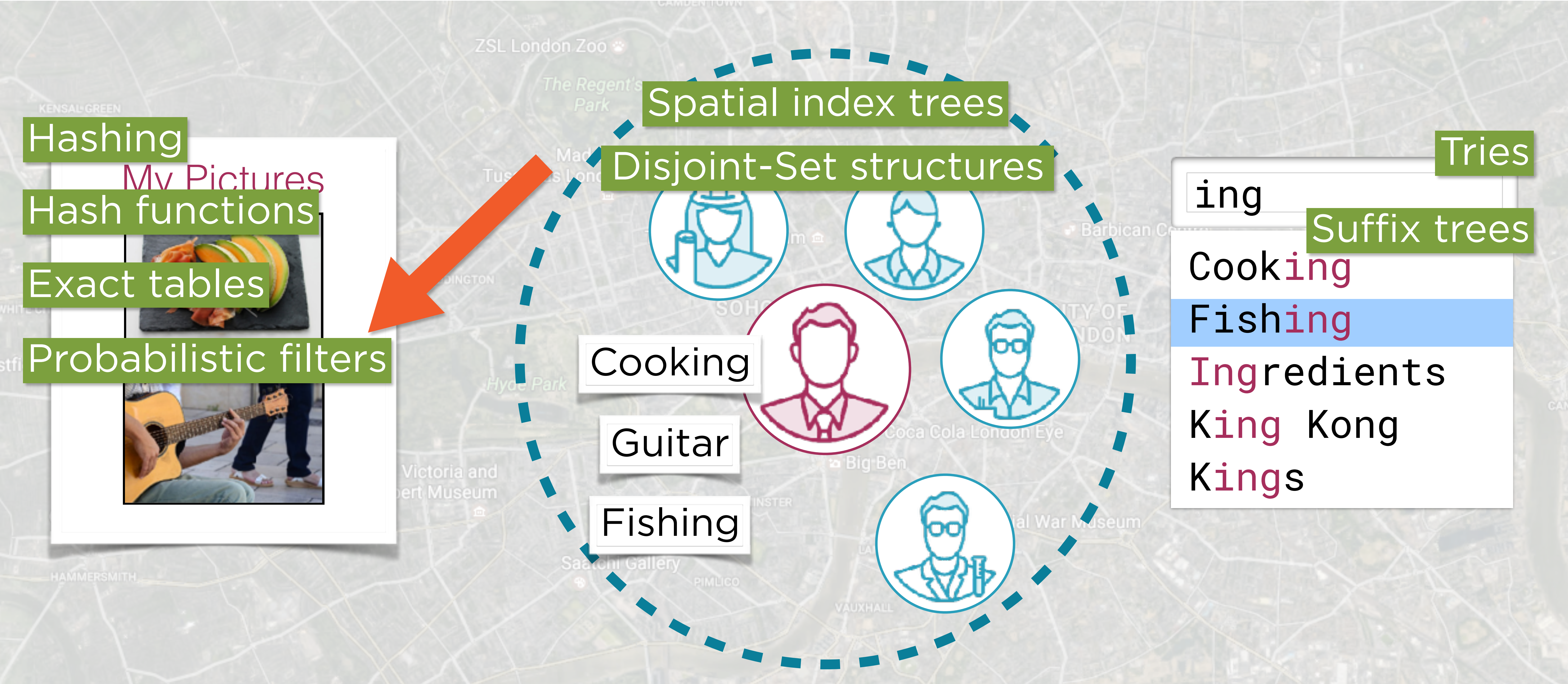
Hashing 3: Saving Space and Allowing Deletion with Cuckoo Filters

Rasmus Resen Amossen

SOLUTION ARCHITECT

rasmus.resen.org

The Match Finder App



Hashing

My Pictures

Hash functions

Exact tables

Probabilistic filters

Spatial index trees

Disjoint-Set structures

Cooking

Guitar

Fishing

Tries

ing

Suffix trees

Cooking

Fishing

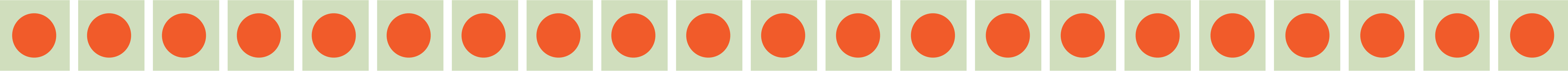
Ingredients

King Kong

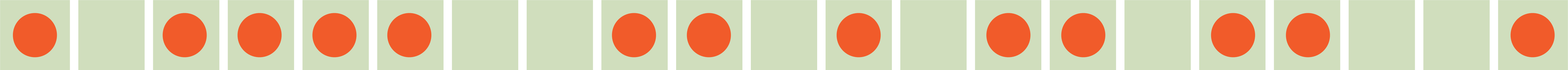
Kings

A Challenge with Bloom Filters

False positive risk increases



Lower false positive risk



Demo

Still wasting cache space

Cuckoo Filters

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
				15			32	64					15		
57		52		12							92		79		67
				49		47				82			26		
	29			33						19			74		31

Fingerprint (fp): $h_{\text{fp}}(\text{key})$

$p_1 = h_{\text{key}}(\text{key})$

XOR (eXclusive OR)

Bitwise operator

0 if bits are equal

1 if bits are different

$$\begin{array}{rr} & 1010 \\ \oplus & 0110 \\ \hline & 1100 \end{array} \qquad \begin{array}{rr} & 1100 \\ \oplus & 0110 \\ \hline & 1010 \end{array}$$

Cuckoo Filters

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
				15			32	64					15		
57		52		12		5					92		10 79		67
				49		47				82			26		
	29			33						19			74		31

Stop after “enough” iterations

Fingerprint (fp): $h_{\text{fp}}(\text{key})$

$p_1 = h_{\text{key}}(\text{key})$

$p_2 = p_1 \oplus h_{\text{key}}(\text{fp})$

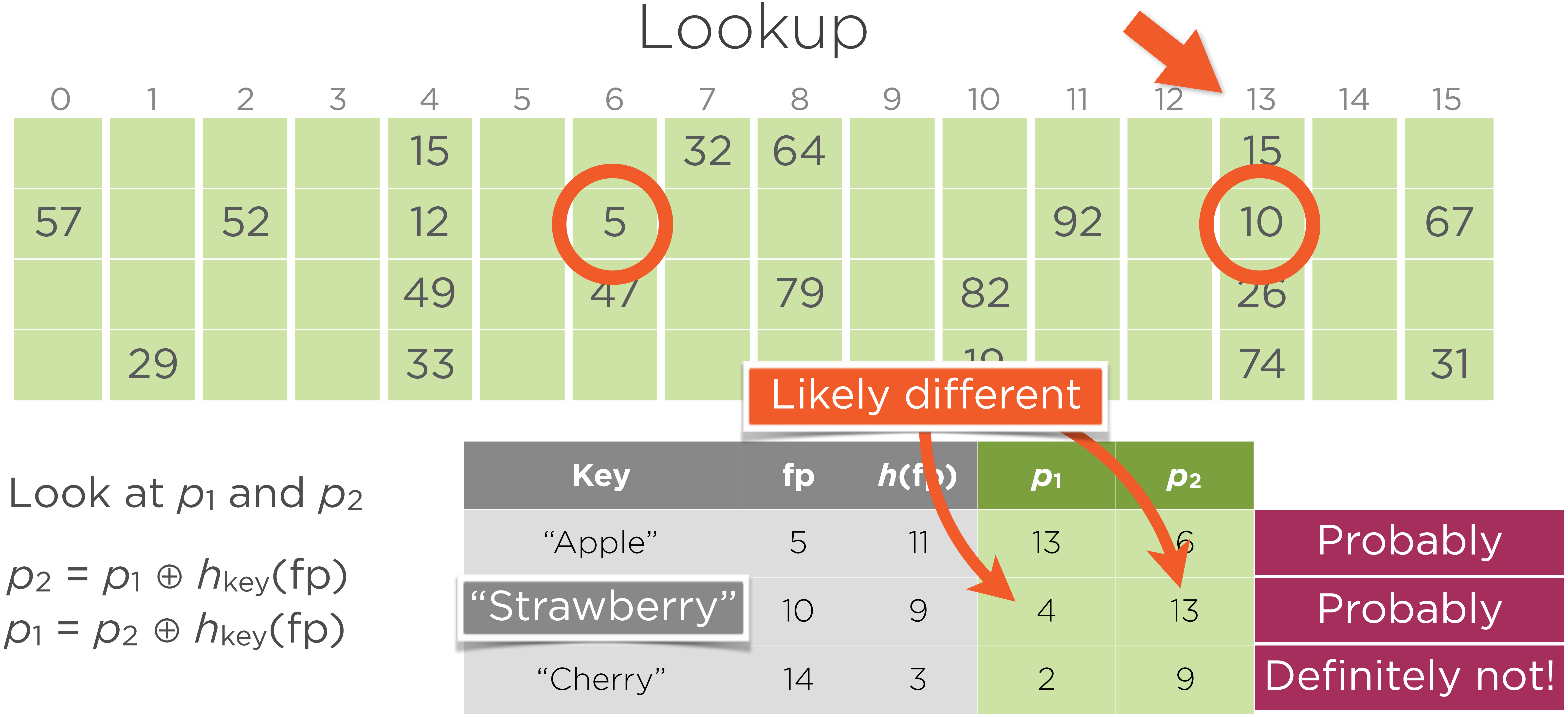
Insert fp at either p_1 or p_2

$p_1 = p_2 \oplus h_{\text{key}}(\text{fp})$

ey	fp	h(fp)	p_1	p_2
“Apple”	5	11	13	6
“Orange”	10	9	4	13
$13 \oplus 5 = 8$	79	5	13	8

Expected $O(1)$ time for insertion

Lookup



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
				15			32	64					15		
57		52		12		5					92		10		67
				49		47		79		82			26		
	29			33						19			74		31

Likely different

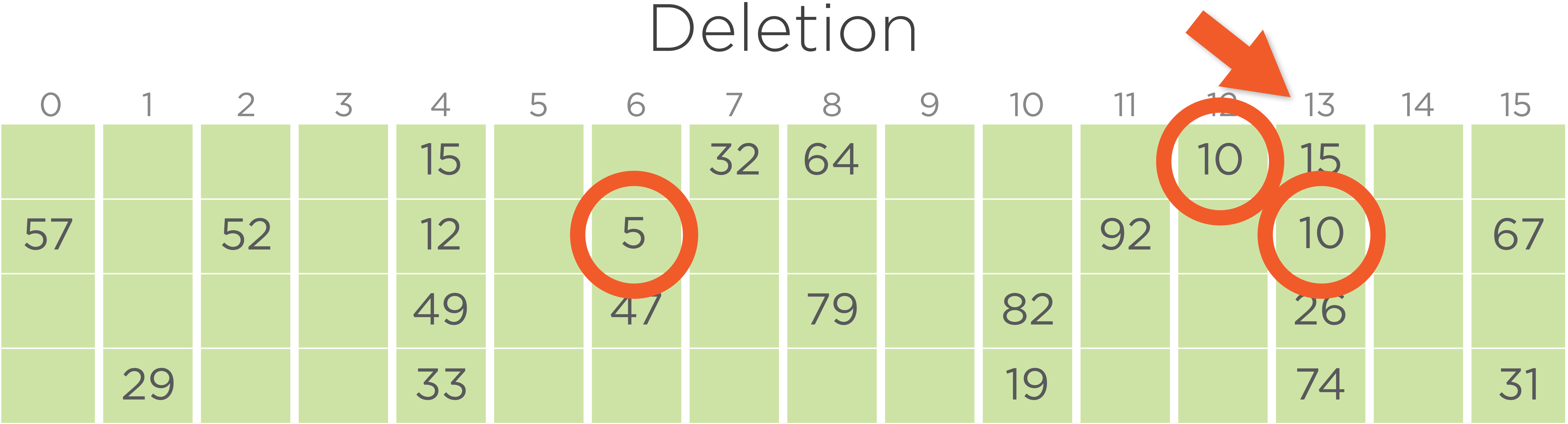
Key	fp	h(fp)	p_1	p_2	
"Apple"	5	11	13	6	Probably
"Strawberry"	10	9	4	13	Probably
"Cherry"	14	3	2	9	Definitely not!

Look at p_1 and p_2

$p_2 = p_1 \oplus h_{\text{key}}(\text{fp})$
$p_1 = p_2 \oplus h_{\text{key}}(\text{fp})$

Worst-case O(1) time for lookup

Deletion



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
				15			32	64				10	15		
57		52		12		5					92		10		67
				49		47		79		82			26		
	29			33						19			74		31

Look at p_1 and p_2

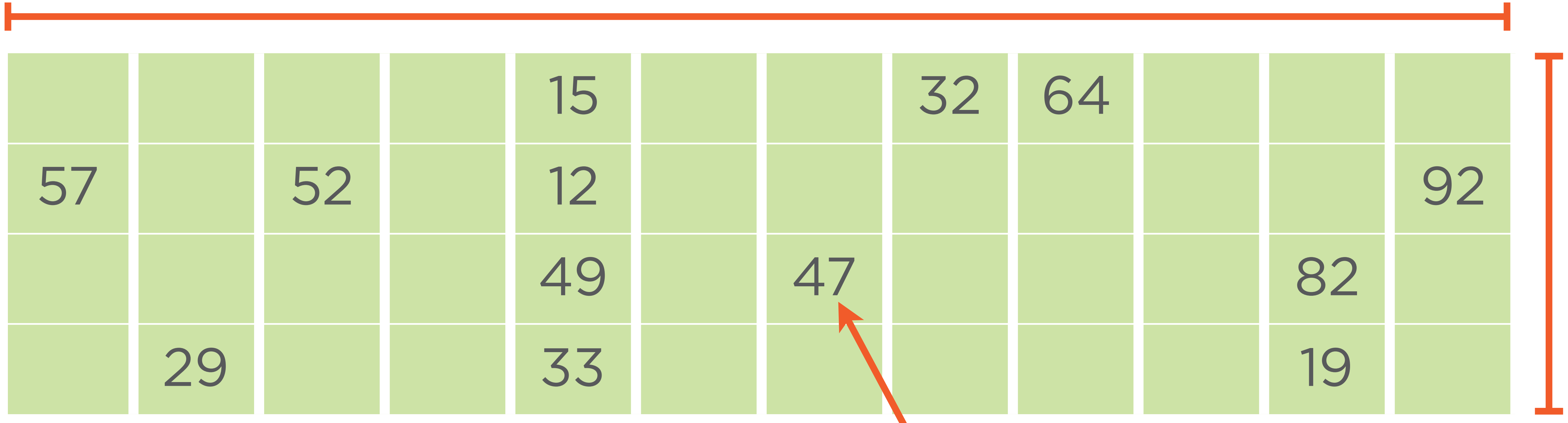
Delete matching fingerprint

(Other matches will remain)

Key	fp	*h*(fp)	p_1	p_2
"Apple"	5	11	13	6
"Orange"	10	9	4	13
"Cherry"	10	9	12	5

Worst-case O(1) time for deletion

Sizing



Around nearest power of 2 above number of items

				15			32	64			
57		52		12							92
				49		47				82	
	29			33						19	

Around 4
slots per
bucket

Around 6 bits per fingerprint
(supporting a few billion items)

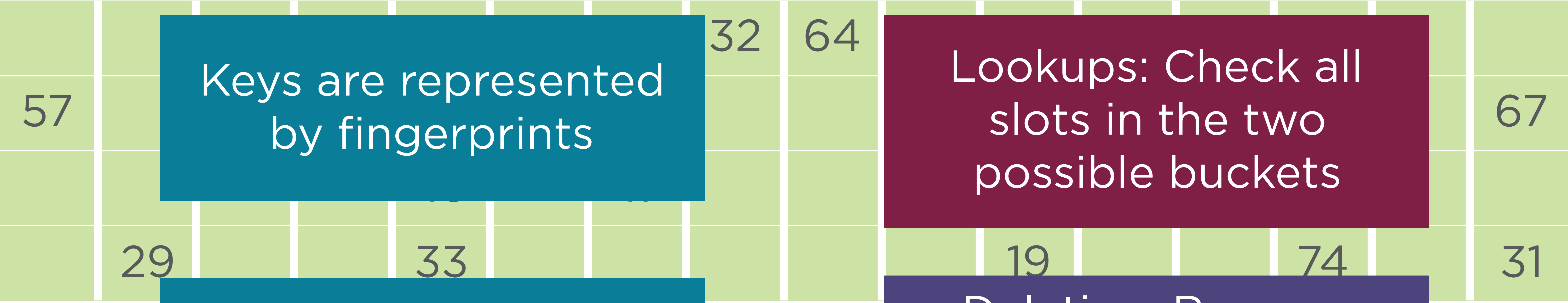
In-depth details: `tinyurl.com/cuckoobloom`

Bloom filter by example: `tinyurl.com/cuckooByExample`

Demo

Saving the cache (again)

Lessons Learned



Keys are represented by fingerprints

Two possible positions

Both can be deduced from the other using the fingerprint

Lookups: Check all slots in the two possible buckets

Deletion: Remove fingerprint

False positives

No false negatives