CSE12 - Lecture 17 - C00

Wednesday, November 2, 2022 11:00 AM

PA5 hard-deadline → tonight
PA6 released today
PA3 Late/Resubmit → due Tuesday

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)

Java .75 separate chaining

expandCapacity() called in set()

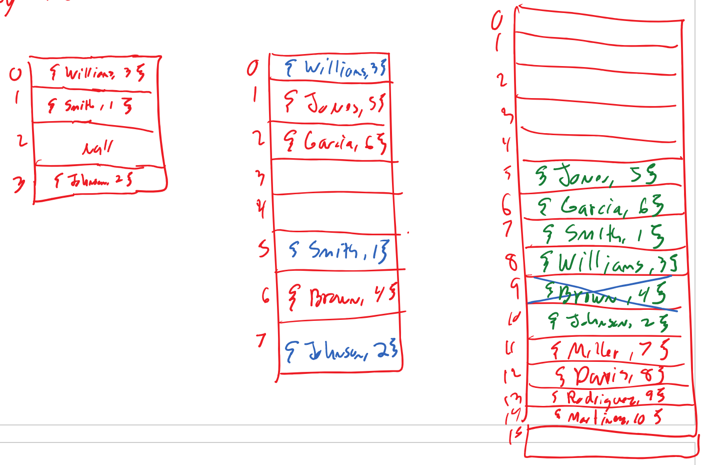
LoadFactor – 0.67

Python map linear probing

	hash	index	lf
`set("Smith", 1);`	5	~~1~~ 5	0/4
`set("Johnson", 2);`	7	~~3~~ 7	1/4
`set("Williams", 3);`	8	~~0~~ ~~0~~ 8	2/4
`set("Brown", 4);`	5	5	3/4
`set("Jones", 5);`	5	5	4/8
`set("Garcia", 6);`	6	6	5/8
`set("Miller", 7);`	6	6	6/8
`set("Davis", 8);`	5	5	7/16
`set("Rodriguez", 9);`	9	9	8/16
`set("Martinez", 10);`	8	8	9/16

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

KeyValuePair<String, Integer>[ ]



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 w/ expand capacity

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

What conditions make up the best case for get()?
No collisions

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

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

tombstone → Key → Null
→ inherit KeyValuePair<>
override equals() return false
→ expand capacity
↳ remove during rehash

What happens if we add something else?

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

Amortized Analysis

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

find()
wc $\Theta(N)$
bc $\Theta(1)$
ac $\Theta(1)$

LF = .75

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

Worst Case $\Theta(N)$

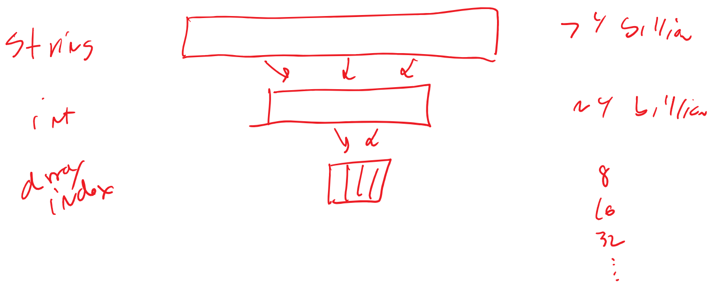
Best Case: $\Theta(1)$

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

LF ≈ .67

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

Why is the hash function important?
even distribution
↳ low # of collisions



$LF \approx .67$

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 set

$$\begin{tikzcd} 2 \arrow[r, bend right] & 4 \arrow[r, bend right] & 8 \arrow[r, bend right] & 16 \arrow[r, bend right] & 32 \arrow[r, bend right] & 64 \arrow[r, bend right] & 128 \end{tikzcd}$$

1000

$\underline{1024}$

~~||||~~ |||| → expand capacity 9 times

512
256
128
64
32
16
8
4
2

Average

$\Theta(1)$ per add