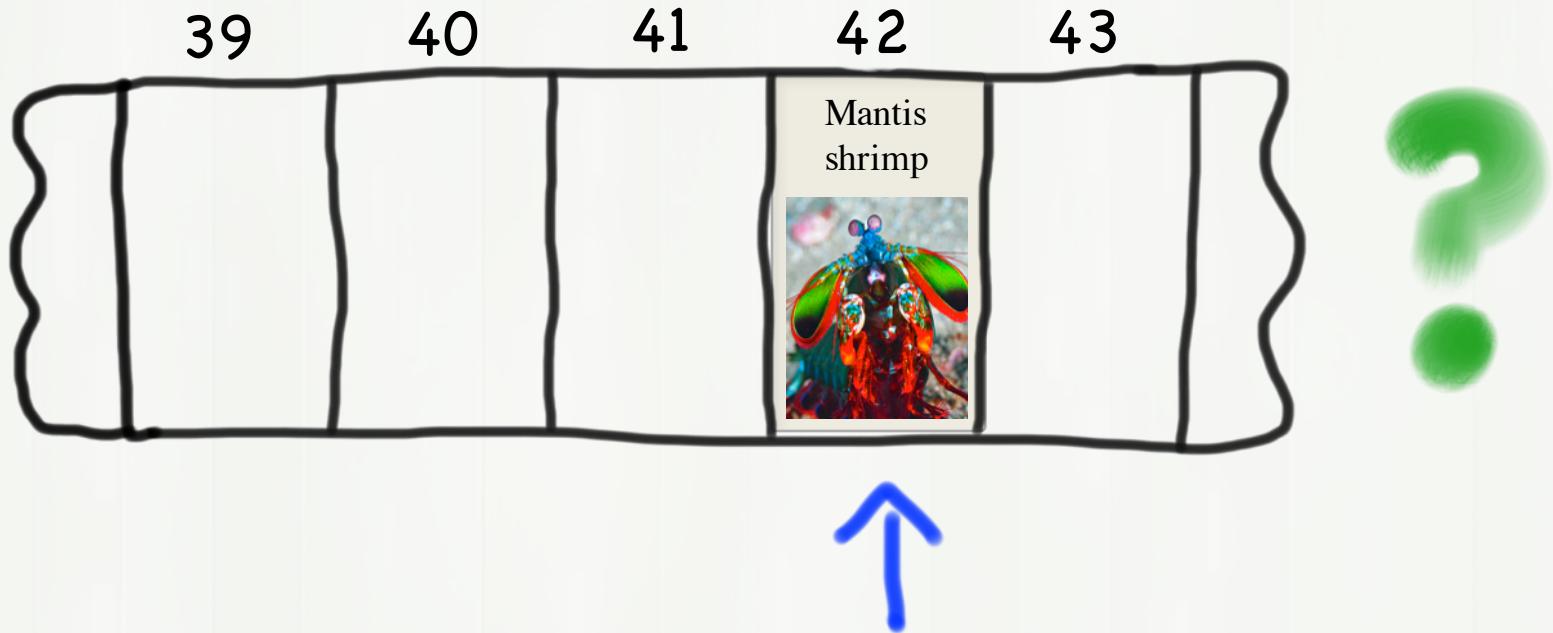


"mantis
shrimp" → Hash
Fn → 42



```
put("mantis shrimp", html)
```

Wikipedia:



"mantis
shrimp" → Hash
Fn → 42



Wonderful!

get("mantis shrimp")

Wikipedia:



Hash
Fn



get("mantis shrimp")

Wikipedia:



"mantis
shrimp" → Hash
Fn



get("mantis shrimp")

Wikipedia:

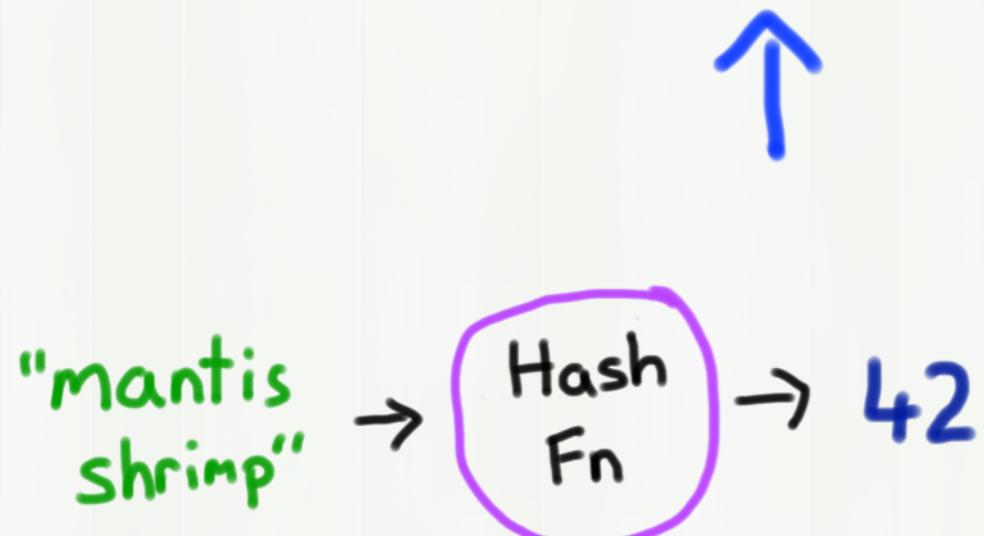


"mantis
shrimp" → Hash
Fn → 42



get("mantis shrimp")

Wikipedia:



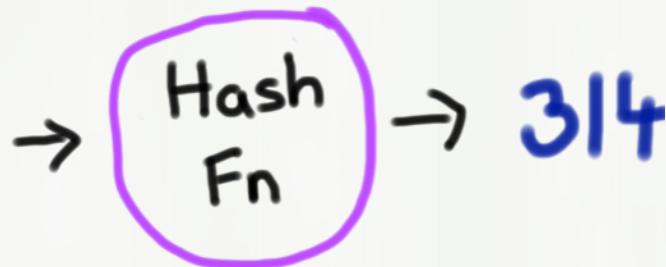


Source: Planet Animal Zone

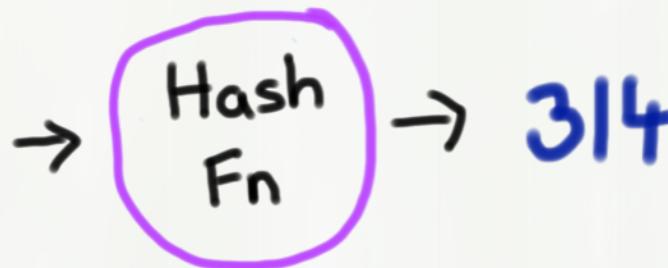
Property #1: Consistent

If you pass in the same input, you will *always* get the same index.

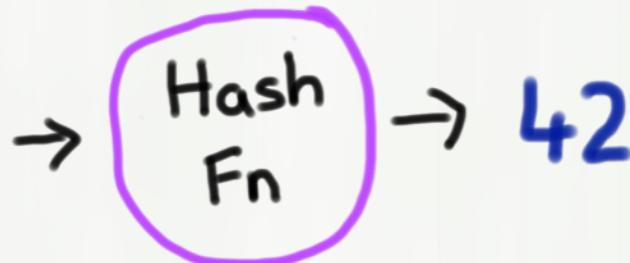
"stanford
band"



"stanford
band"

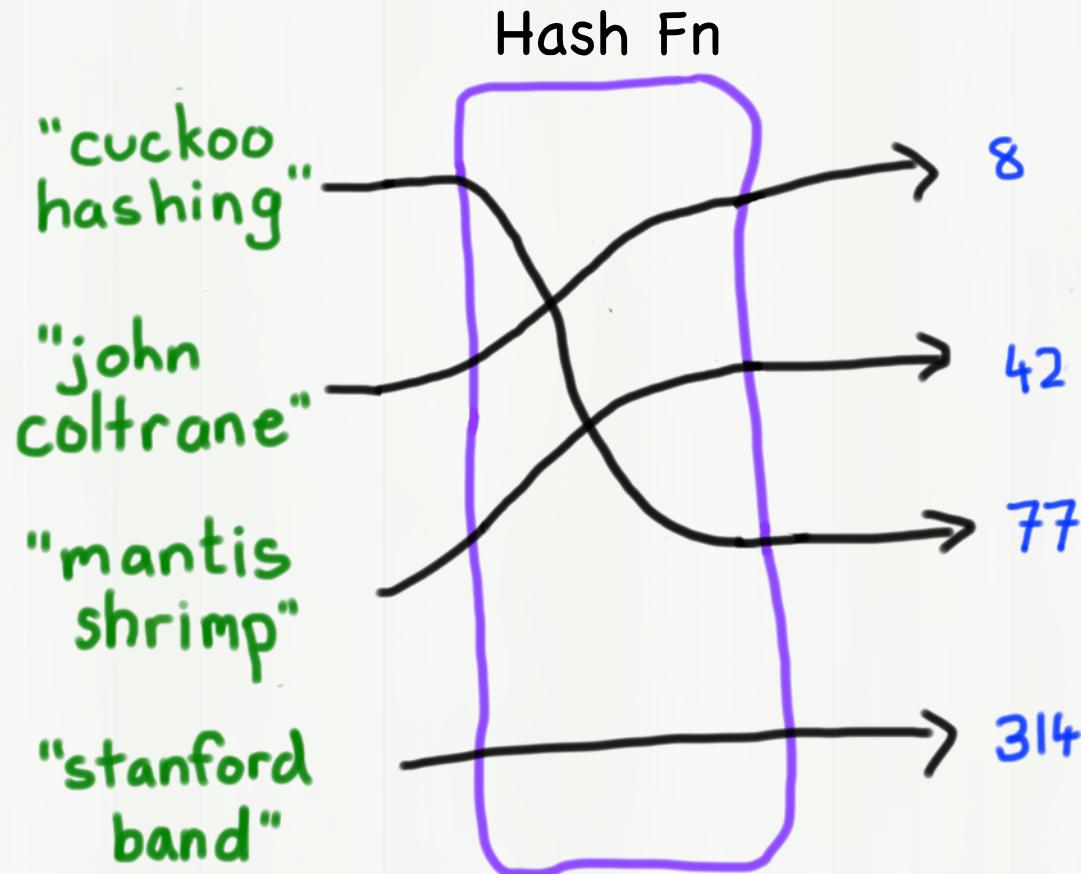


"mantis
shrimp"



Property #2: Well Distributed

If you pass in the different inputs, a hash function will return different indices as often as possible.



What Does a Hash Function Look Like?

The Hashing Hat

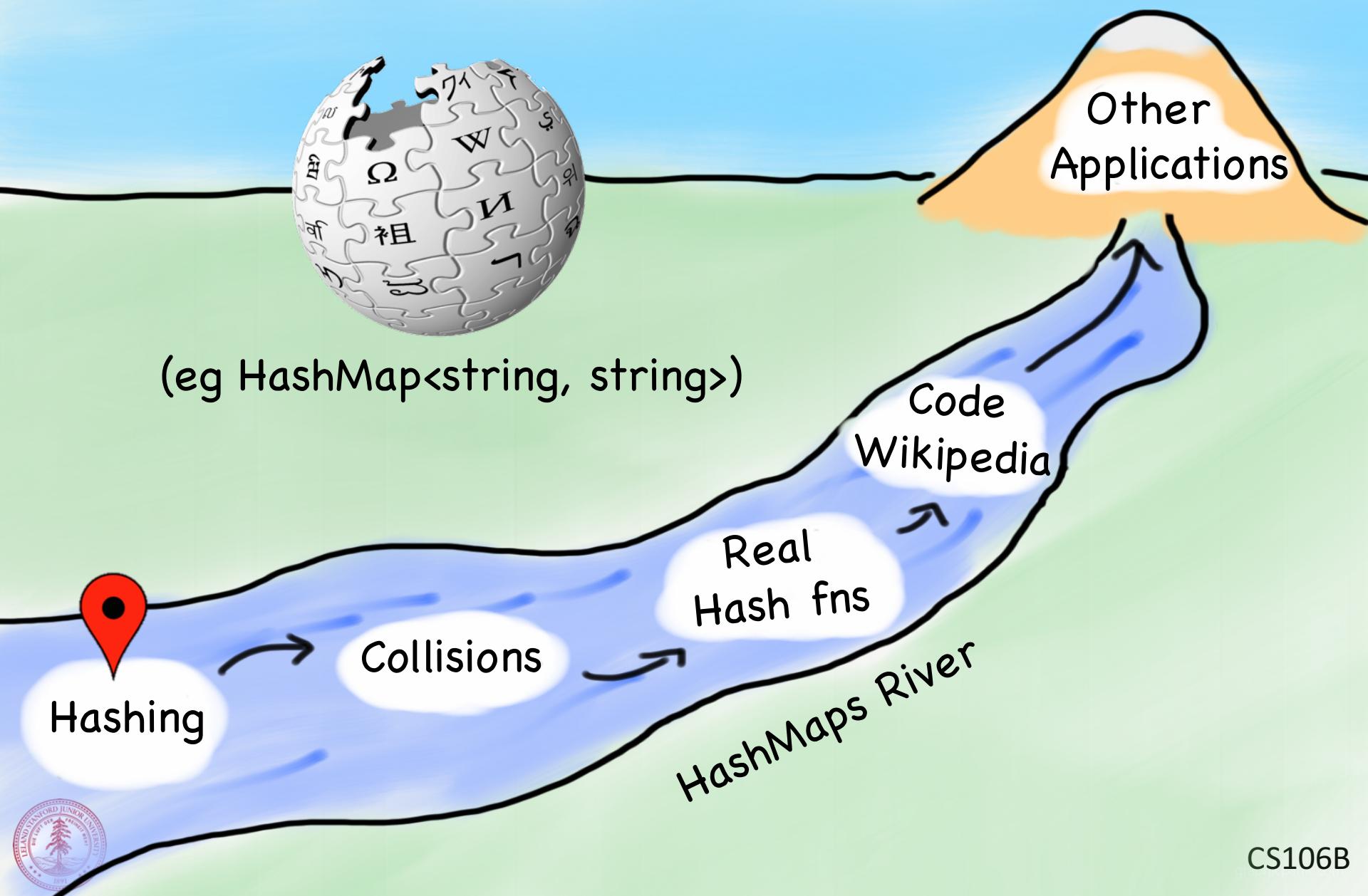
The sorting hat looks
at its “key” and
chooses a “bucket” to
put it into



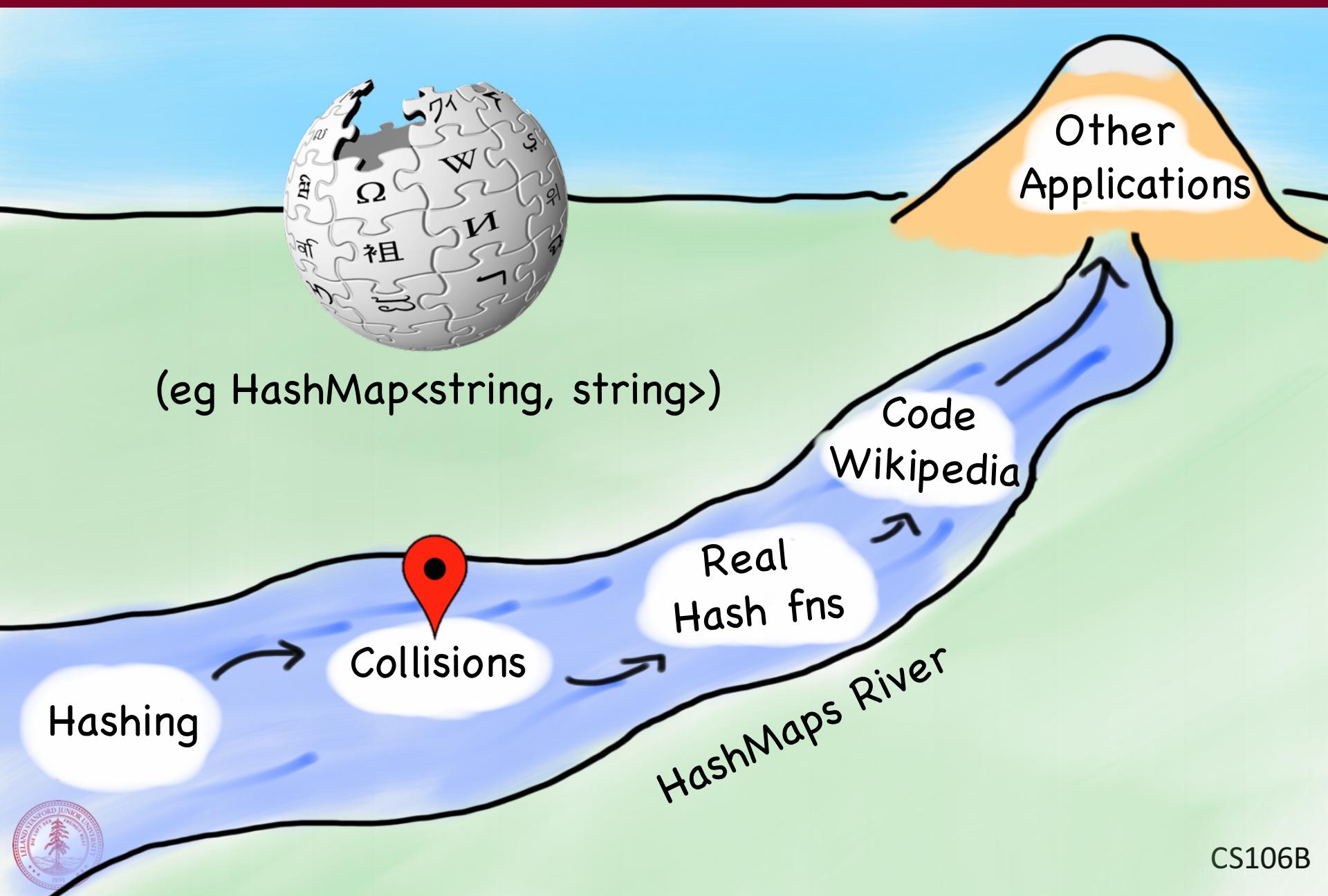
Volunteer



Today's Route

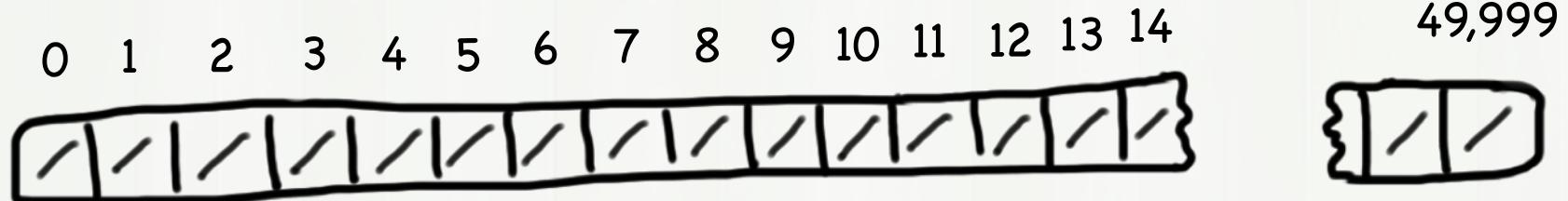


Today's Route



HashMap

data:



of linked lists Dynamic array

Article ** data;

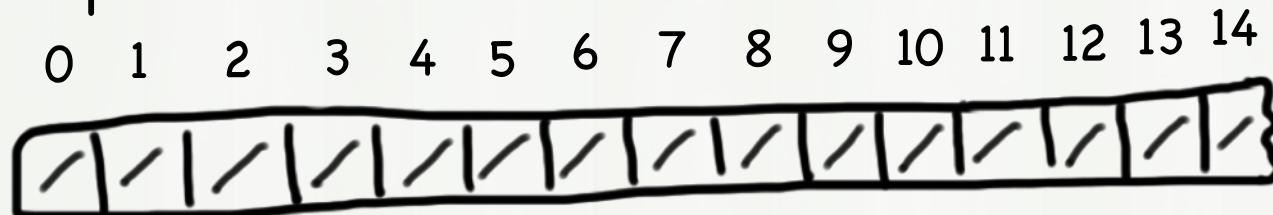
where the node type is Article

```
Article ** data = new Article* [NUM_BUCKETS];
```



Wikipedia

Wikipedia:



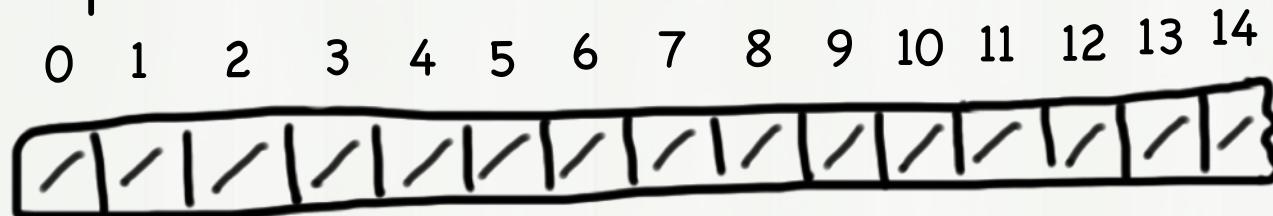
49,999



CS106B

HashMap

Wikipedia:



49,999

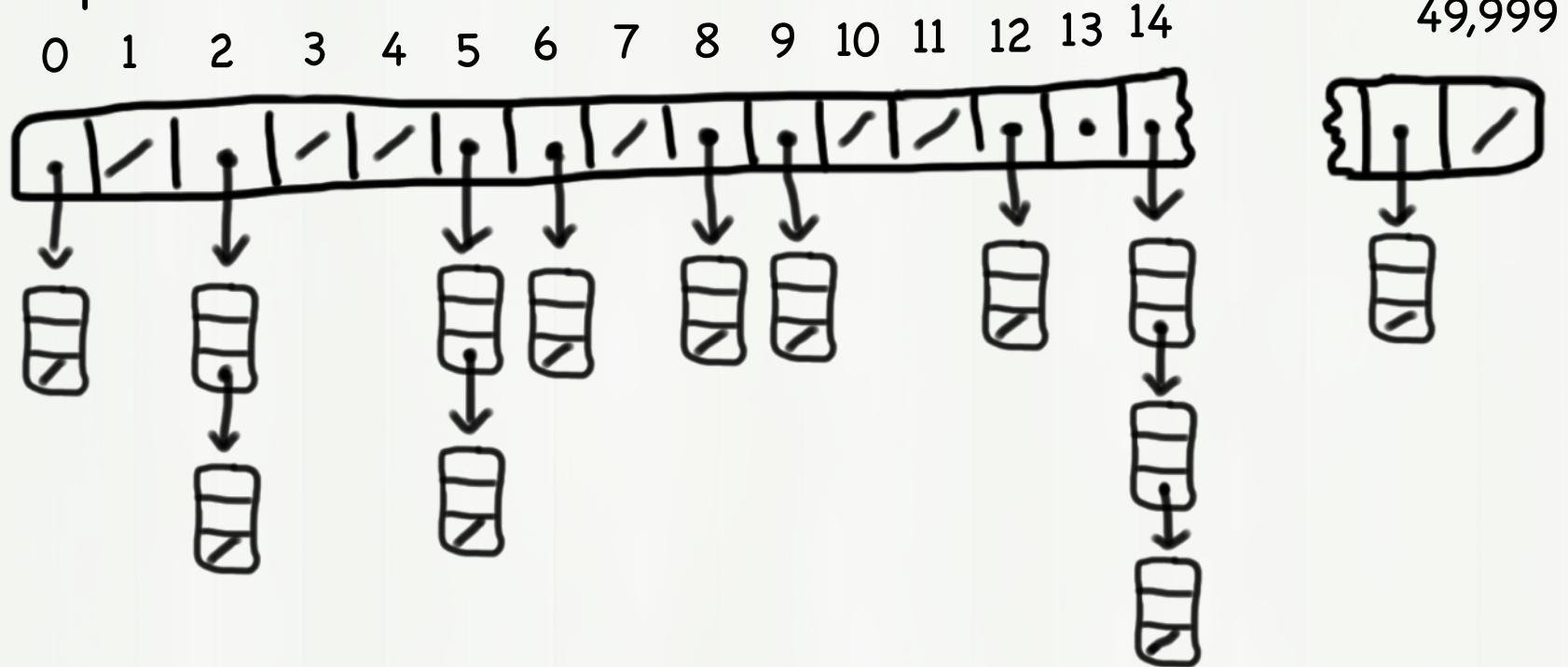


Hash
Fn



HashMap

Wikipedia:



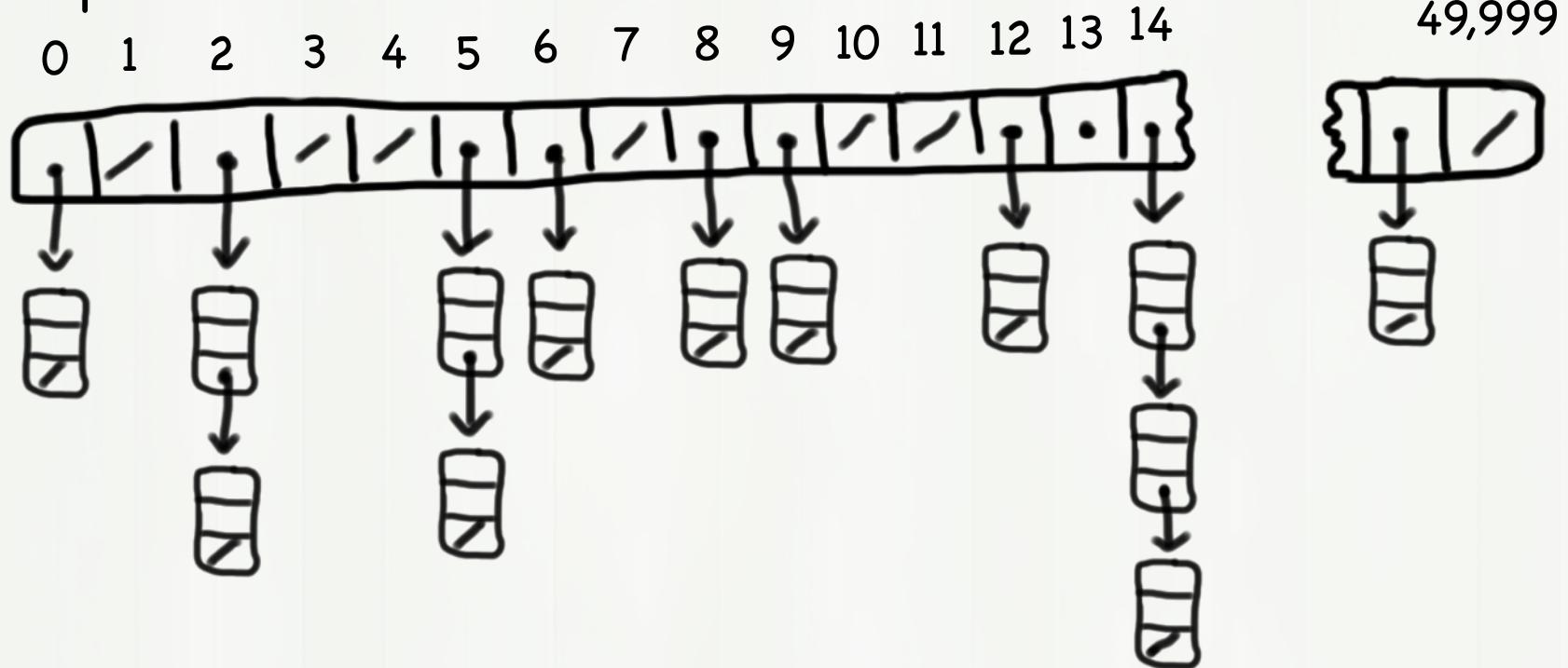
Hash
Fn



Our old friend, linked lists!

HashMap

Wikipedia:



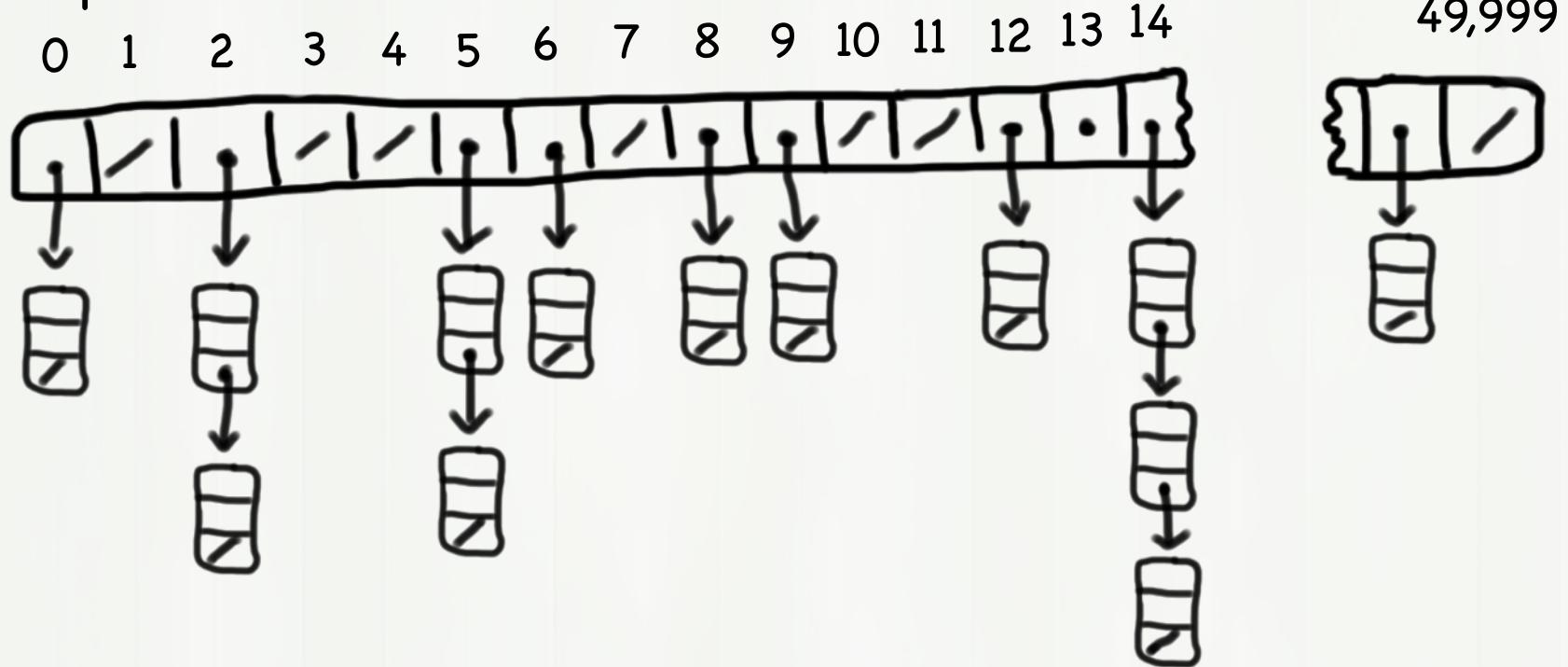
Article:

Key
Value
Next



HashMap

Wikipedia:

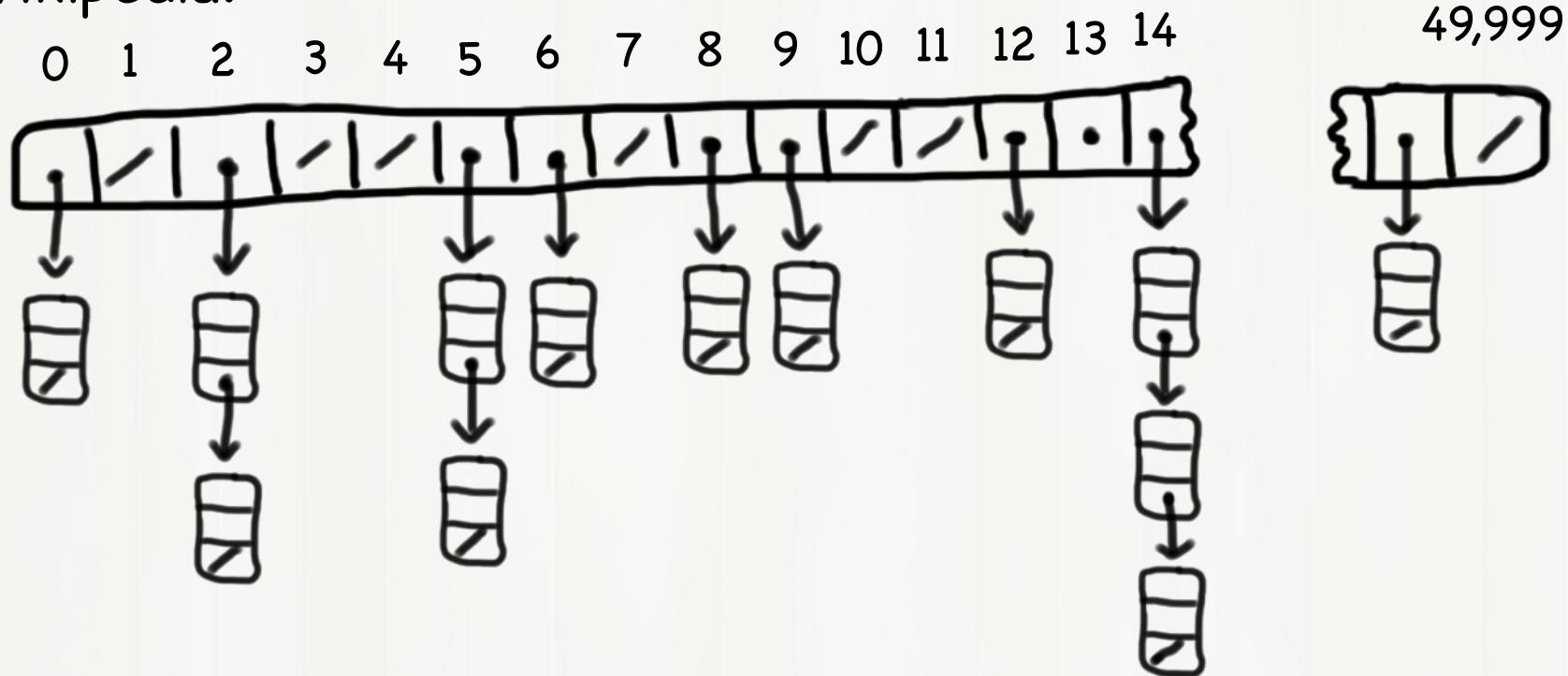


Hash
Fn



get("Antelope Canyon")

Wikipedia:

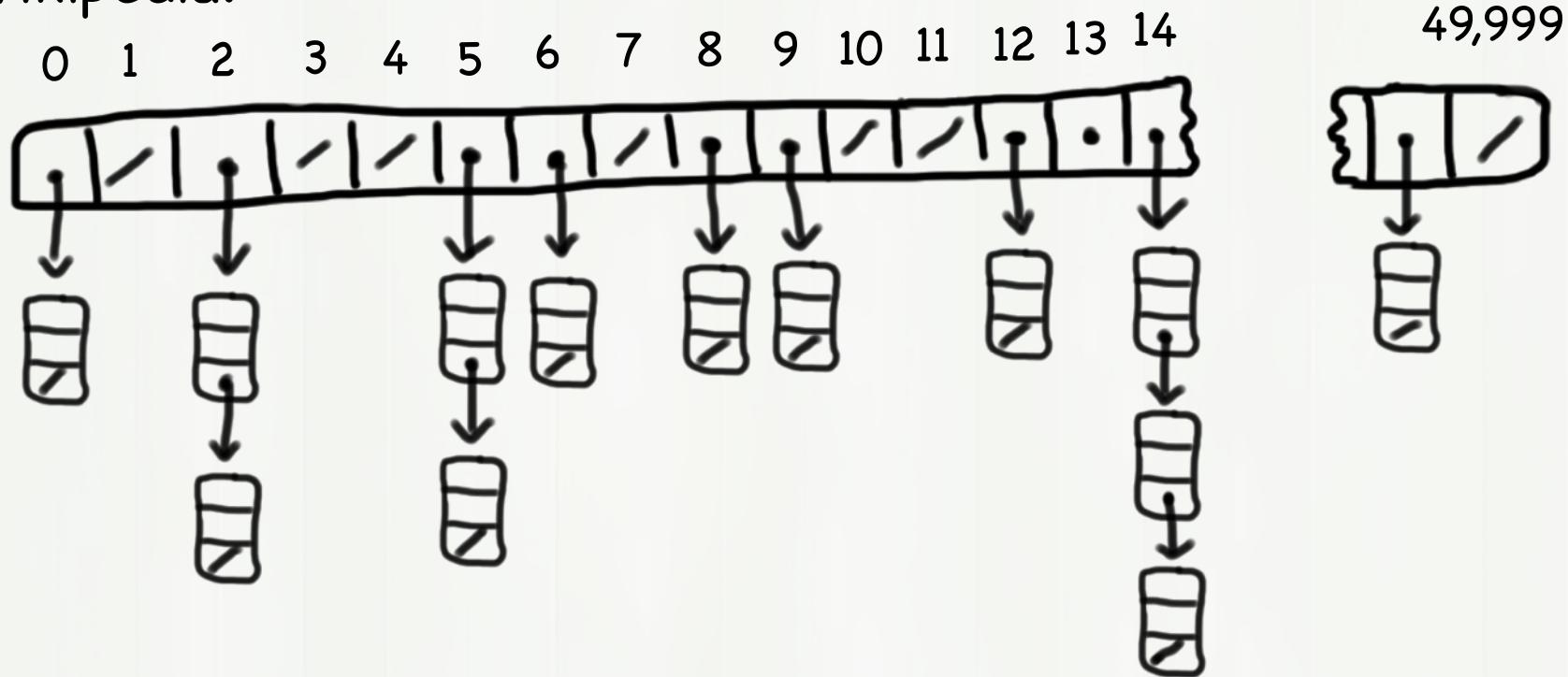


Hash
Fn



get("Antelope Canyon")

Wikipedia:

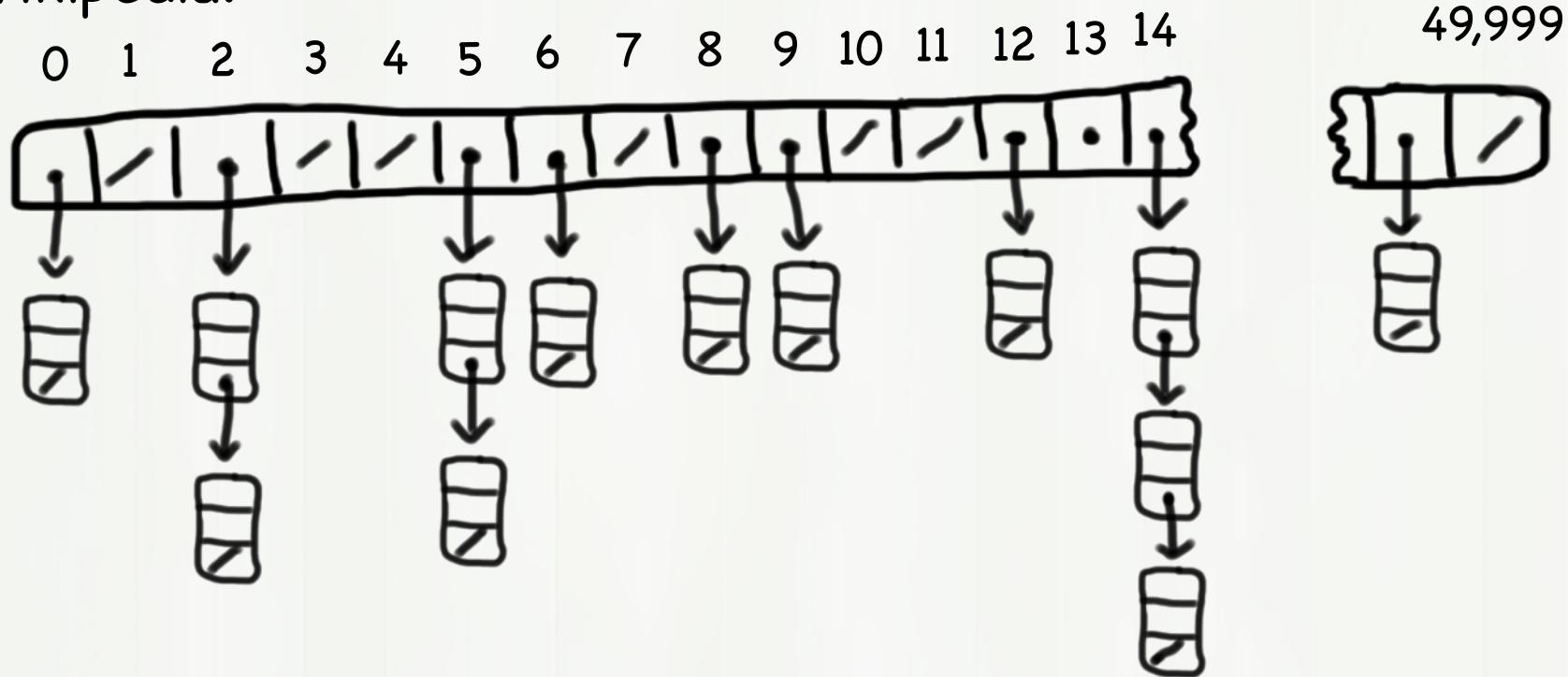


"Antelope
Canyon" → Hash
Fn



get("Antelope Canyon")

Wikipedia:

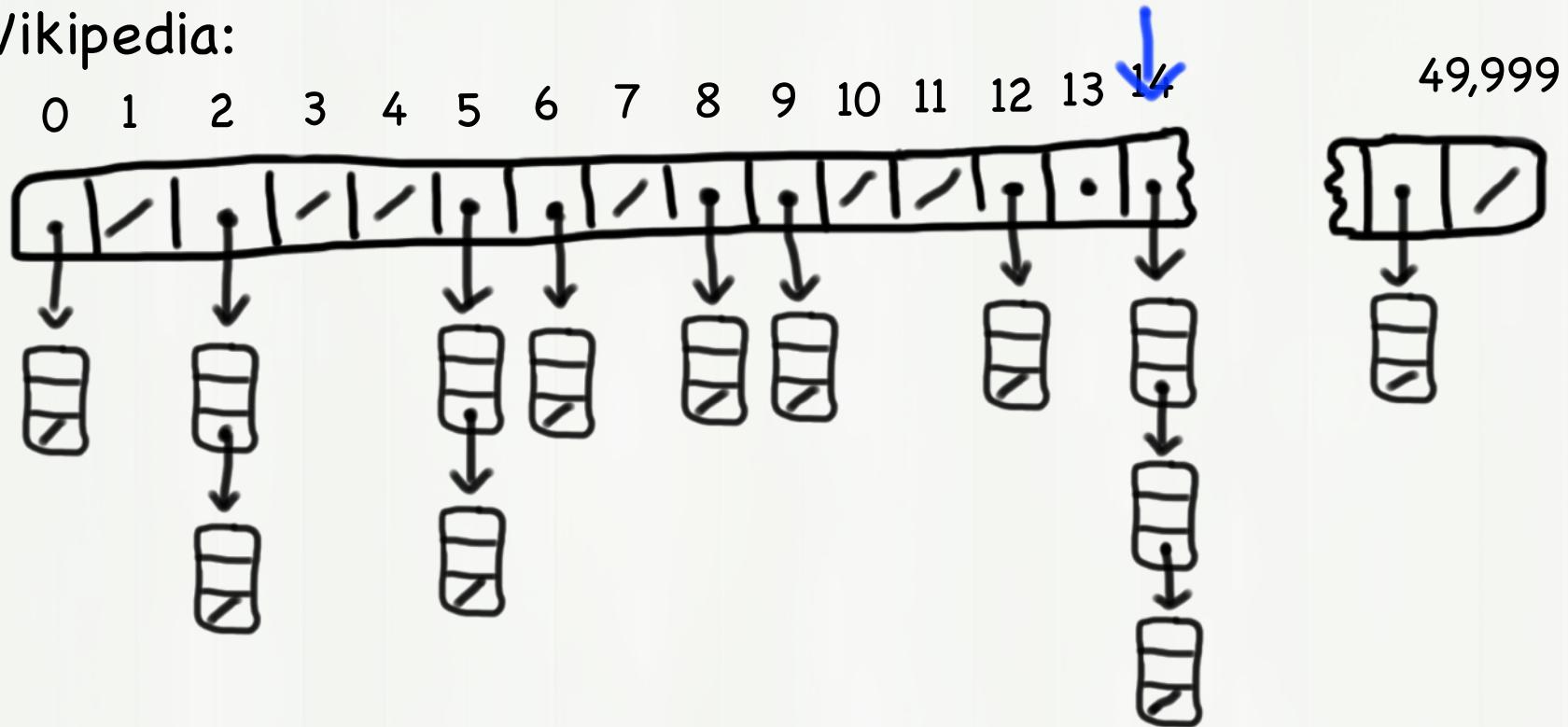


"Antelope
Canyon" → Hash
Fn → 14



get("Antelope Canyon")

Wikipedia:

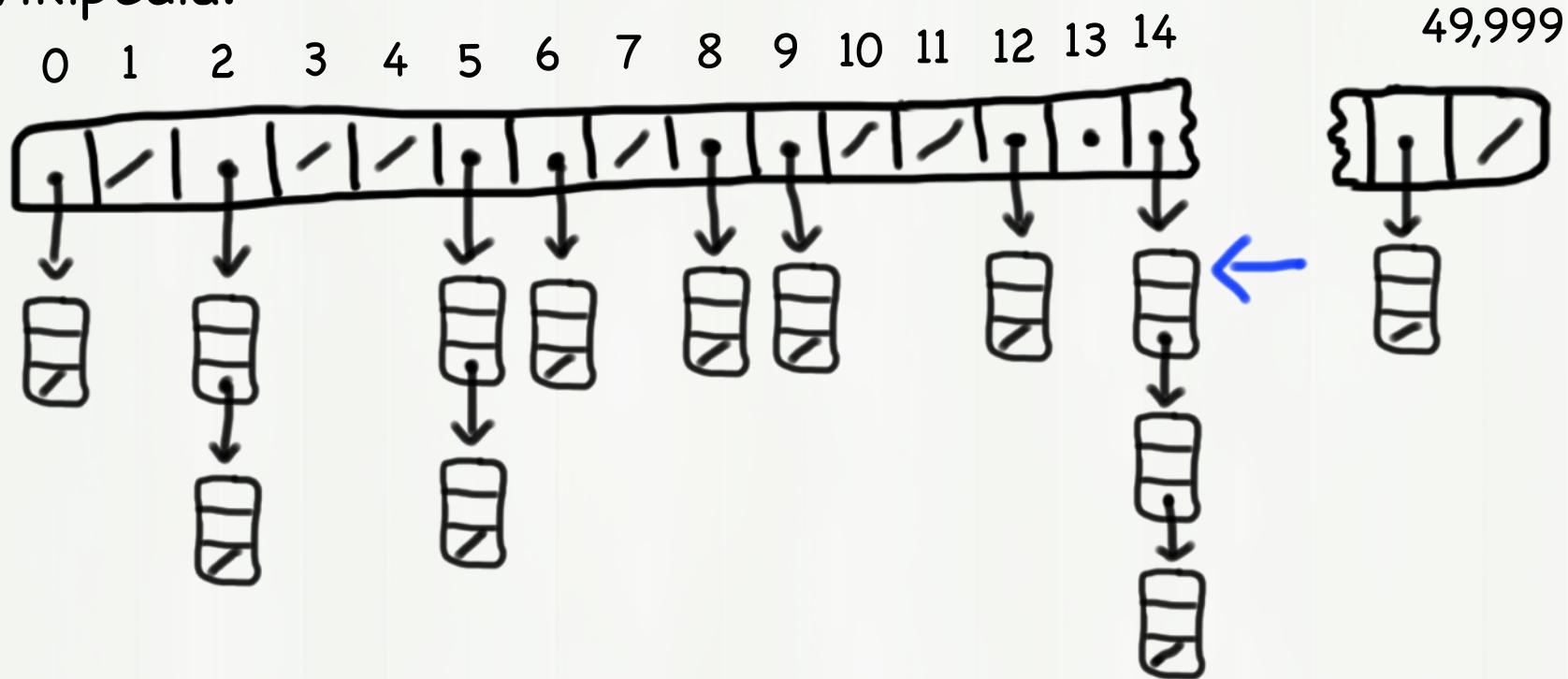


"Antelope
Canyon" → Hash
Fn → 14



get("Antelope Canyon")

Wikipedia:

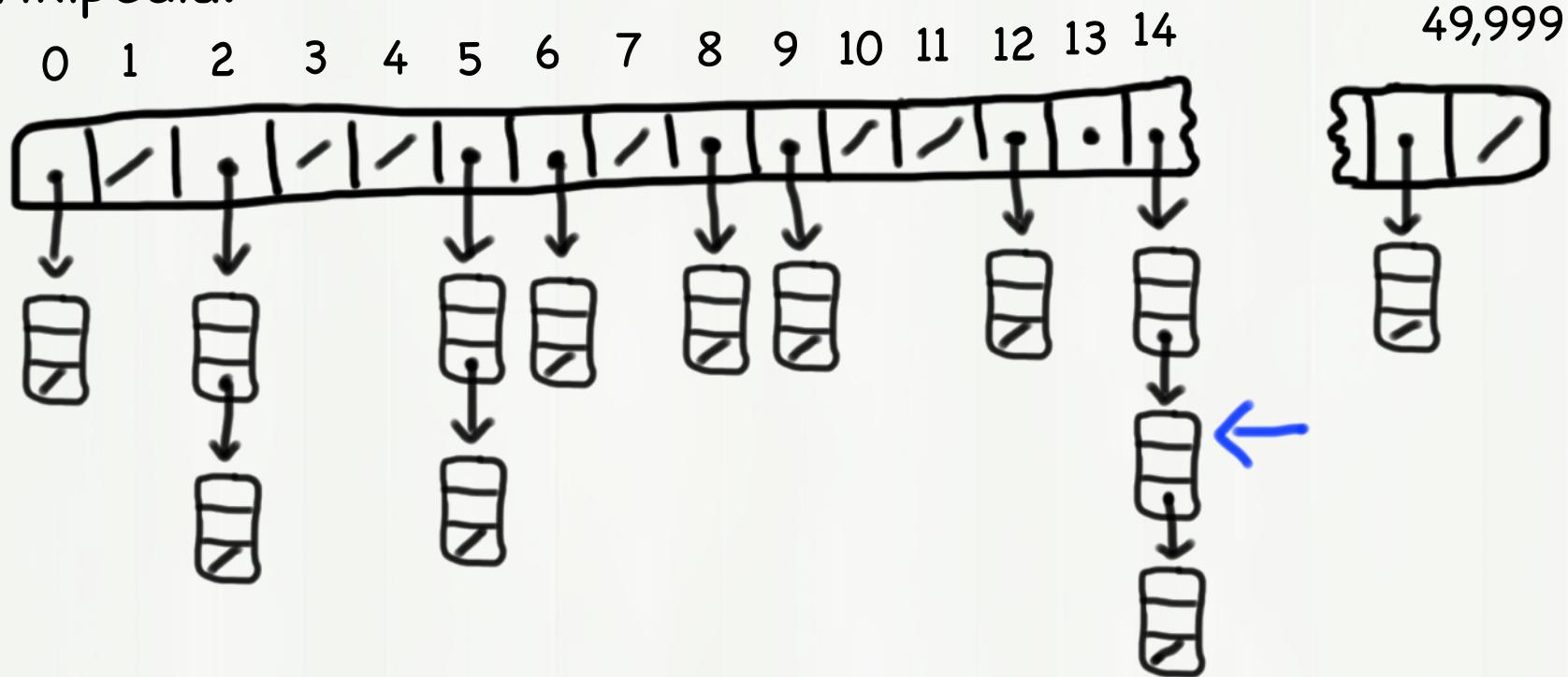


"Antelope
Canyon" → Hash
Fn → 14



get("Antelope Canyon")

Wikipedia:

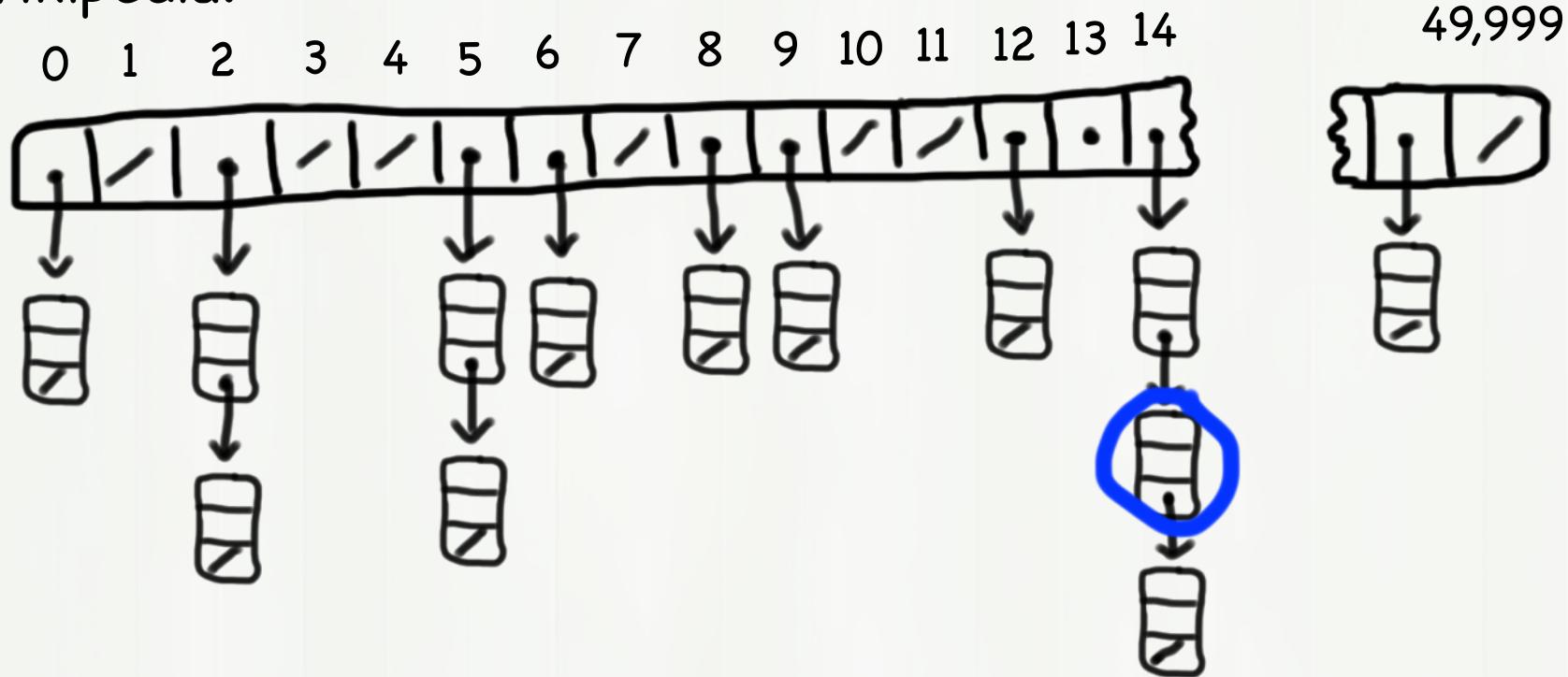


"Antelope
Canyon" → Hash
Fn → 14



get("Antelope Canyon")

Wikipedia:



"Antelope
Canyon" → Hash
Fn → 14

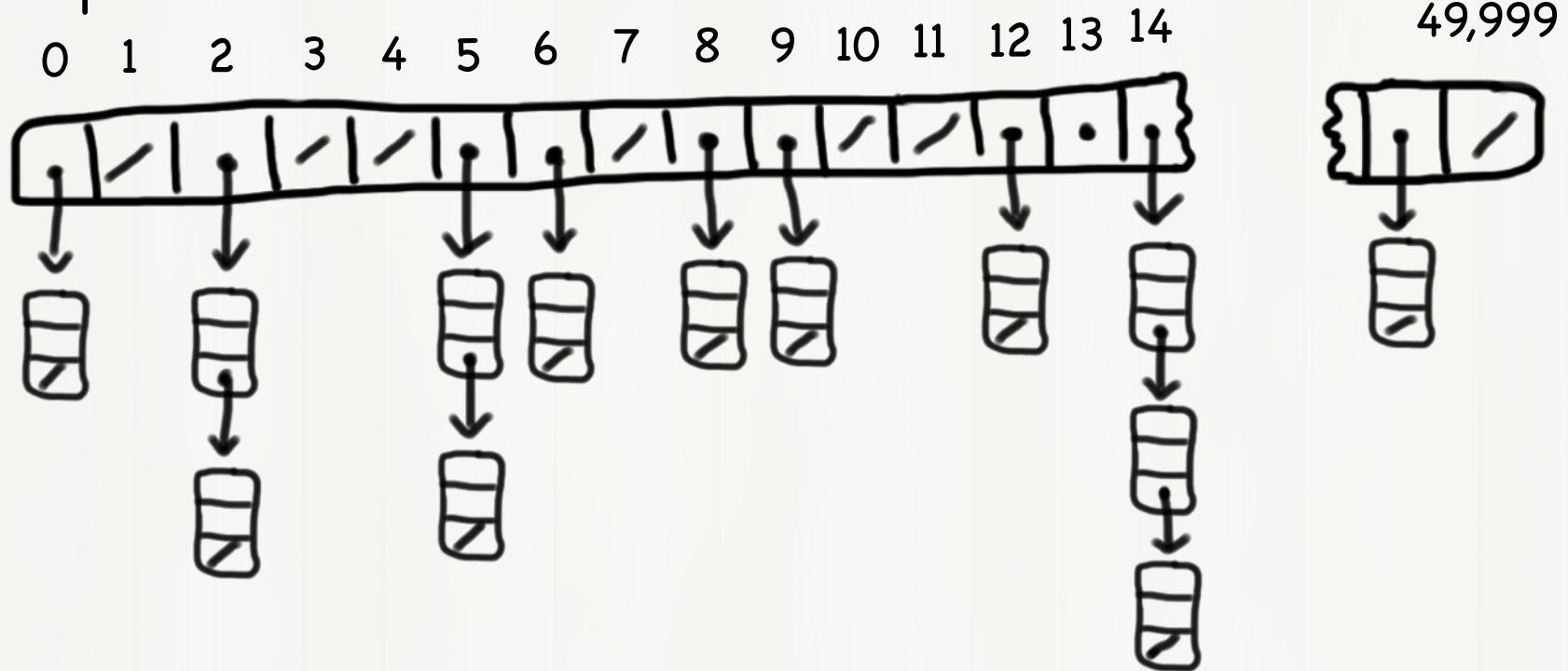




Photo Credit: Alex Mironyuk

HashMap

Wikipedia:

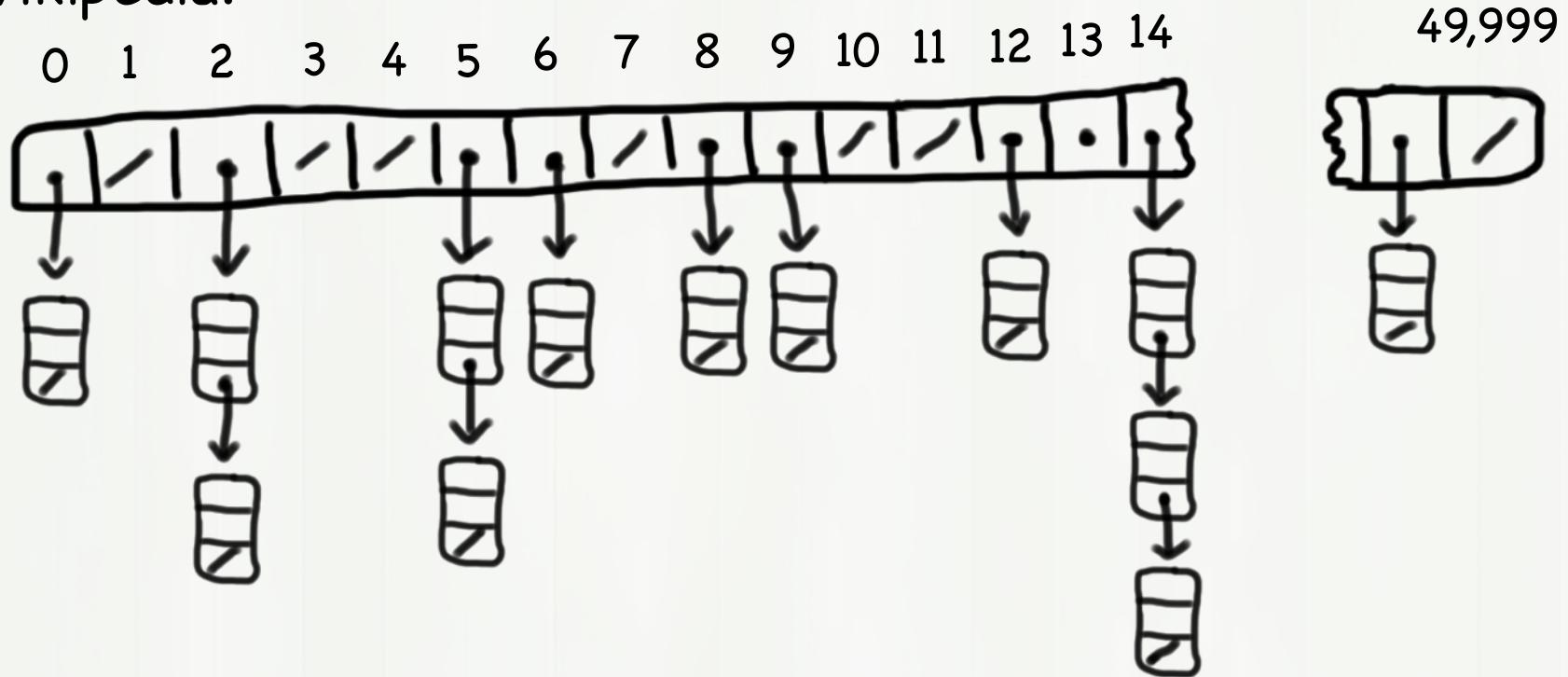


Hash
Fn



put("John Coltrane", html)

Wikipedia:

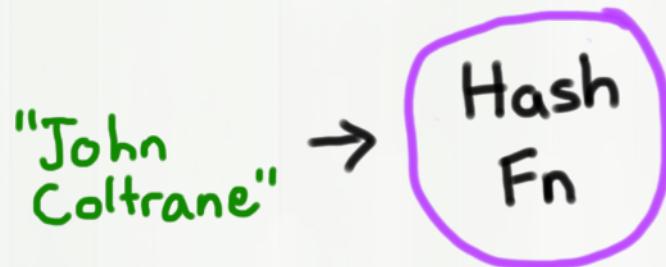
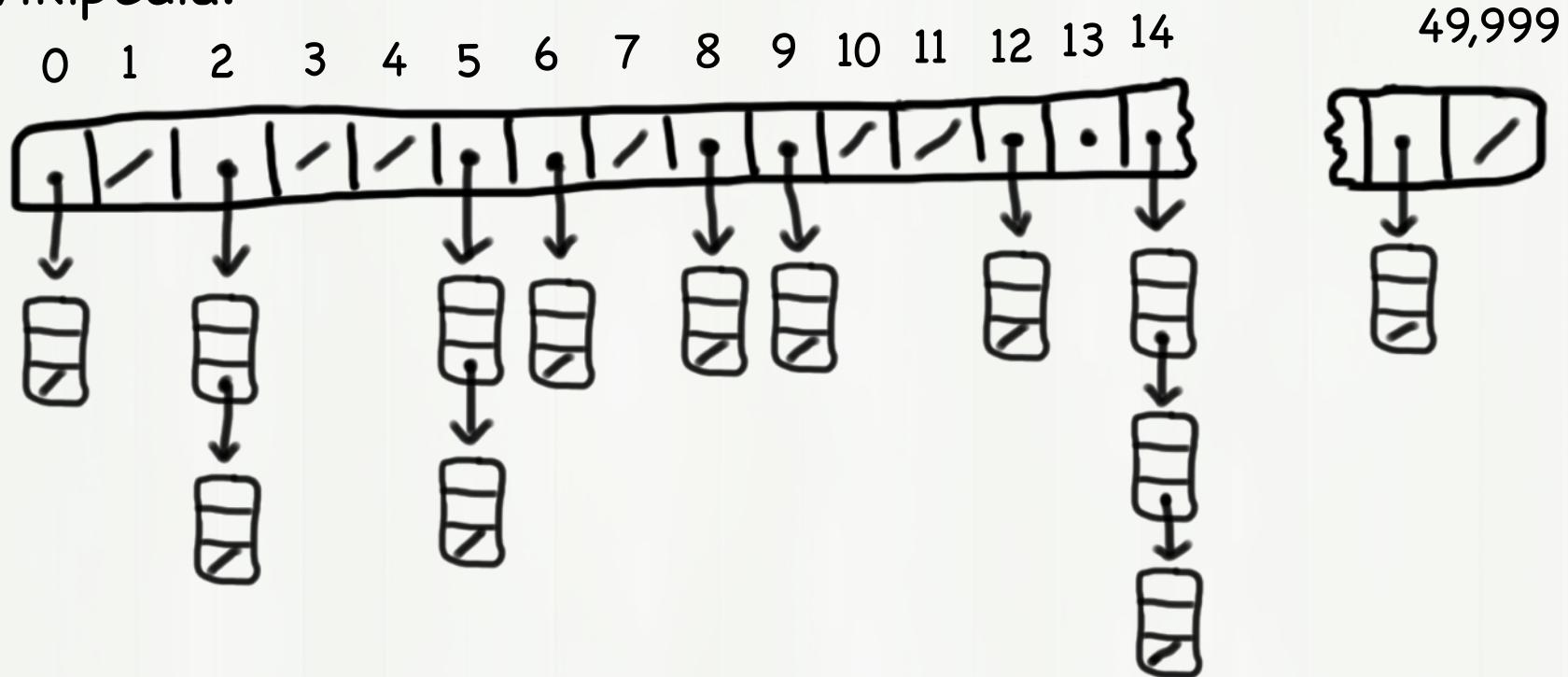


Hash
Fn



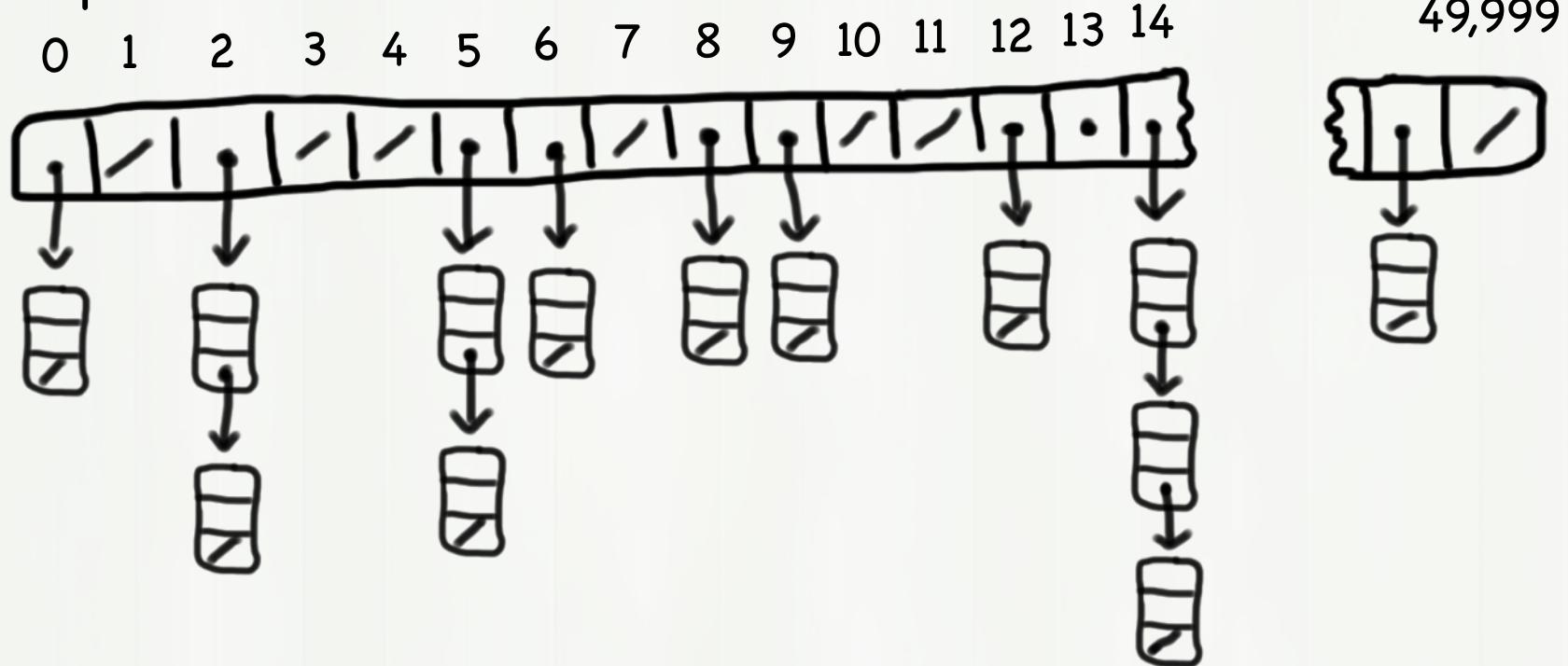
```
put("John Coltrane", html)
```

Wikipedia:



`put("John Coltrane", html)`

Wikipedia:

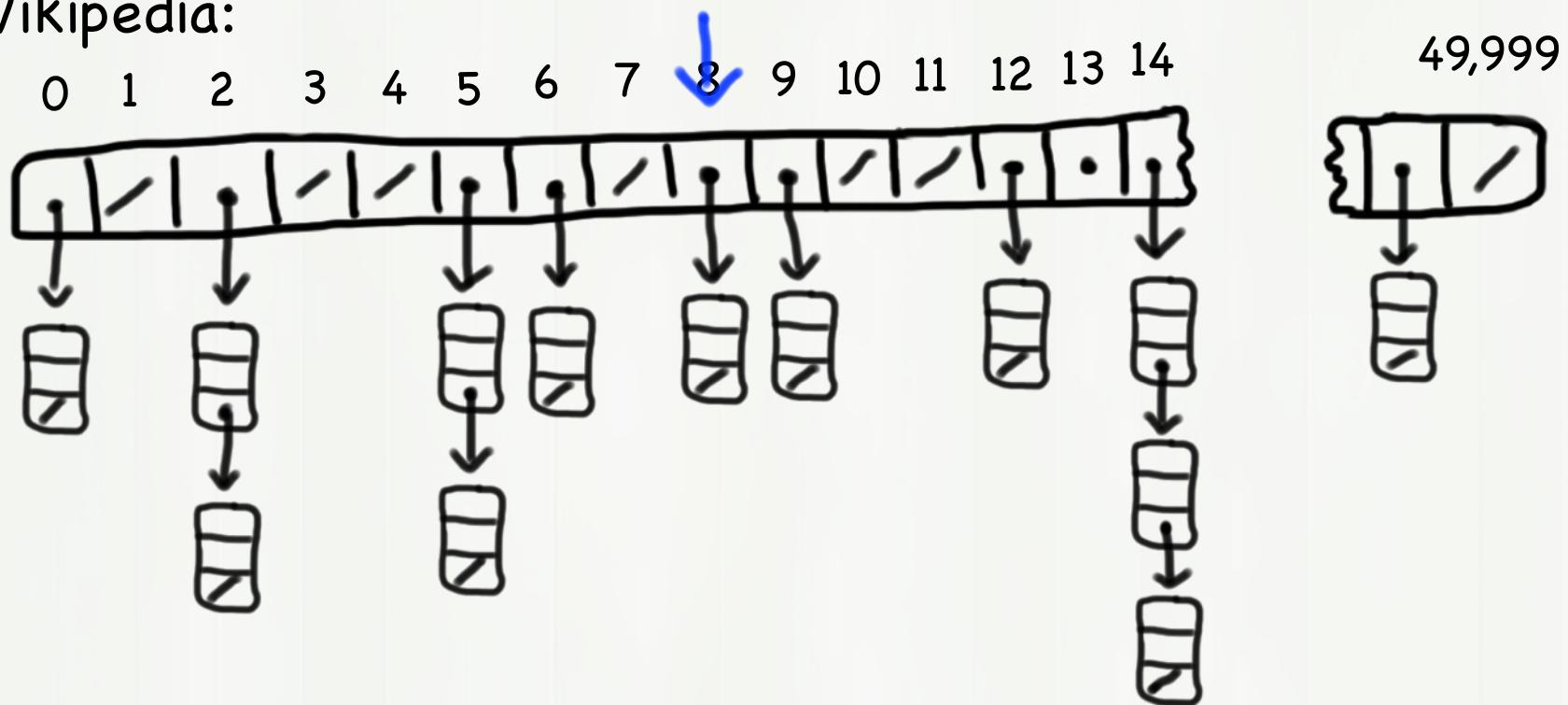


"John Coltrane" → Hash Fn → 8



`put("John Coltrane", html)`

Wikipedia:

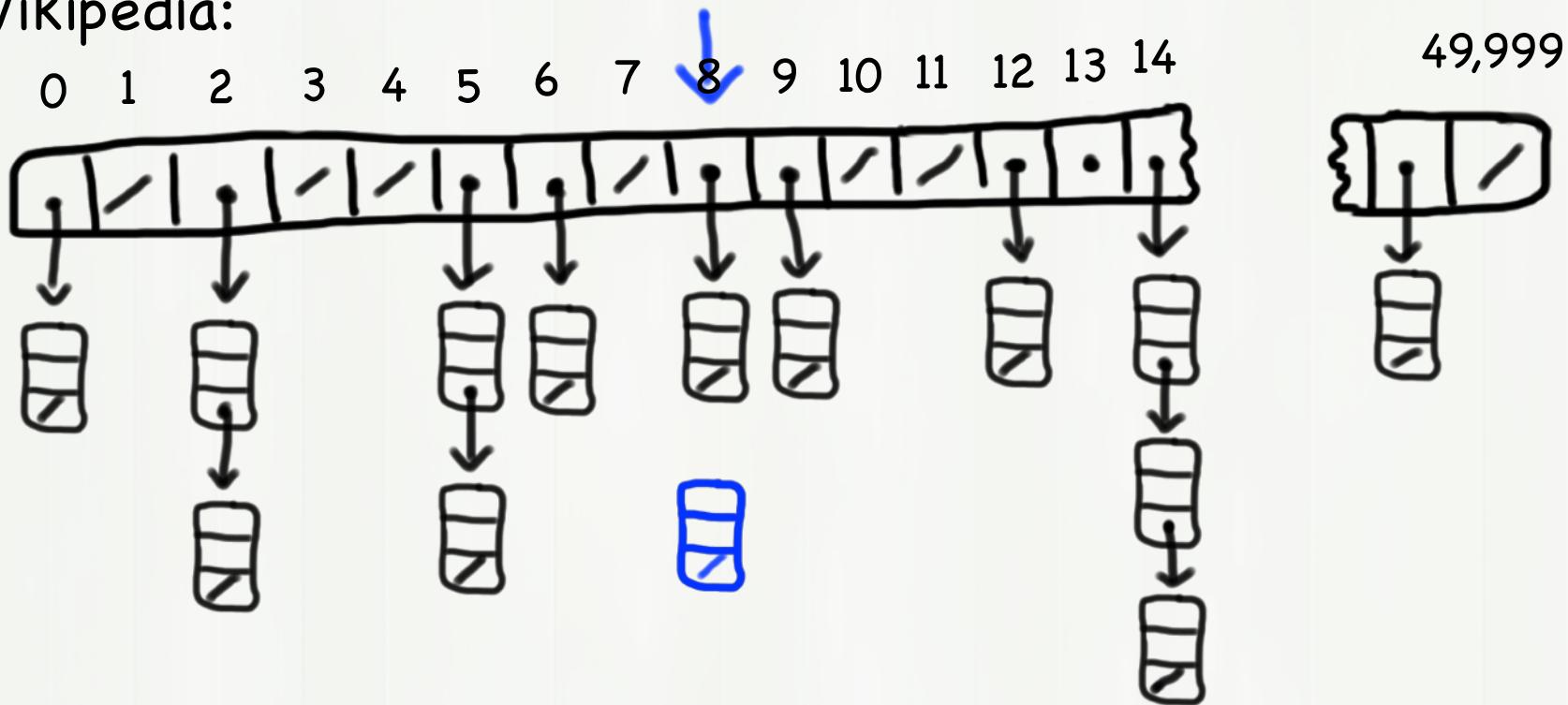


"John
Coltrane" → Hash
Fn → 8



`put("John Coltrane", html)`

Wikipedia:

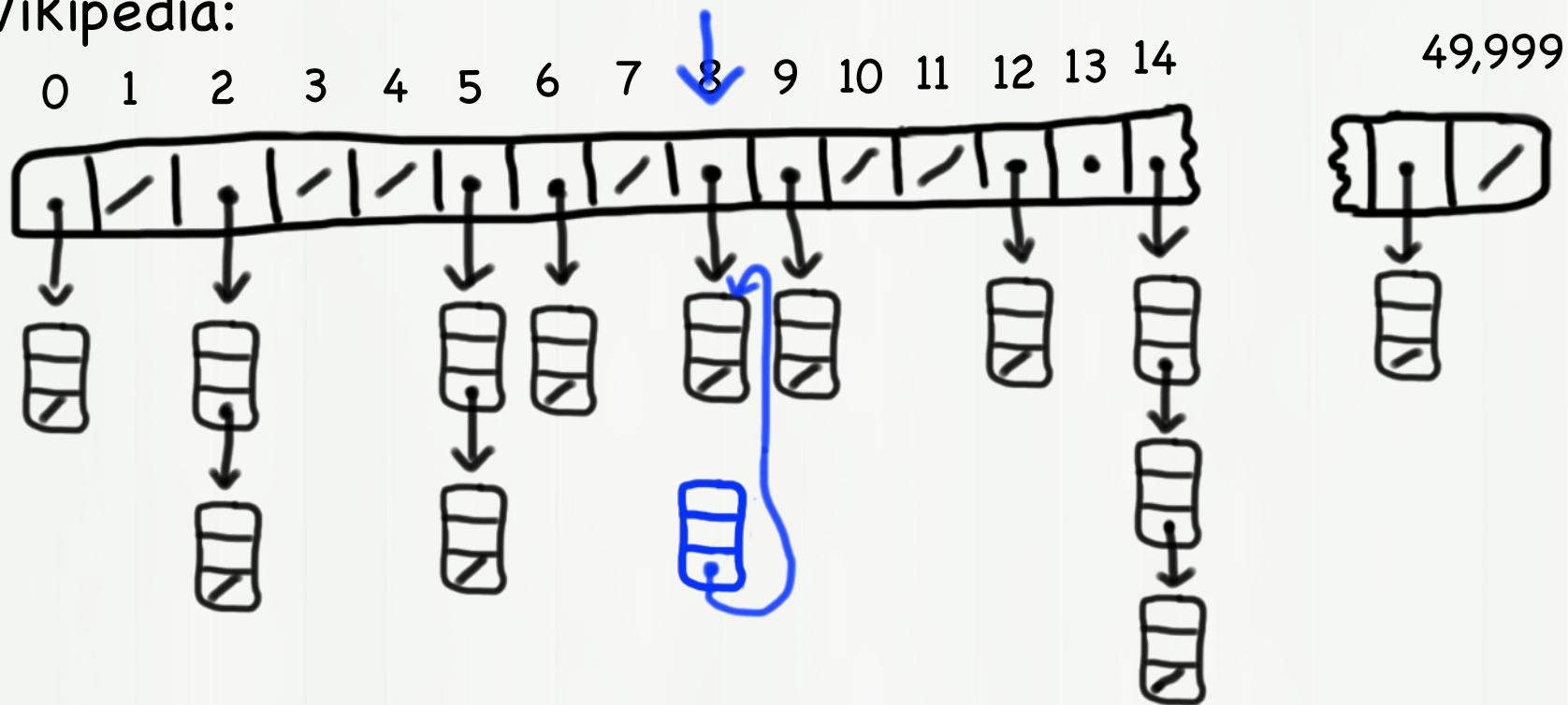


"John Coltrane" → Hash Fn → 8



`put("John Coltrane", html)`

Wikipedia:

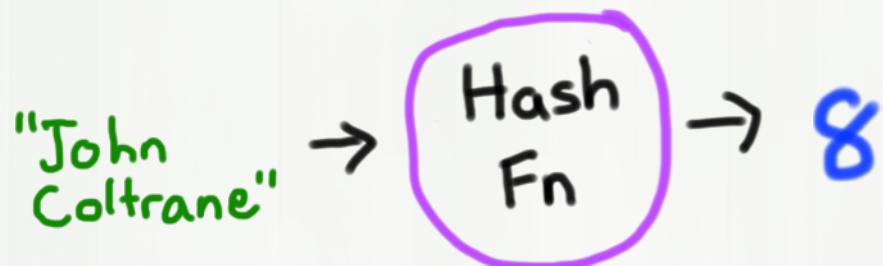
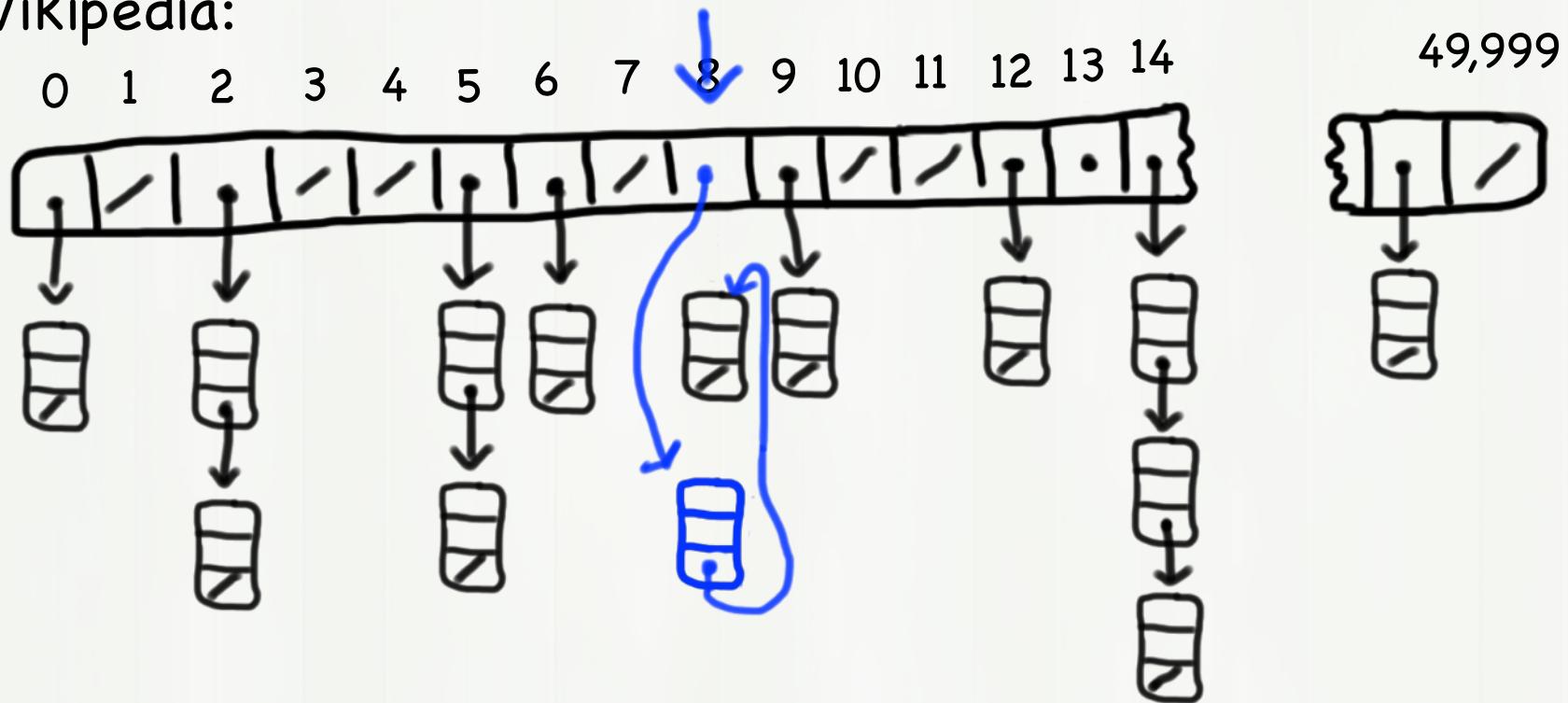


"John
Coltrane" → Hash
Fn → 8



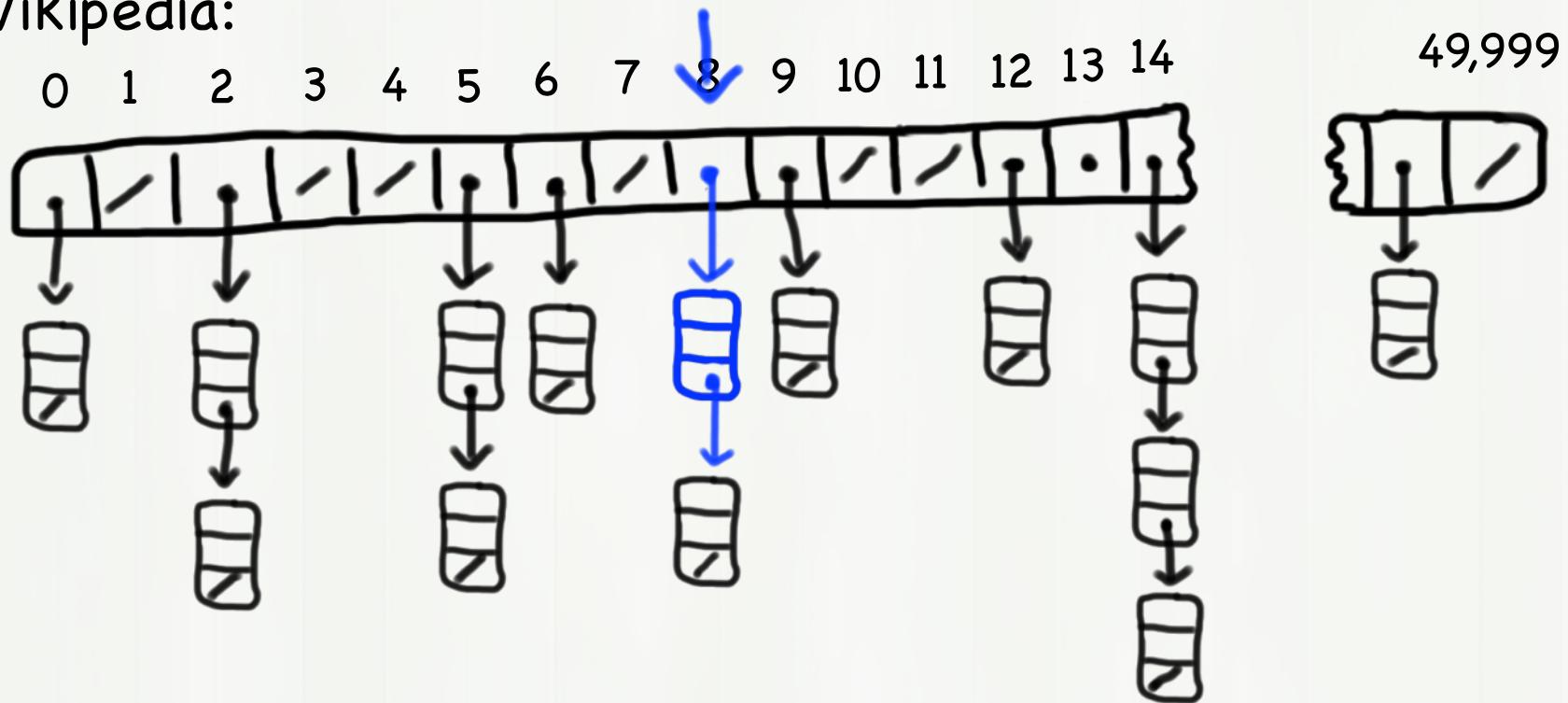
`put("John Coltrane", html)`

Wikipedia:



`put("John Coltrane", html)`

Wikipedia:



"John Coltrane" → Hash Fn → 8



HashMap Algorithm Summary

Initialization:

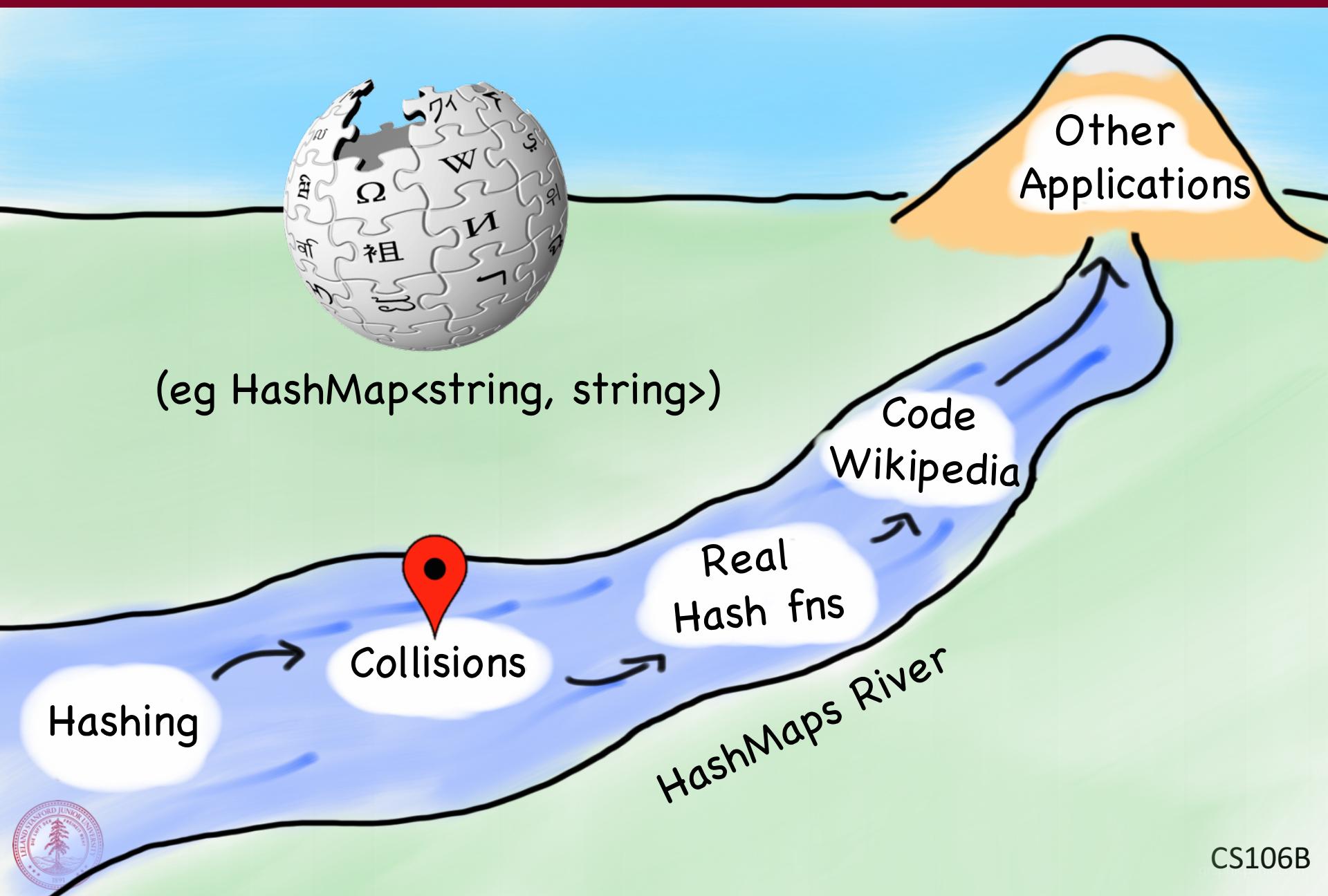
0. Make a large array of linked lists

Getting and Setting:

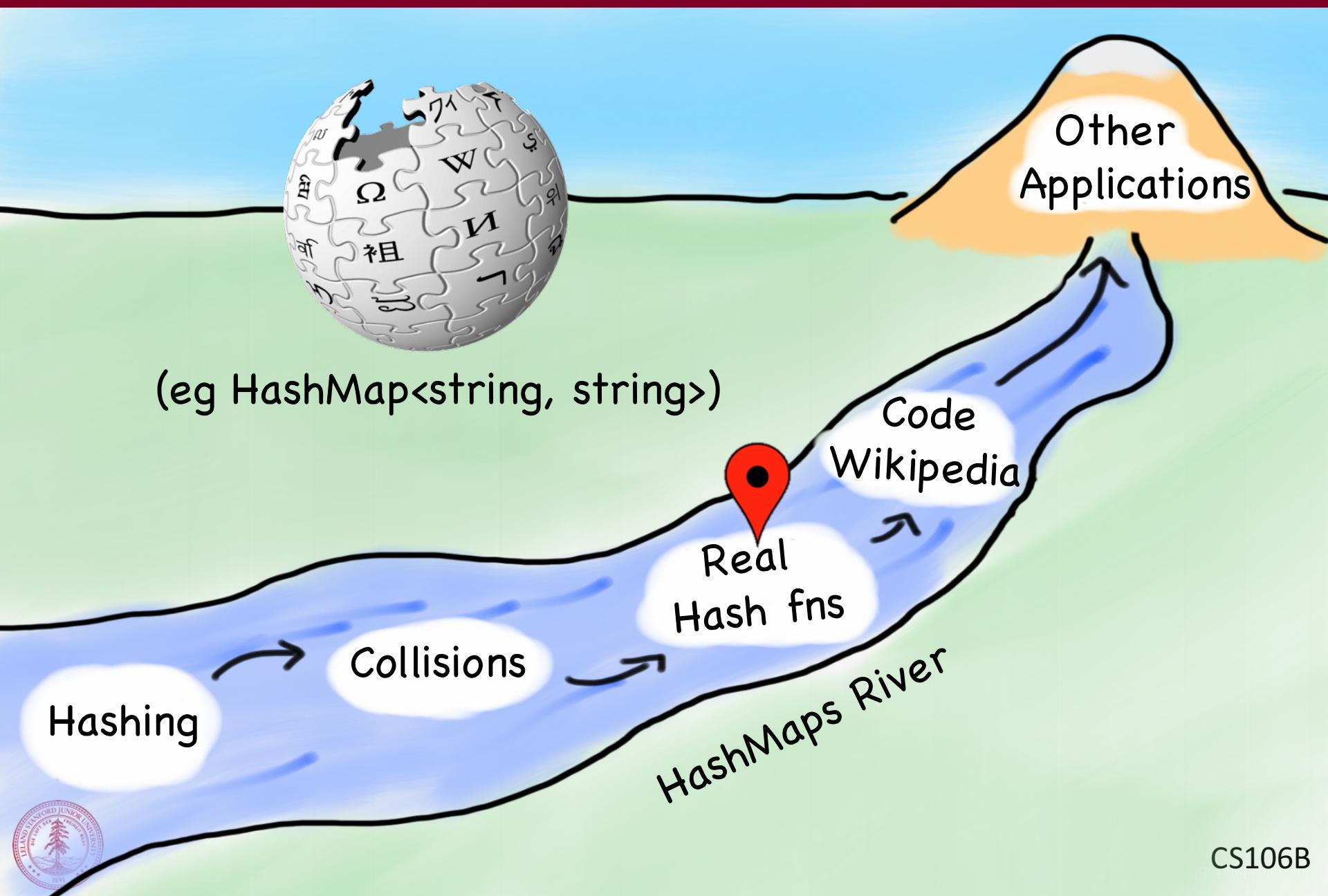
1. Hash: key \rightarrow bucketIndex.
2. Jump directly to the bucket head.
3. Run linear search over the short list.



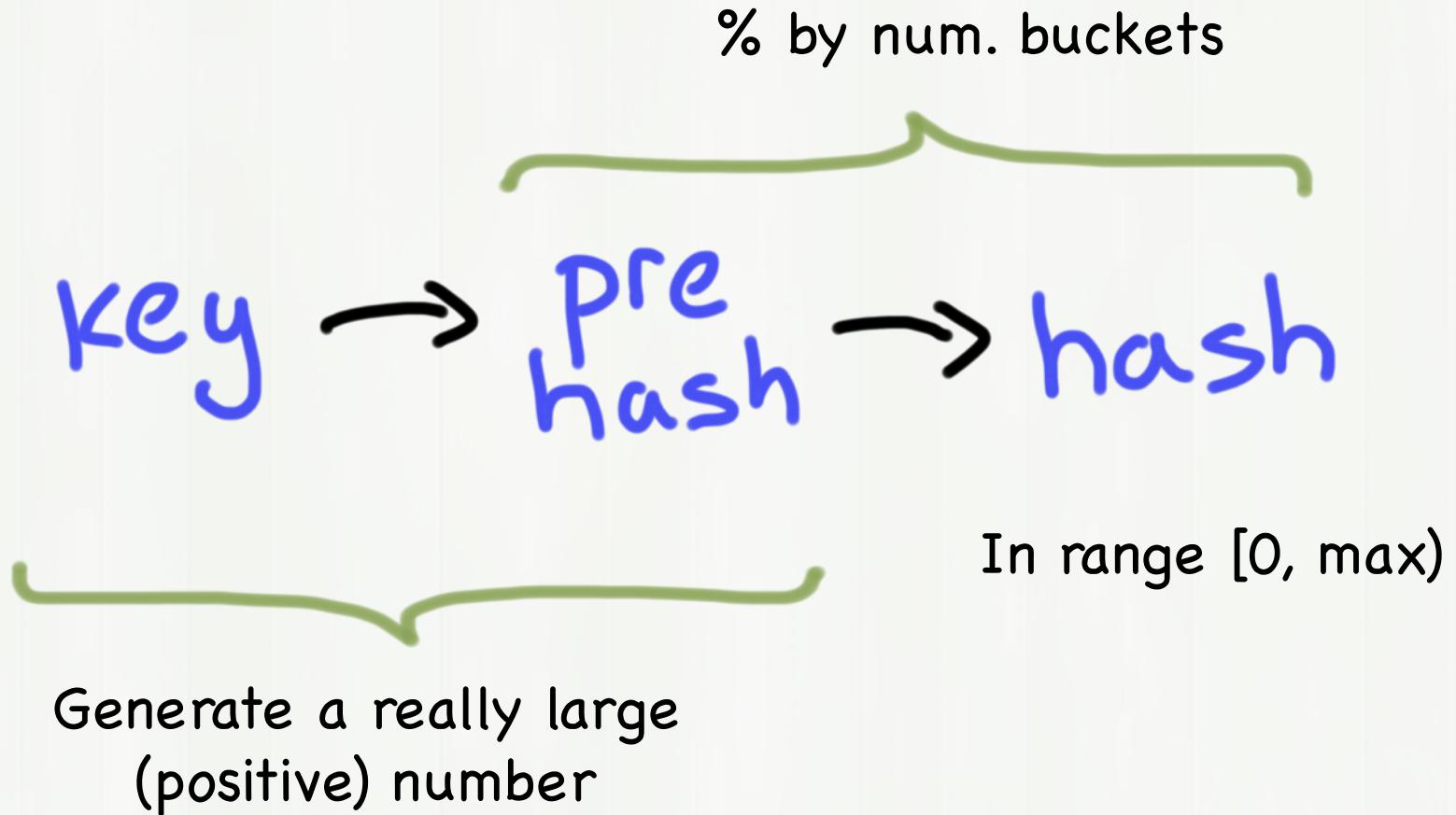
Today's Route



Today's Route



Hash Function Template



Simple String Hash:

preHash = The sum of all of the
characters



A Basic Hash Function

Given string a as a key.

```
prehash = a[0] + a[1] + ... + a[n - 1]
```

```
return (prehash % numBuckets)
```



A Basic Hash Function

```
int basicHash(string key) {  
    int preHash = 0;  
    for(int i = 0; i < key.length() i++){  
        preHash += key[i];  
    }  
    return preHash % NUM_BUCKETS;  
}
```

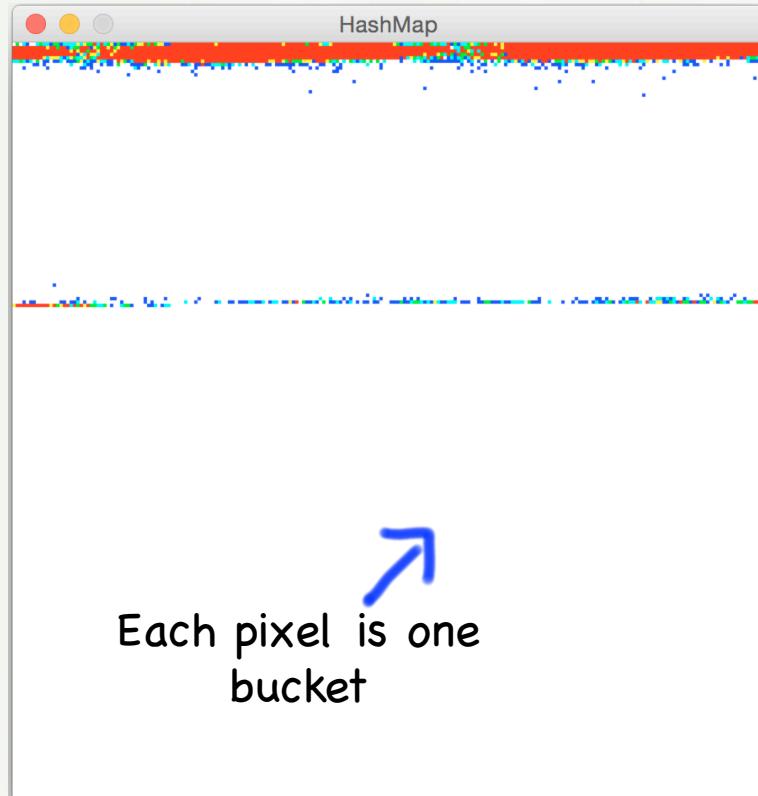


Experiment: I hashed 50 thousand wikipedia articles into 50 thousand buckets and looked at the number of collisions in each bucket.

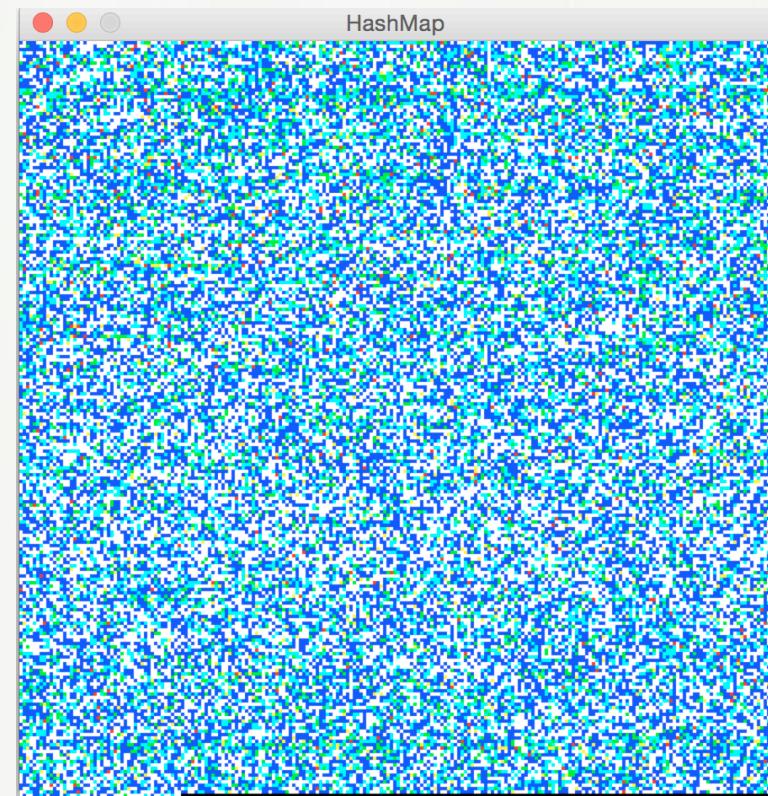


Judge Strings by the Contents of their Characters

preHash = Add each character in a string



preHash = Add each character in a string, weighted by 31^i



0



2



4



1



3



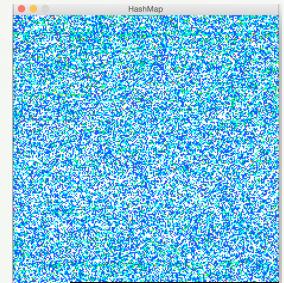
≥ 5

Number of collisions



The C++ Hash

Given string a as a key.



```
prehash =
```

$$a[0] + 31 \cdot a[1] + 31^2 \cdot a[2] + \cdots + 31^n \cdot a[n]$$

```
return (prehash % numBuckets)
```

* In C++, the weights are in reverse order. That is not important :).

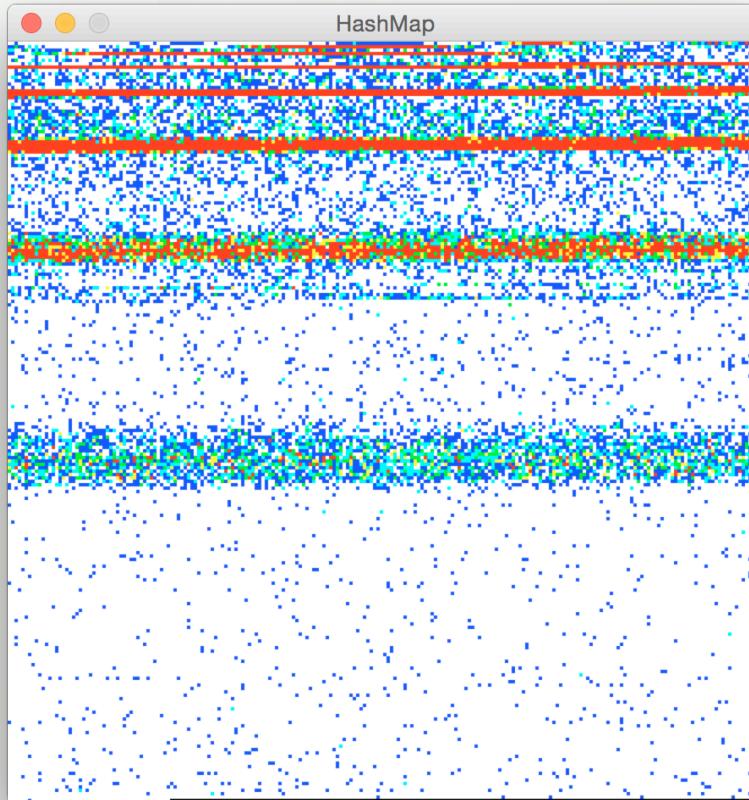


Why 31?



What if the constant was 2?

$$a[0] + 2 \cdot a[1] + 2^2 \cdot a[2] + \cdots + 2^n \cdot a[n]$$



Bucket distribution of 50,000 article titles
from Wikipedia



Question:

What is the big-O of
the **get** method?



[suspense]

Answer:



* In expectation

CS106B



What if nBuckets is too small?!?

HashMap Big-O





Knead to Program It



Source: timescity

Gopher It



Source: Planet Animal Zone

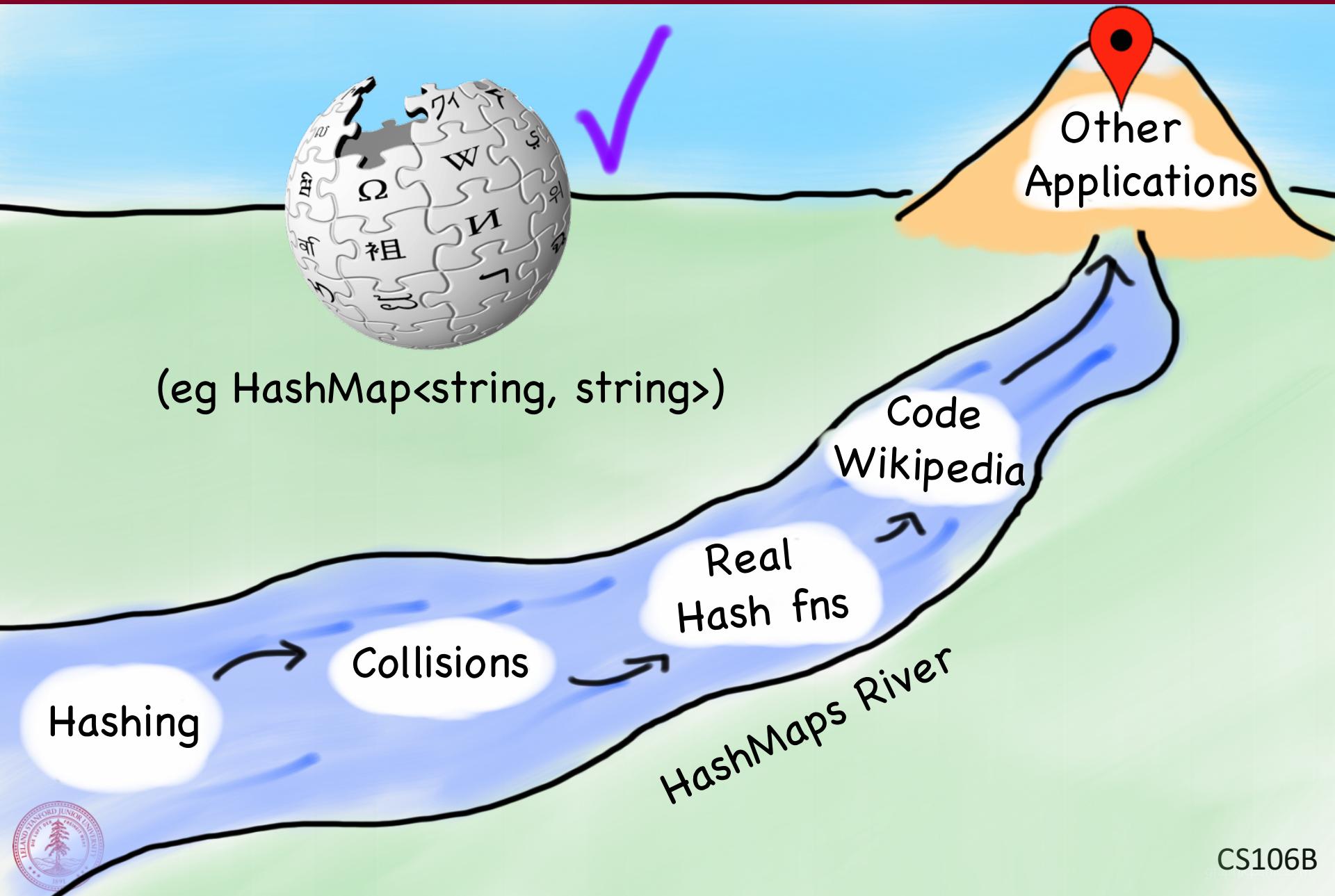
Today's Route



Today's Route



Today's Route



Many Applications of Hashing



Jeff Jarvis @jeffjarvis

6 Nov

Hey @Snowden, for context, how long would it take the NSA to dedupe 650k emails?



Edward Snowden

@Snowden

Follow

@jeffjarvis Drop non-responsive To:/CC:/BCC: hash both sets, then subtract those that match. Old laptops could do it in minutes-to-hours.

5:19 PM - 6 Nov 2016

The first laptop ever made...



3,789

5,166



And Here We Are



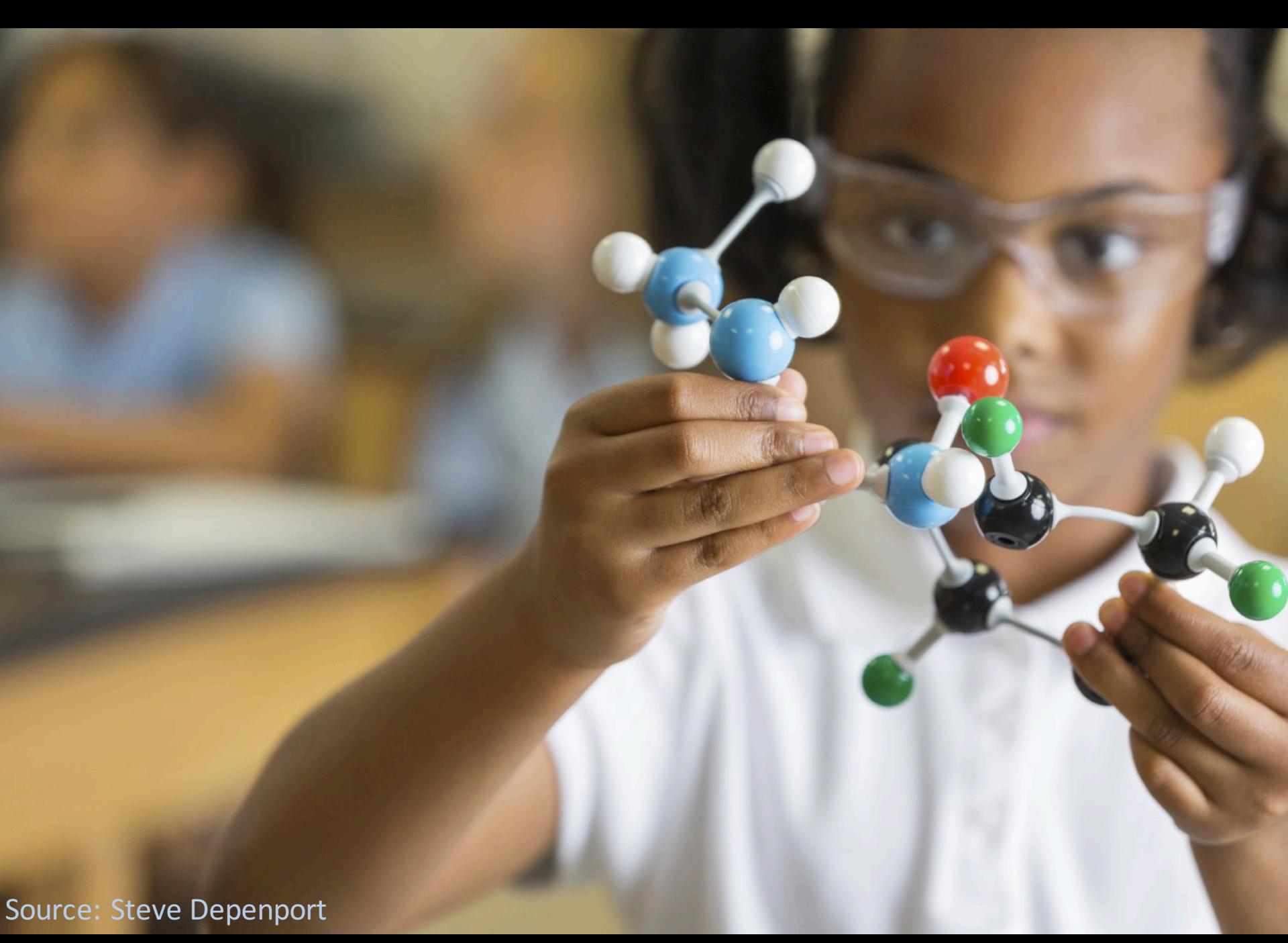
Shazam



Source: Shazam

CS106B



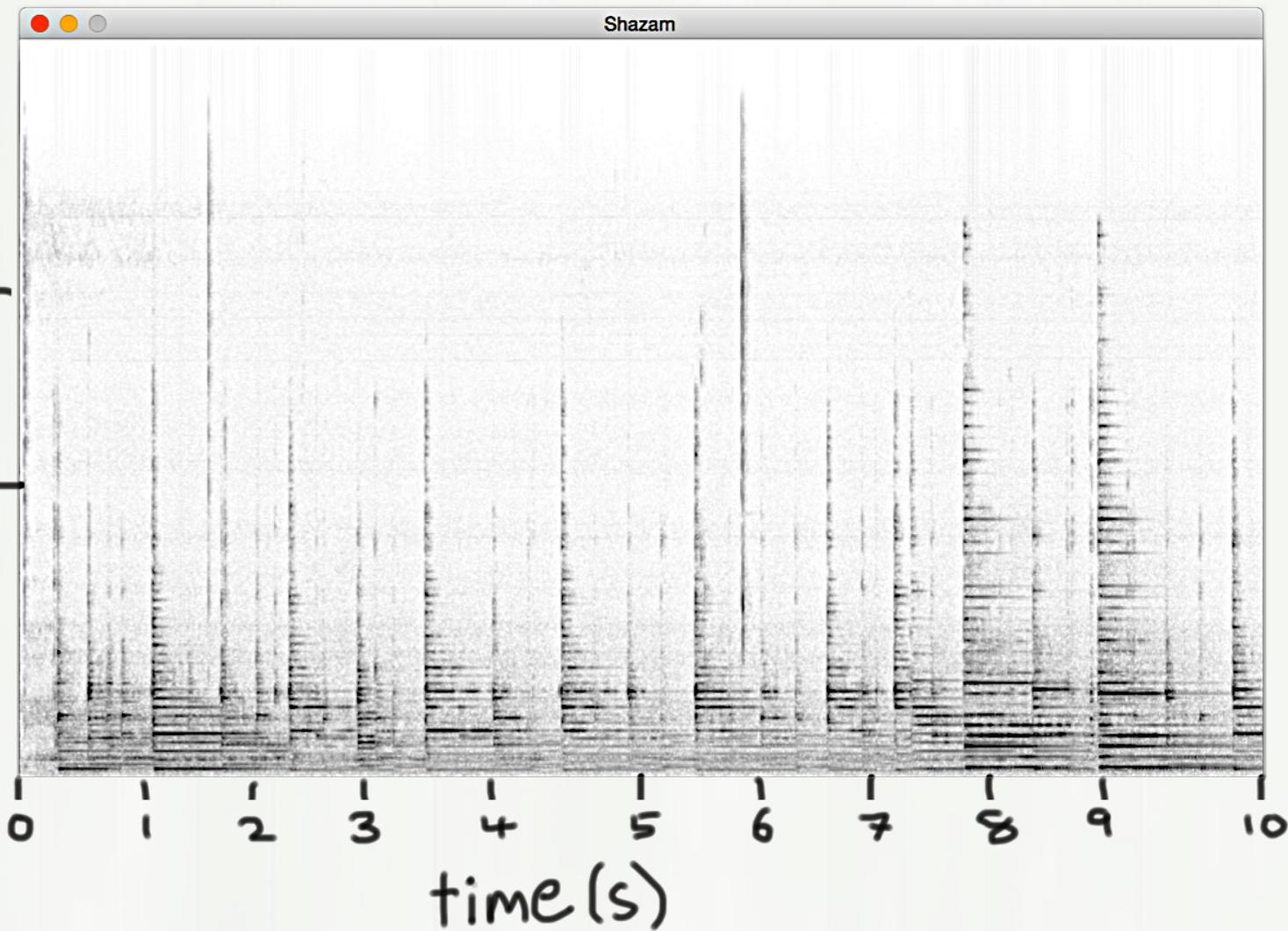


Source: Steve Depenport

Similar to Wikipedia:
Huge search problem.

Spectrogram

frequency (Hz)



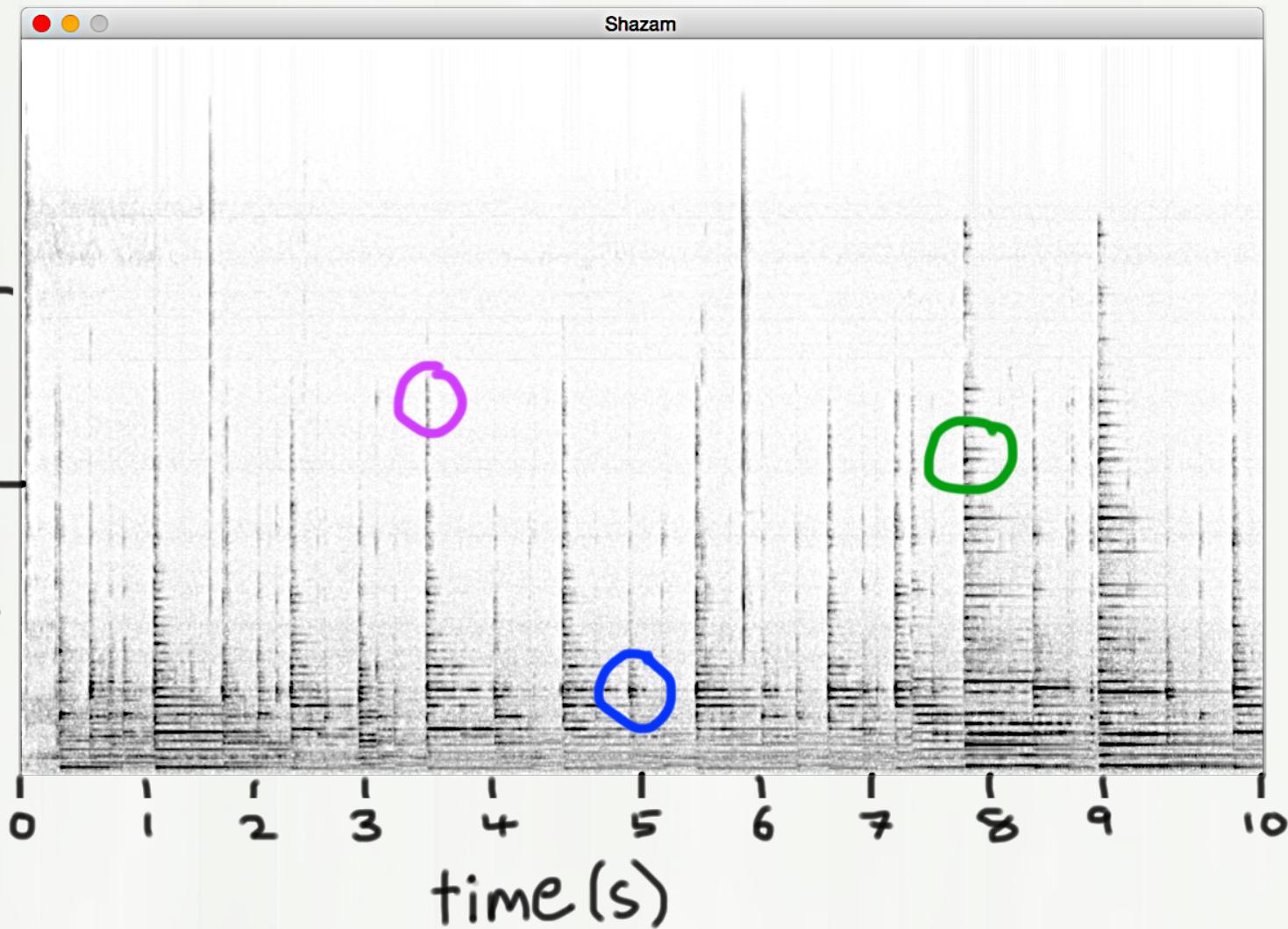
Does anyone recognize this song?



CS106B

Spectrogram

frequency (Hz)

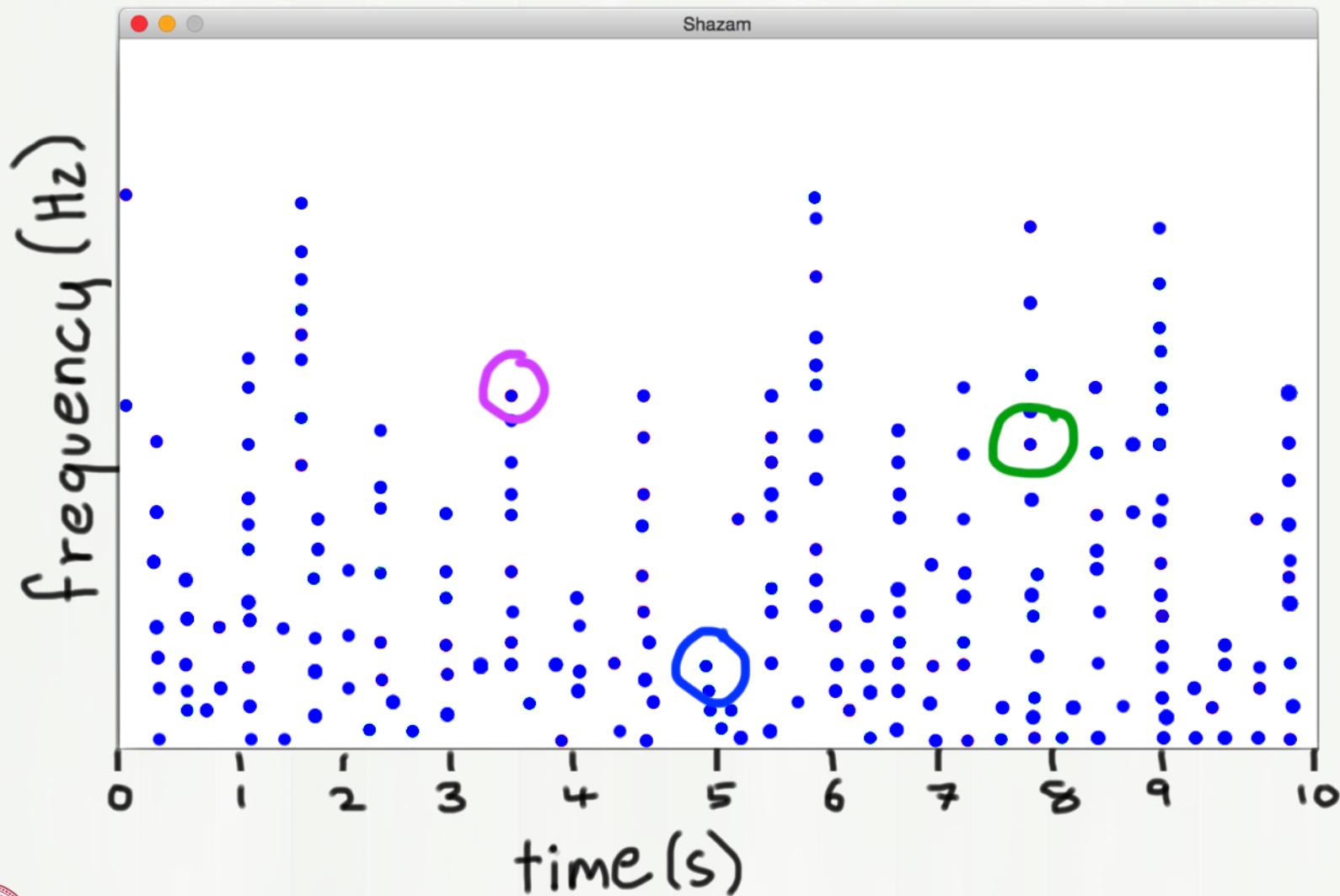


Does anyone recognize this song?



CS106B

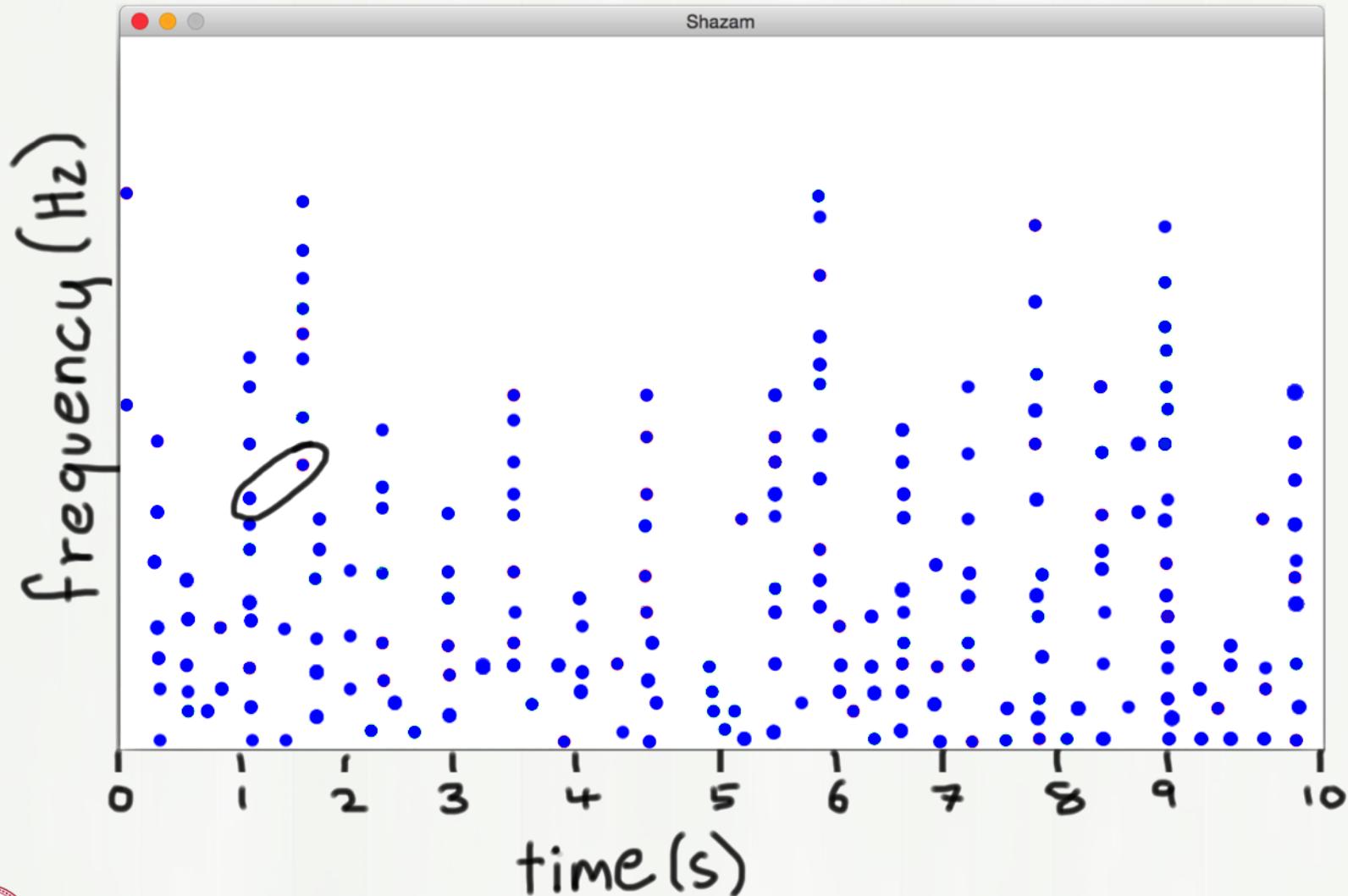
Notes Over Time



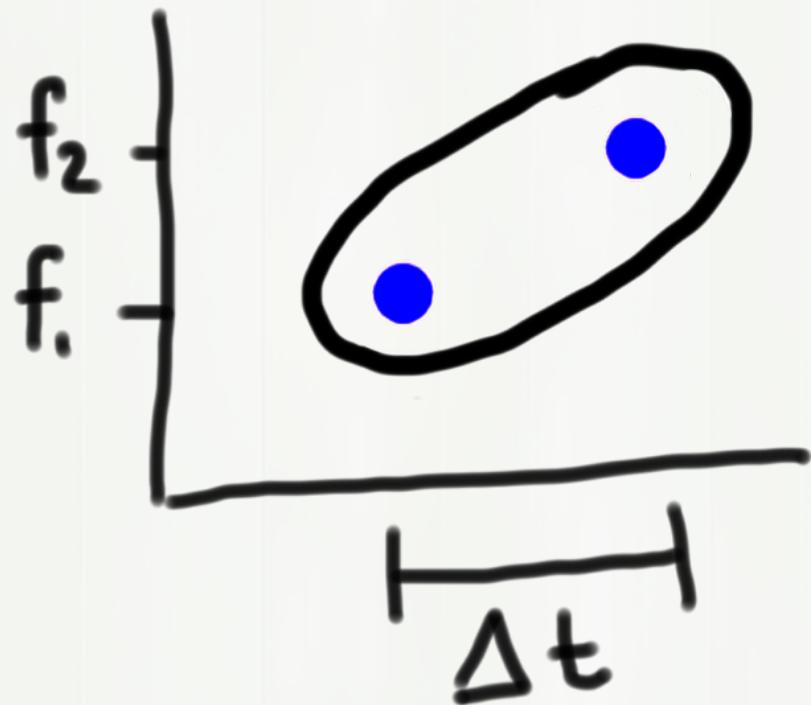
Hash the whole thing?

No

Find Pairs of Notes



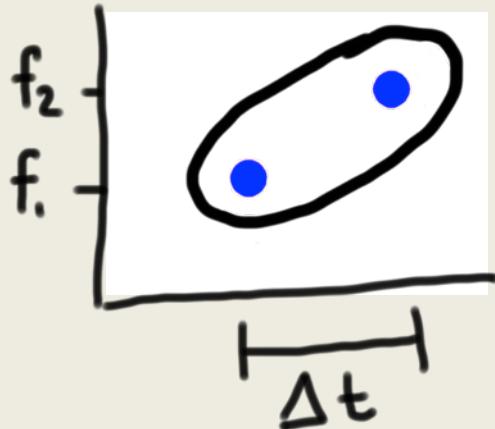
Note Pairs



Note/Songs Pairs

Shazam

Key :



Value :

You Can Call Me Al – Paul Simon. 7s,
You Can Call Me Al – Paul Simon. 43s,
All Right Now – Police. 18s

Wikipedia

Key :

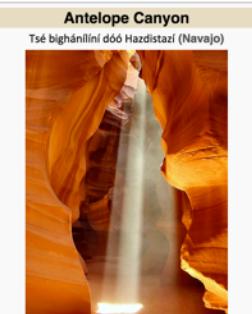
“Antelope Canyon”

value :

Antelope Canyon is a slot canyon in the American Southwest. It is located on Navajo land east of Page, Arizona. Antelope Canyon includes two separate, photogenic slot canyon sections, referred to individually as *Upper Antelope Canyon* or *The Crack*; and *Antelope Canyon* or *The Corkscrew*.^[2]

The Navajo name for Upper Antelope Canyon is Tsé bighánílíní, which means "the place where water runs through rocks." Lower Antelope Canyon is Hazdistazi (advertised as "Hasdestwazi" by the Navajo Parks and Recreation Department), or "spiral rock arches." Both are located within the LeChee Chapter of the Navajo Nation.^[4]

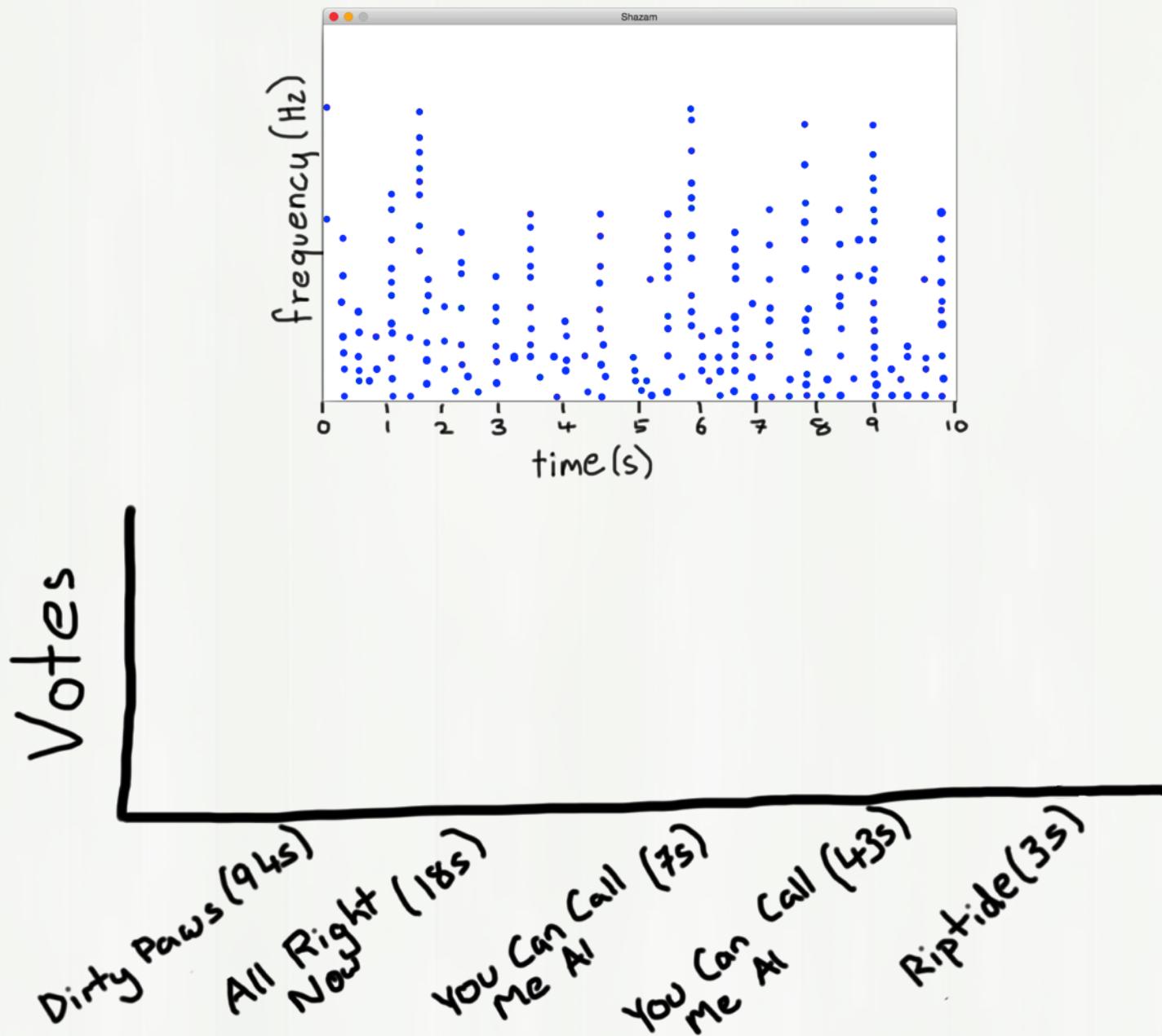
Contents [hide]
1 Geology
2 Tourism and photography
2.1 Upper Antelope Canyon



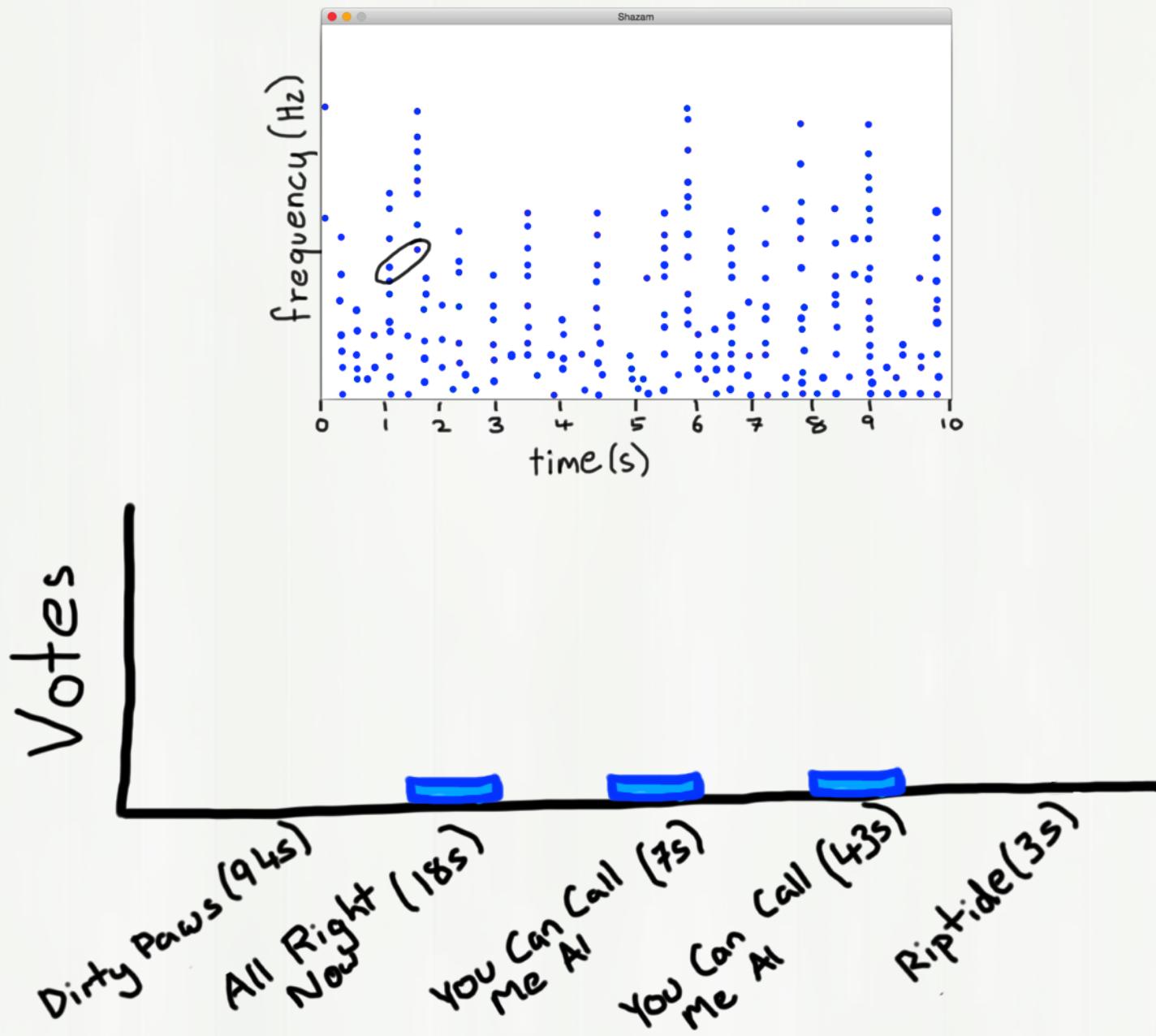
A beam of light in Upper Antelope Canyon



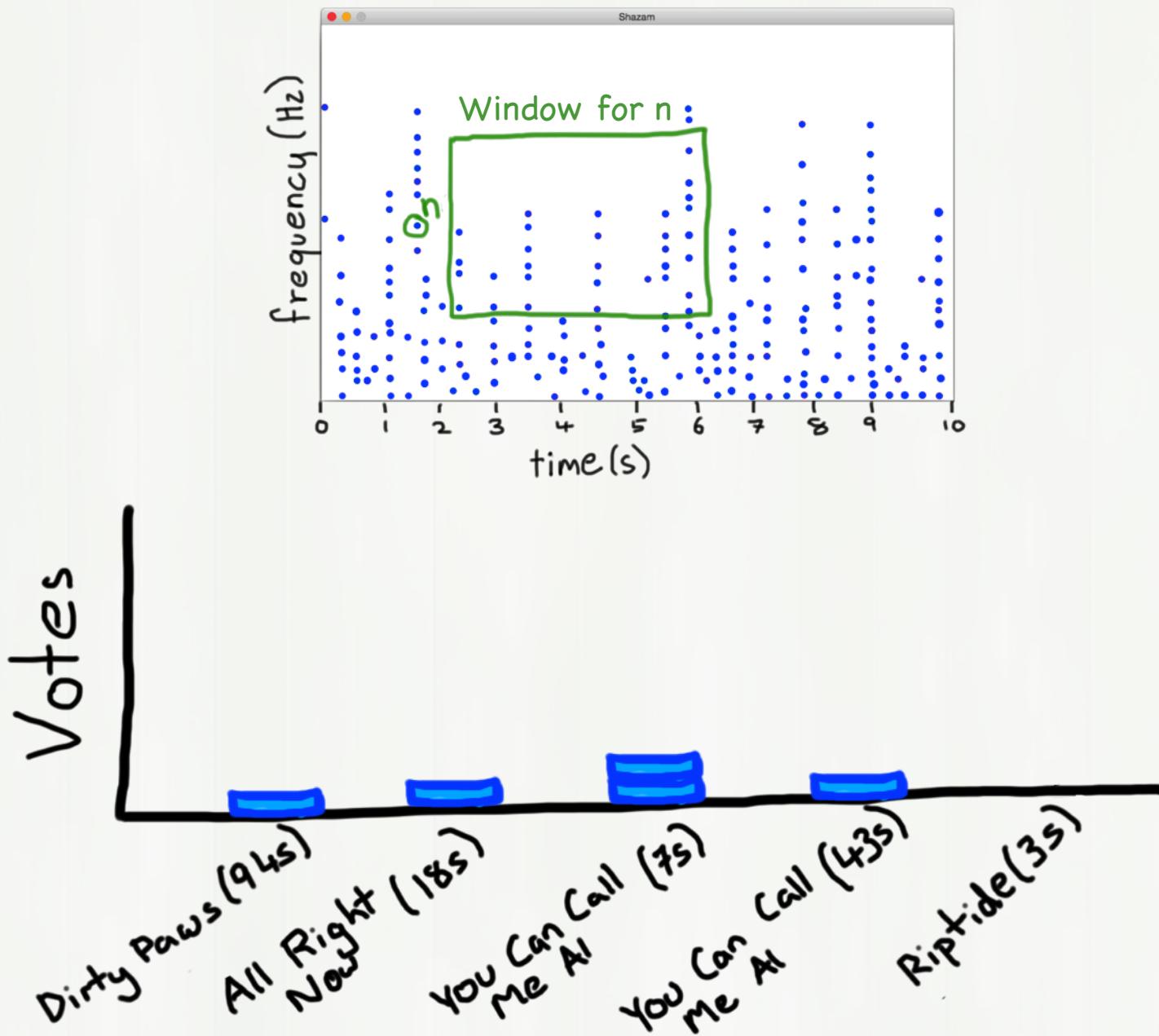
Note Pair Hashing



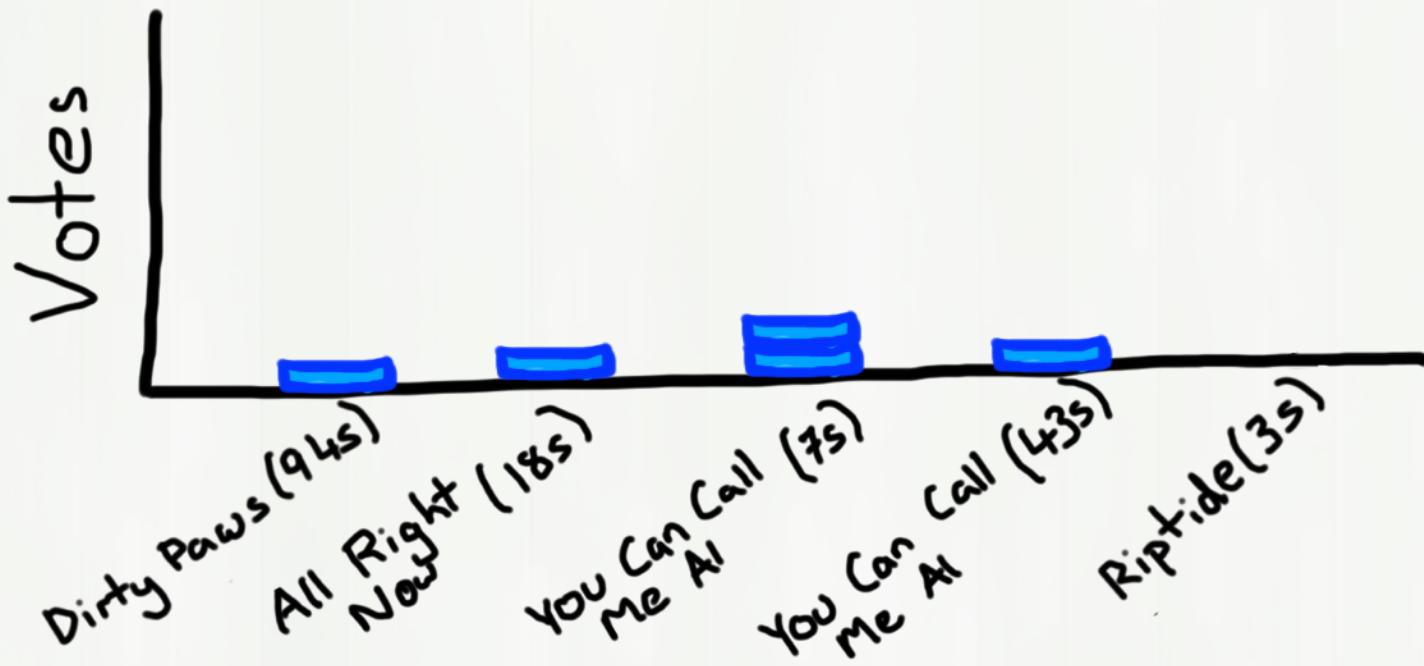
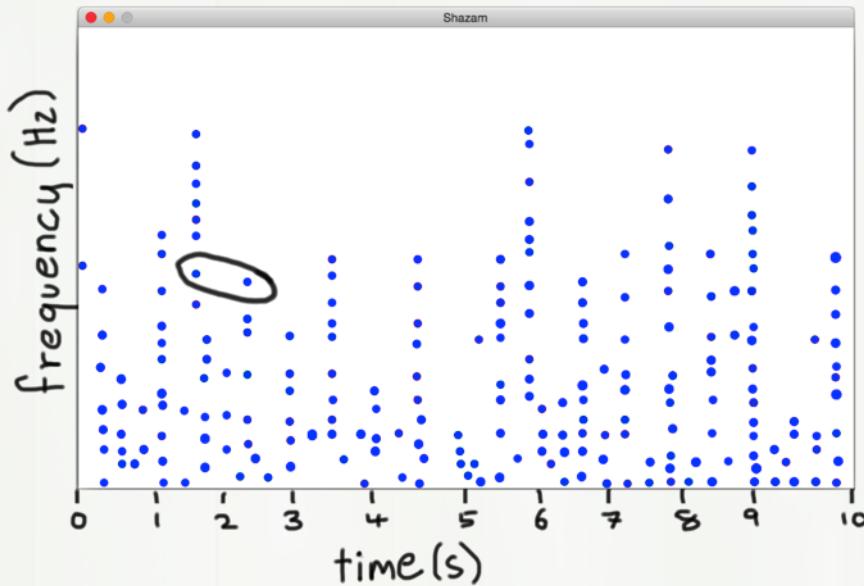
Note Pair Hashing



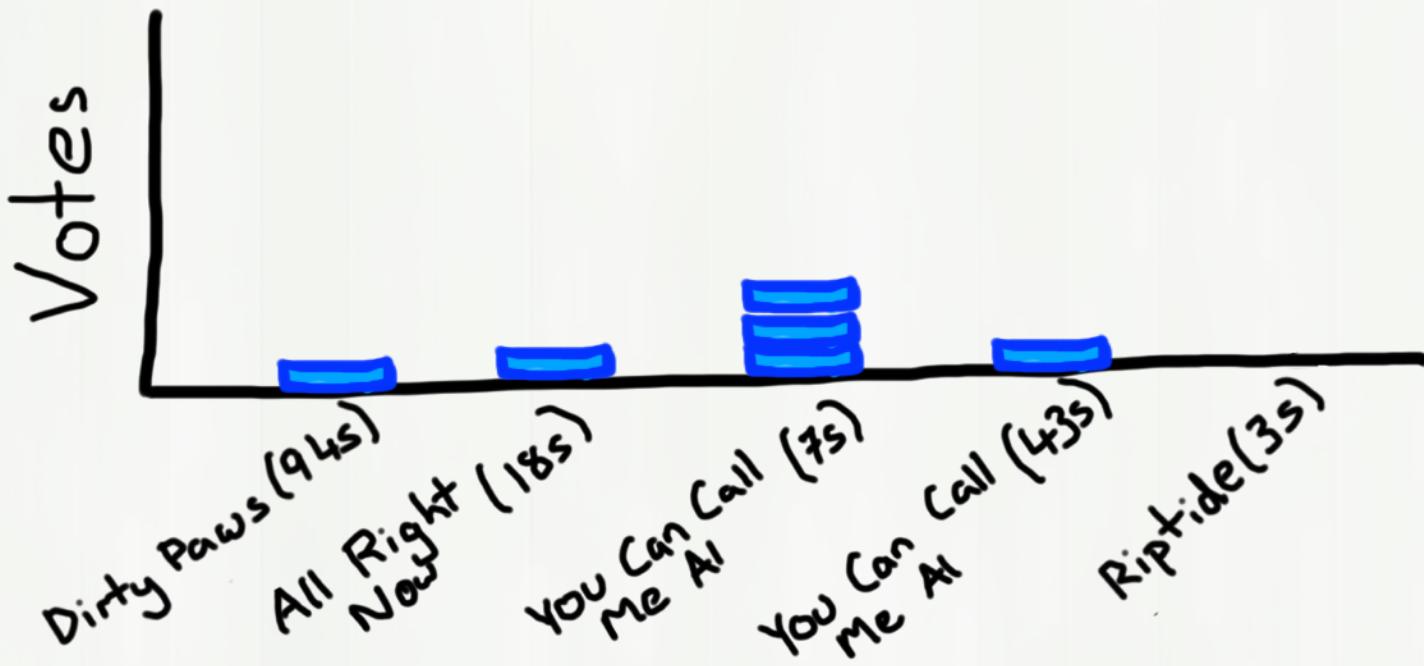
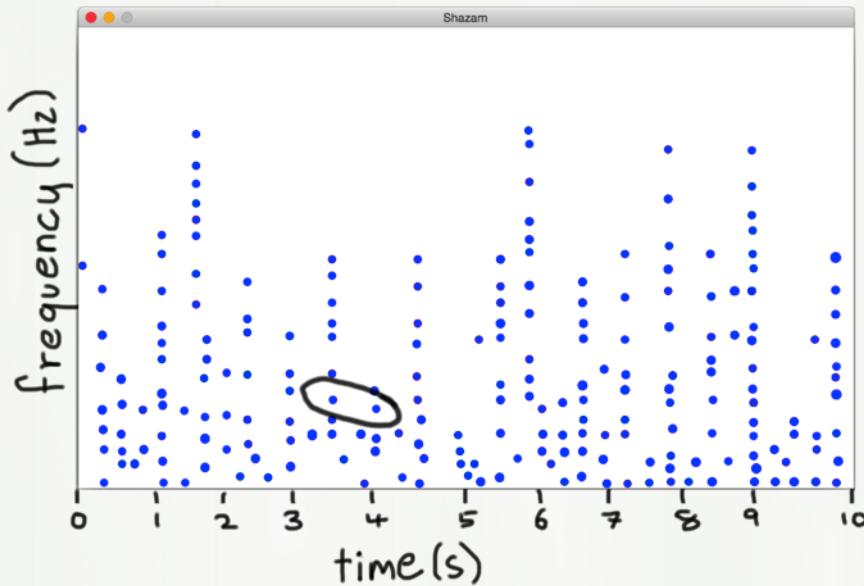
Note Pair Hashing



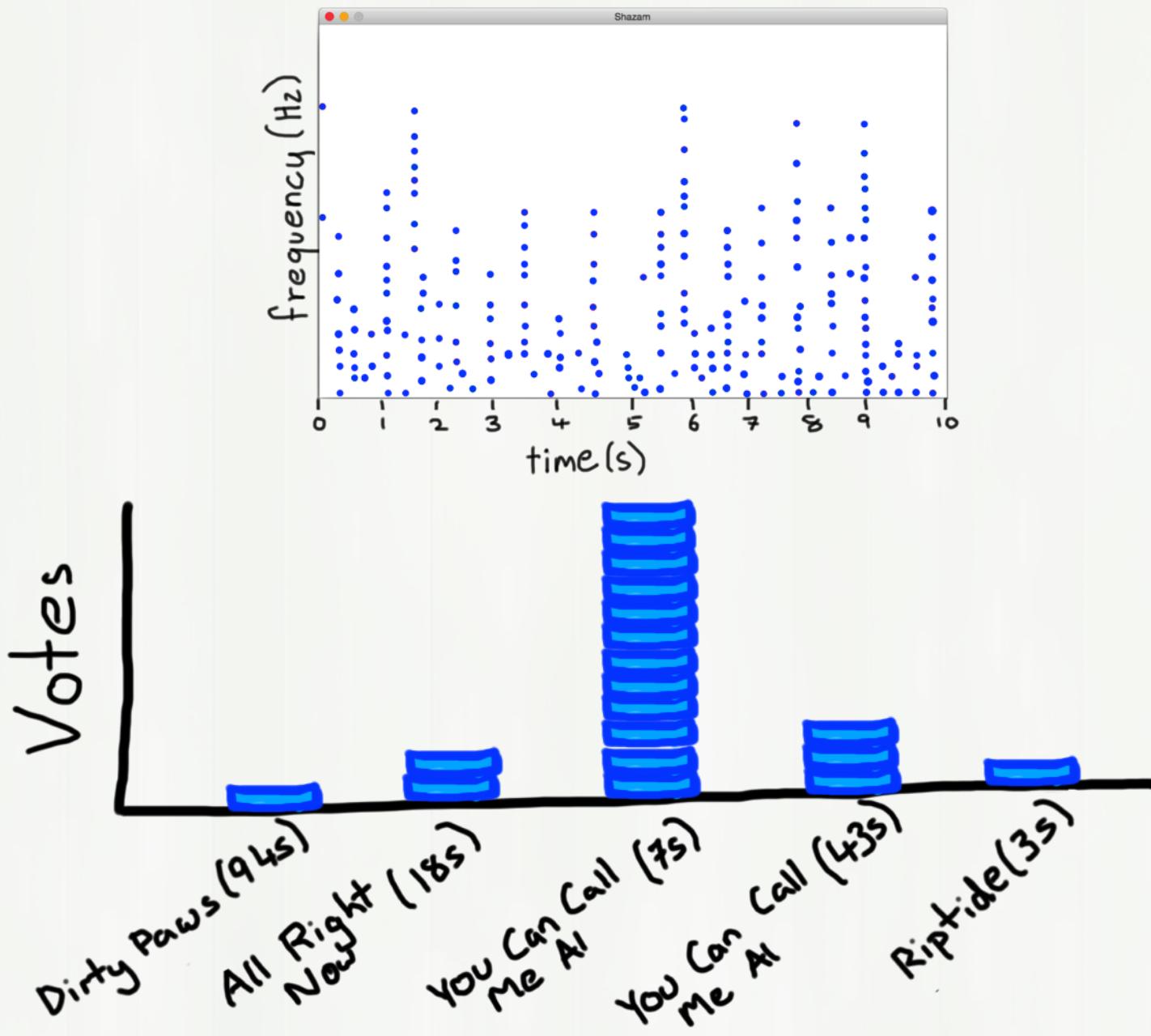
Note Pair Hashing



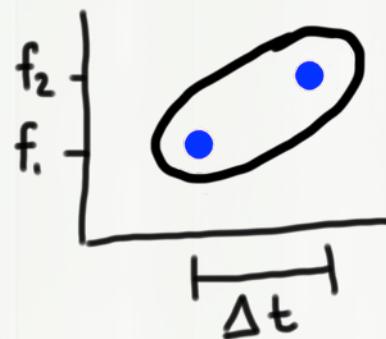
Note Pair Hashing



Note Pair Hashing



Hashing Note Pairs



```
int hash(int f1, int f2, int timeDelta) {  
    int p = 31;  
    int pre = f1 + (p * f2) + (p * p * timeDelta);  
    return pre % NUM_BUCKETS;  
}
```

You Can Call Me Al – Paul Simon. 23s,
You Can Call Me Al – Paul Simon. 54s,
Message in a Bottle – Police. 92s



Code Available

Review

HashMap Algorithm Summary

Initialization:

0. Make a large array of linked lists

Getting and Setting:

1. Hash: key \rightarrow bucketIndex.
2. Jump directly to the bucket head.
3. Run simple search over the list.

* If ever the # elements of the hash map is much larger than the number of buckets, double size and “rehash”



The HashMap needs a Hash Function

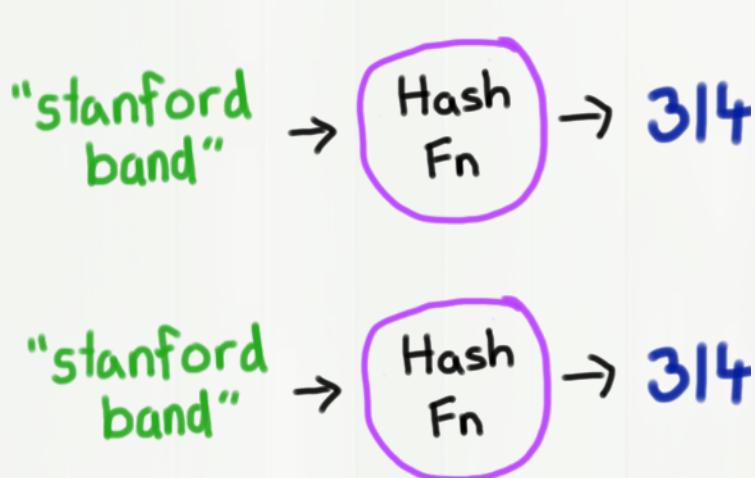


Hash Function

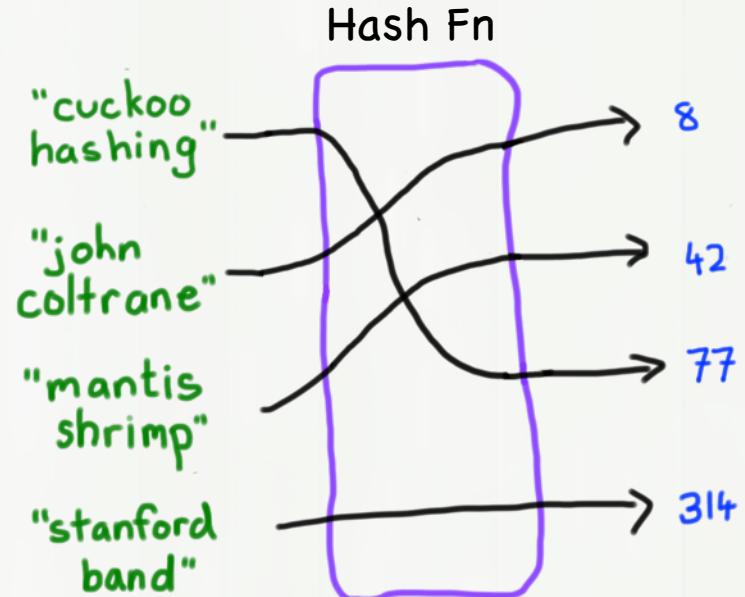
```
int hash(string key);
```



1. Consistent



2. Well Distributed



Today's Goals

1. Understand how the
HashMap works
2. Add hashing to your
algorithmic toolbox
3. Practice linked lists



Today's Goals

1. Understand how the
HashMap works
2. Add hashing to your
algorithmic toolbox
3. Practice linked lists



Thank you

Life in the internet

