CSE12 - Lecture 15

Friday, November 3, 2023 8:00 AM

Exam 1 → done grading → 21 points → 18 points

Lecture 15

Map and HashTable

Hash Function

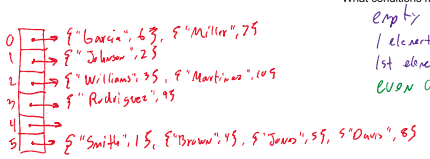
```
int getIndex(String k) {
    return k.length;
}
```

# 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
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()
- replace duplicates
- Worst Case: O(N)
- Best Case: O(1)

What conditions make up the best case for set()?
- empty bucket → no collision
- even distribution

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

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

N = 10
set → add more 5 letter names
~~↳ LL → add O(N)~~
↳ AL → add O(1) → O(N) expand capacity
→ contains → get → O(N)
↳ LL → prepend O(1)

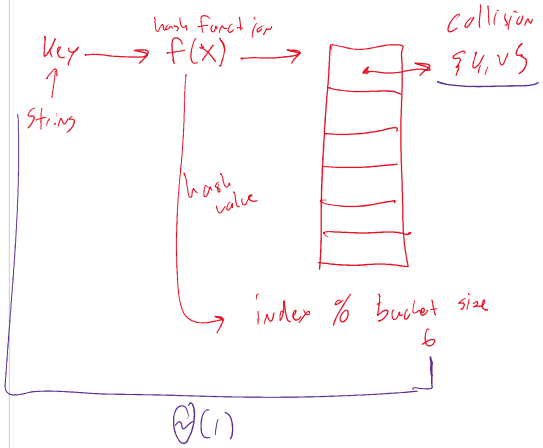
get("Davis")
4 comparisons

get("Gregg")
4 comparisons

get("Greg")
0 comparisons

Name: _____ PID: _____ Code: 9696

Map<String, Integer>
key value
ArrayList<KeyValuePair<String, Integer>[] contents
= new _____[6];



collision resolution → separate chaining