CSE12 - Lecture 15 - C00

Friday, October 28, 2022 11:00 AM

PA5 due Tuesday
PA2 Late/Resubmit → due Tuesday

Map and HashTable

Hash Function

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

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

```
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);
```

hash	index
5	5
7	1
8	2
5	5
5	5
6	0
5	5
9	3
8	2

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

Mapping keys to values

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

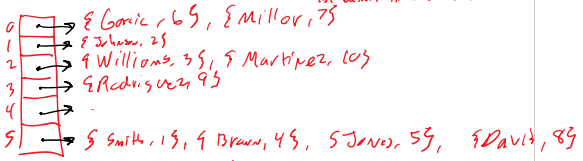
What is the run-time for this HashTable (do picture first):

set()
- Worst Case: Θ(1) or Θ(N) w/ AL expand capacity
- Best Case: Θ(1)

What conditions make up the best case for set()?
No collision → empty list
even distribution

get()
- Worst Case: Θ(N)
- Best Case: Θ(1)

What conditions make up the best case for get()?
empty bucket
1 element in the bucket
1st element in the bucket



N = 10
set → add more 5 letter names
→ LL add Θ(N)
→ AL add Θ(1) → [Θ(N) expand capacity]
→ LL prepend Θ(1)

get("Dana")
4 comparisons

get("Gress")
4 comparisons

get("Grey")
0 comparisons

even distribution

6	12	600	6000
1/6	2	100	1000
1	2	100	1000
1	2	100	1000
1	2	100	1000
1	2	100	1000

key value

Map<String, Integer>

ArrayList<KeyValuePair<String, Integer>[]> contents = new [6];