SPELL CHECKER

Diogo Dantas
11403718

Higor Anjos
11403767

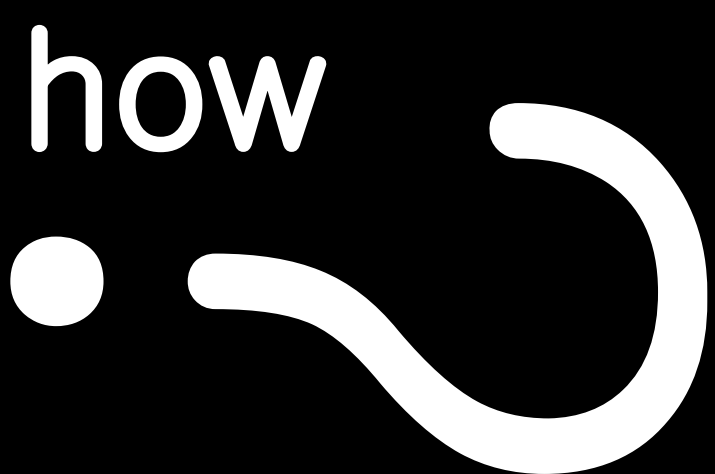
THE problem

WORD
WORD
WORD

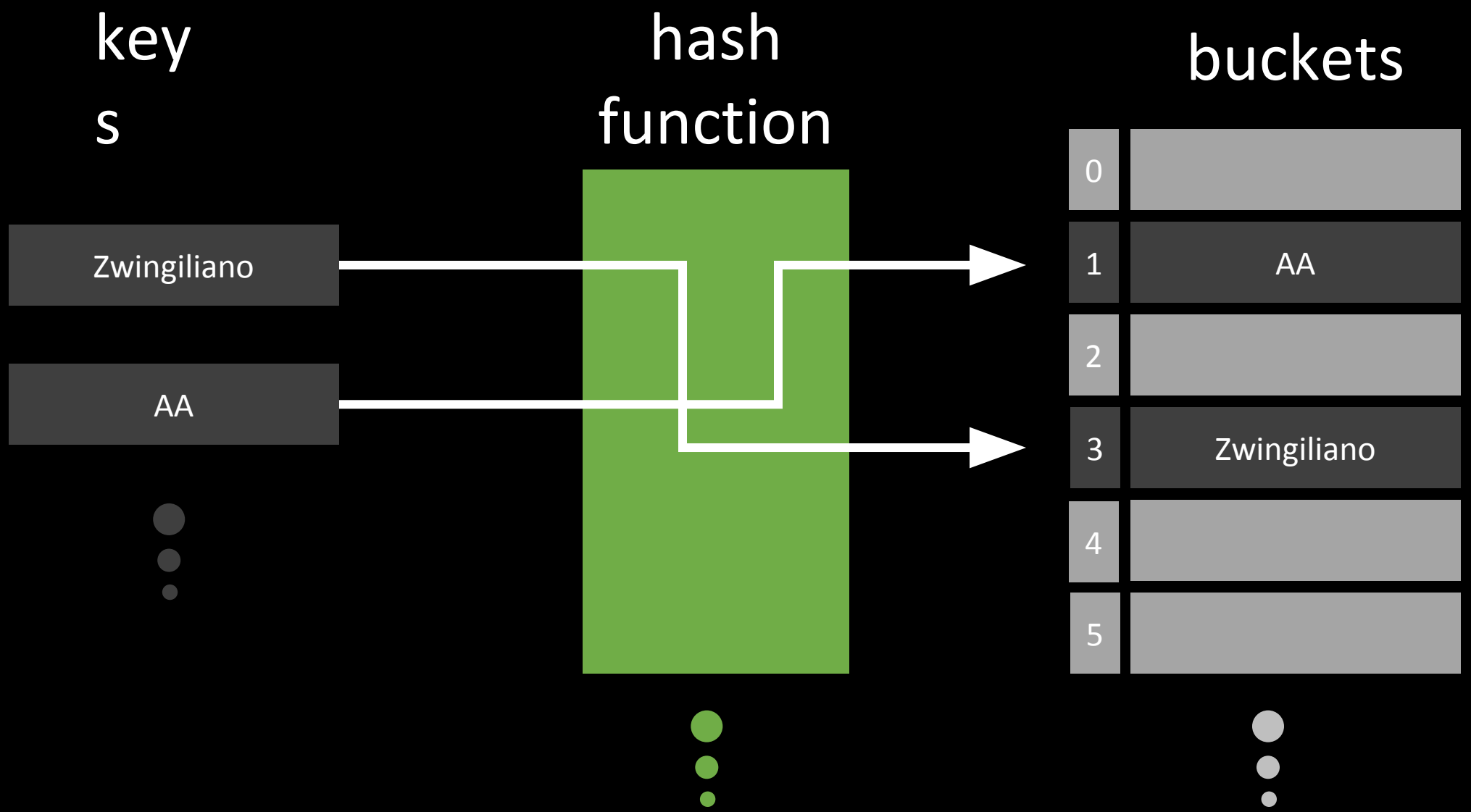
size

time

too easy


how

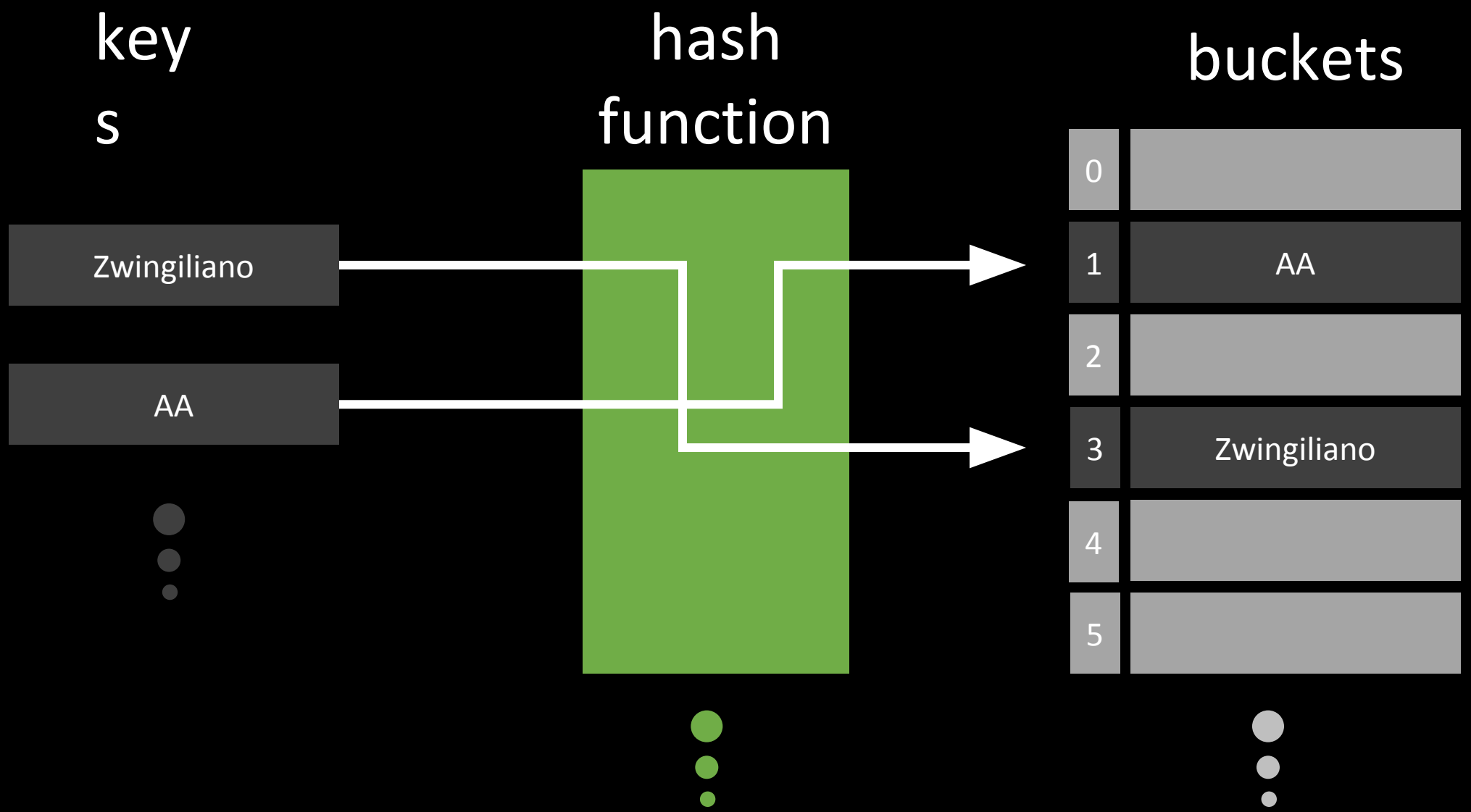
HASH
table


key
s
hash
function
buckets
Zwingiliano
AA
0
1
AA
2
3
Zwingiliano
4
5

HOW MANY

BUCKETS?

25.609


key
s
hash
function
buckets
Zwingiliano
AA
0
1
AA
2
3
Zwingiliano
4
5

HASH
function

```
hash_lose_lose (unsigned char *str) {
unsigned int hash = 0;
int c;

while (c = *str++)
    hash += c;

return hash;
}
```

"This is not the best possible algorithm, but it has the merit of extreme simplicity."

- *The C Programming Language* (book), by Brian Kernighan and Dennis Ritchie

```
hash_djb2(unsigned char *str) {
unsigned long hash = 5381;
int c;

while (c = *str++)
    hash = ((hash << 5) + hash) + c;

return hash;
}
```

- *Daniel J. Bernstein.*

```
hash_sdbm(unsigned char *str) {
unsigned long hash = 0;
int c;

while (c = *str++)
    hash = c + (hash << 6) + (hash << 16) - hash;

return hash;
}
```

- *Ozan Yigit, creator of sdbm database library.*

lose lose

7.785 ms

djb2

0.807 ms

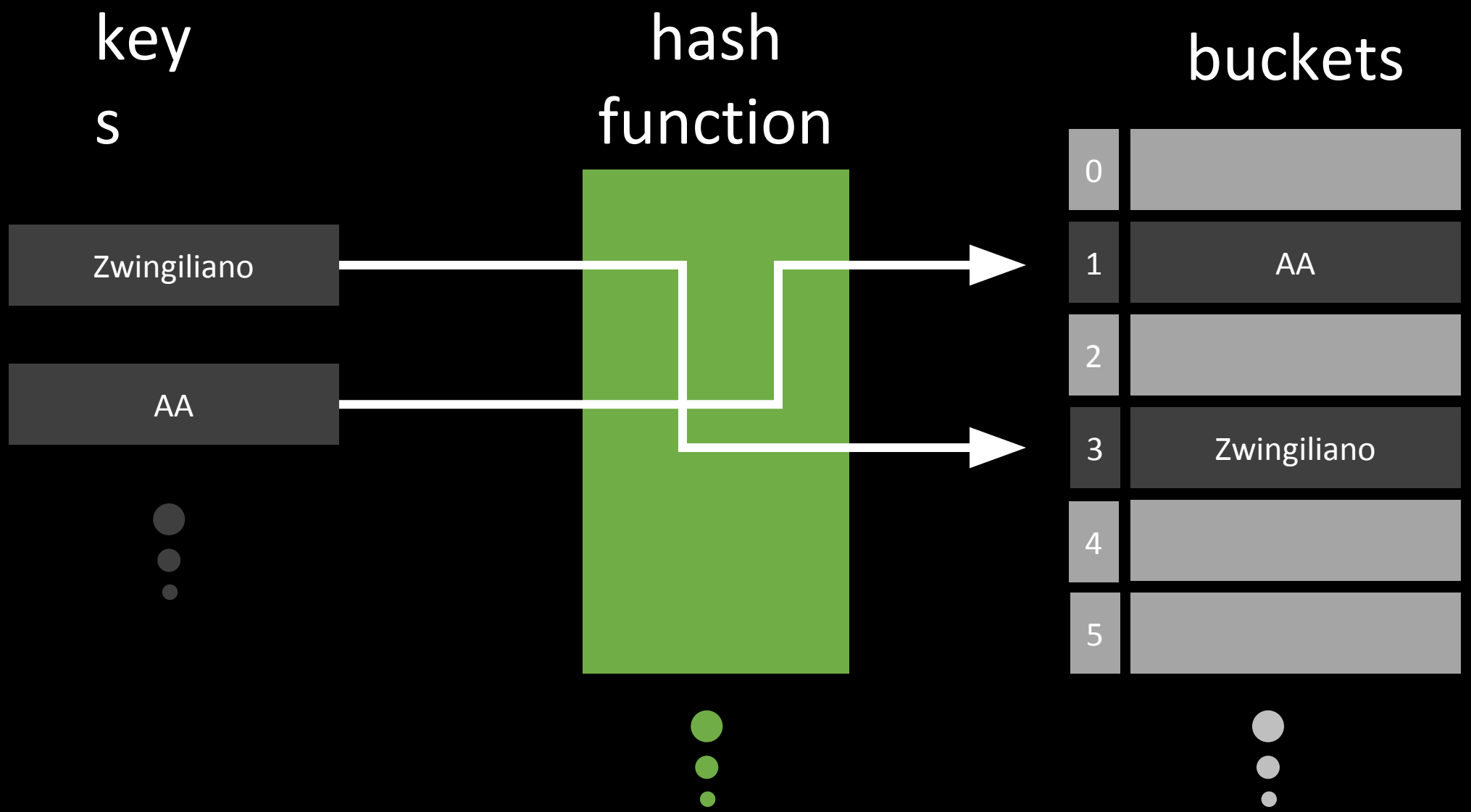
sdbm

0.787 ms

DONE?

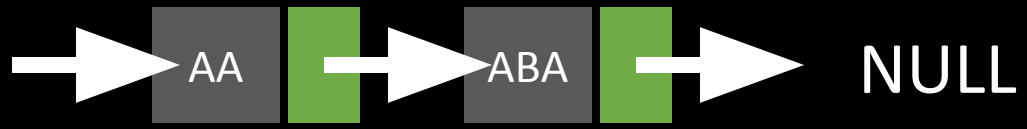
~300k words

25.609 buckets


key
s
hash
function
buckets
Zwingiliano
AA
0
1
AA
2
3
Zwingiliano
4
5

COLLISIONS

LINKED list

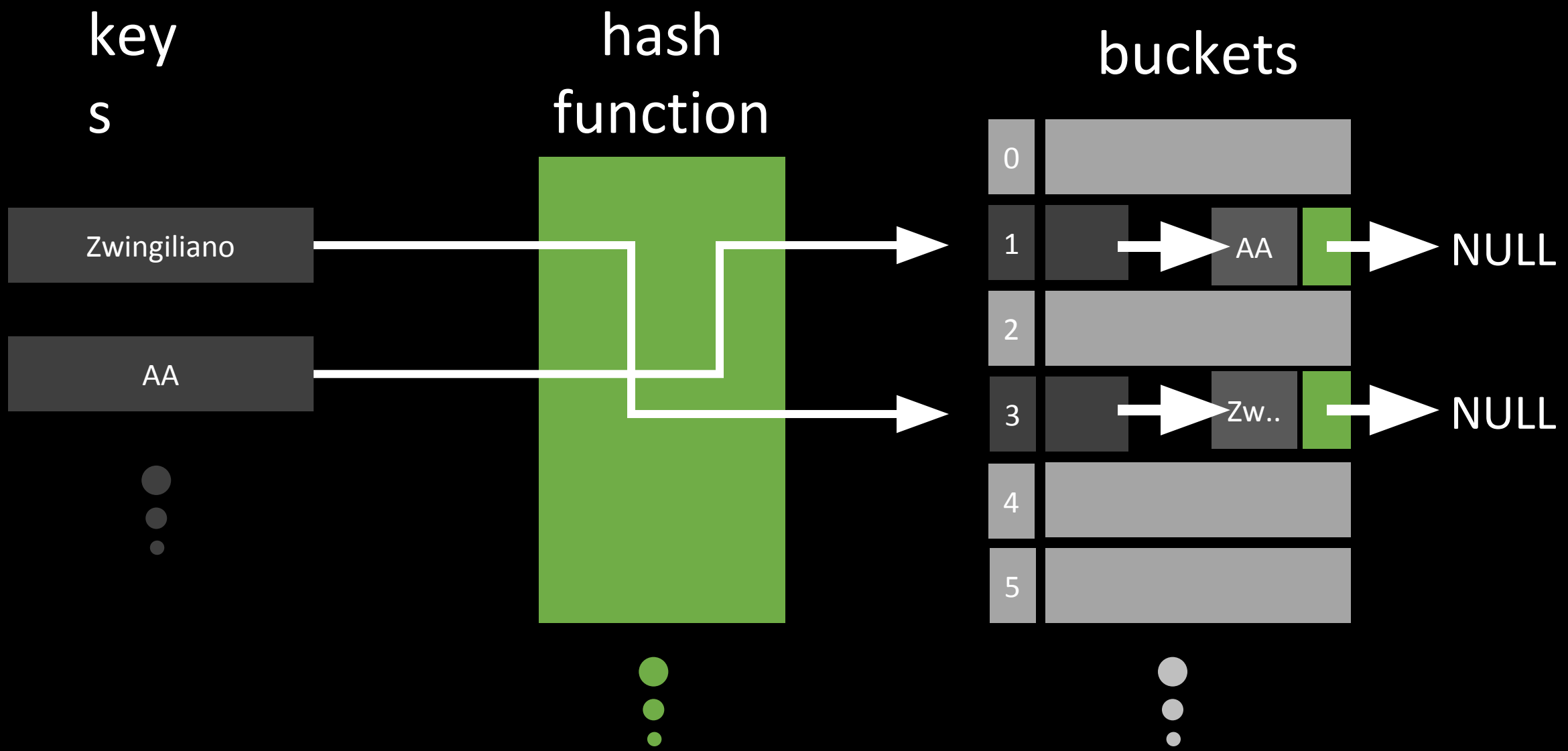


LINKED list

Insertion is constant O(1)

Dynamic allocation

The cost won’t be big


key
s
hash
function
buckets
Zwingiliano
AA
0
1
2
3
4
5
AA
NULL
Zw..
NULL

TIME TO

RUN

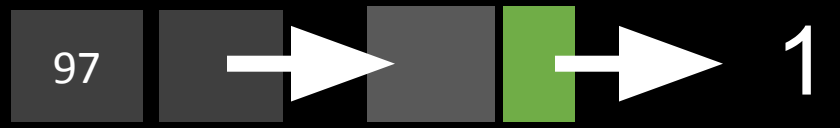
CONCLUSION

$$\frac{307308\ \text{words}}{25609\ \text{buckets}} \approx 12\ \text{avg collisions}$$

lose_lose

1779 above 12
23801 below 12
29 equal 12

‘Best Bucket’

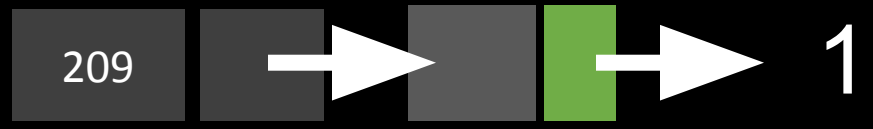


‘Worst Bucket’

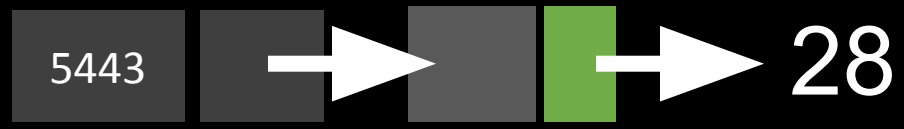
sdbm

10954 above 12
11753 below 12
2902 equal 12

‘Best Bucket’



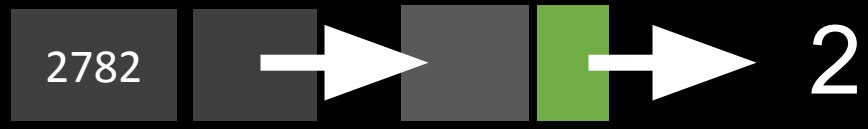
‘Worst Bucket’



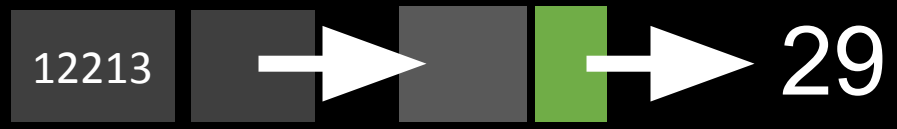
djb2

10908 above 12
11744 below 12
2957 equal 12

‘Best Bucket’



‘Worst Bucket’



Memory waste due to

FAULTY DISTRIBUTION

lose_lose	sdbm	djb2
23801	11753	11744
- 22930	- 0	- 1
871	11753	11743

possible

ENHANCEMENT

Any **QUESTIONS?**