

CSE12 - Lecture 17

Wednesday, November 8, 2023 8:00 AM

PA5 and PA3 Late/Resubmit - slip day Thursday @ 8am

PA6 released - due next week

Friday - holiday - no class

Exam2 - next Wed - covers from last exam up to and including today

Lecture 17

Map and HashTable

Hash Function (same as previous)

```
int getIndex(String k) {
    return k.length;
}
```

of buckets – 4
(i.e. the size of the array)

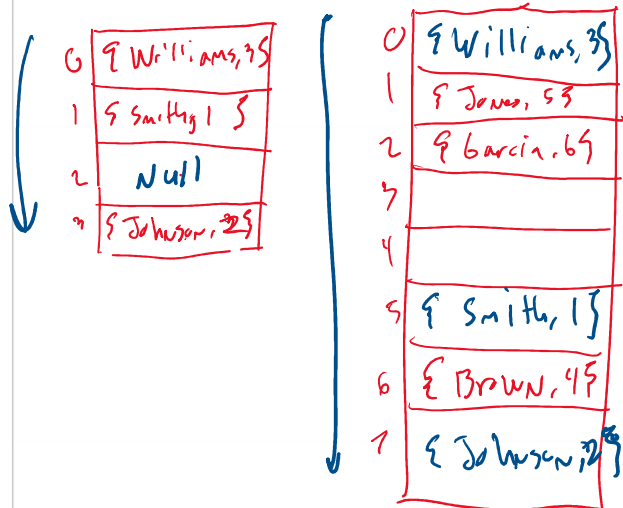
expandCapacity() called in set()

LoadFactor – 0.67

```
set("Smith", 1);
set("Johnson", 2);
set("Williams", 3);
set("Brown", 4);
set("Jones", 5);
set("Garcia", 6);
set("Miller", 7);
set("Davis", 8);
set("Rodriguez", 9);
set("Martinez", 10);
```

Draw the picture of the HashTable using Linear Probing
(using expandCapacity)

Key Value Pair < String, Integer > []



Name: _____ PID: _____ Code: **7708**

What is the run-time for this HashTable (do picture first):

set() Worst Case $\Theta(n^2)$
Best Case: $\Theta(1)$

What conditions make up the best case for set()?

NO collisions, NO EC

get() Worst Case $\Theta(n)$
Best Case: $\Theta(1)$

What conditions make up the best case for get()?

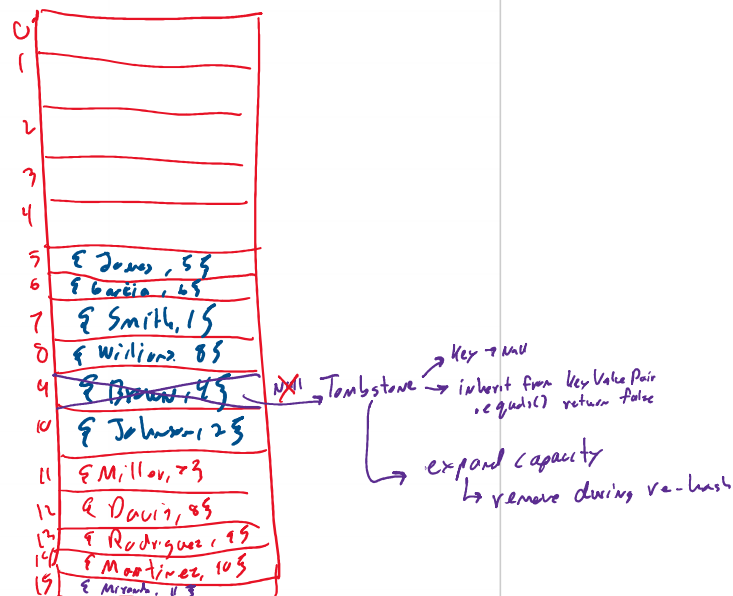
NO collisions, or maybe 1 collision

What happens if we remove something, then try to find something that collided it?

```
remove("Brown");
get("Davis");
```

What happens if we add something else?

```
set("Miranda", 11);
```



Amortized Analysis

Why is the hash function important?

even distribution
↳ less collisions

What is the run-time for ArrayList add()?

Worst Case: $\Theta(1) + \Theta(n) \rightarrow \Theta(n)$

Best Case: $\Theta(1)$

Average Case: $\Theta(1)$ per add

What is the run-time for HashTable set() using Separate Chaining and a good hash function?

Worst Case $\Theta(1) + \frac{\Theta(n)}{b \sim} \rightarrow \Theta(n)$

Best Case: $\Theta(1)$

Average Case: $\Theta(1)$ per add

What is the run-time for HashTable set() using Linear Probing and a good hash function?

Worst Case $\Theta(n)$

Best Case: $\Theta(1)$

Average Case: $\Theta(1)$ per add