CSE12 - Lecture 15

Monday, May 8, 2023 8:00 AM

PA5
PA2 late resubmit > due tomorrow

Map and HashTable

Hash Function

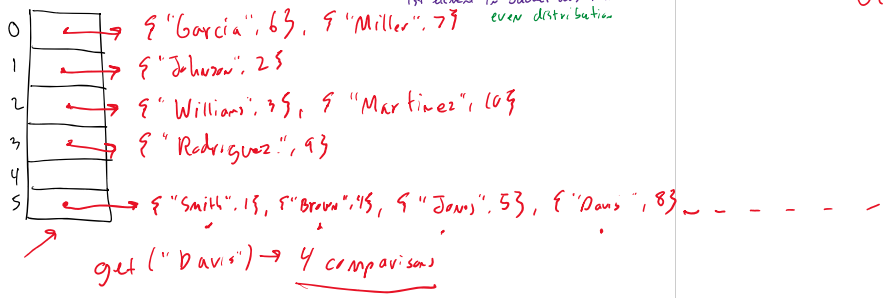
```
int getIndex(String k) {
    return k.length;
}
```

# of buckets – 6
(i.e. the size of the array)

	hash	index
set("Smith", 1);	5	5
set("Johnson", 2);	7	1
set("Williams", 3);	8	2
set("Brown", 4);	5	5
set("Jones", 5);	5	5
set("Garcia", 6);	6	0
set("Miller", 7);	6	0
set("Davis", 8);	5	5
set("Rodriguez", 9);	9	3
set("Martinez", 10);	8	2

Draw the picture of the HashTable using Separate Chaining (no expandCapacity)

Low array



get("Davis") → 4 comparisons

Mapping keys to values

```
class KeyValuePair<K, V> {
    K key;
    V value;
}
```

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

set()
Worst Case: $\Theta(1) + \Theta(N) \rightarrow \Theta(N)$ (ALEC)
Best Case: $\Theta(1)$

What conditions make up the best case for set()?
empty list for a bucket
even distribution

get()
Worst Case: $\Theta(N)$ → long chain
Best Case: $\Theta(1)$

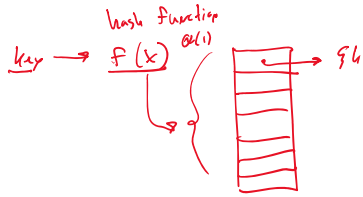
What conditions make up the best case for get()?
empty bucket, 1 element in bucket, 1st element in bucket was found
even distribution

Name: _____ PID: _____ Code: 8840

Map < String (key), Integer (value) >

ArrayList < KeyValuePair < String, Integer >> [] contents;

collision resolution
→ separate chaining



index % bucket size
hash

Θ(1)

Linked List
set w/ prepend → Θ(1)
add() → Θ(N)
get() → Θ(N) long chain