CSE12 - Lecture 17 - A00

Wednesday, November 2, 2022 8:00 AM

PA5 hard deadline → today
PA6 released today
PA7 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)

expandCapacity() called in set()

LoadFactor – 0.67

Java HashMap .75
Python dict

```
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	lf
5	1 5	0/4
7	3 7	1/4
8	0 8	2/4
5	5	3/4
5	5	1/8
7	6	5/8
6	6	6/8
5	5	7/16
9	9	8/16
8	6	9/16

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

Key Value Pair < 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, no expand capacity

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

What conditions make up the best case for get()?
No collisions → good hash function

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);
```

tombstone → Key → null
in hash from Key Value Pair or .equals() → returns false

→ expand capacity to remove during rehash

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

LF → .75

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

Worst Case $\Theta(1) * \Theta(N)$

Best Case: $\Theta(1)$

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

LF → .67

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

Worst Case $\Theta(1) * \Theta(N)$

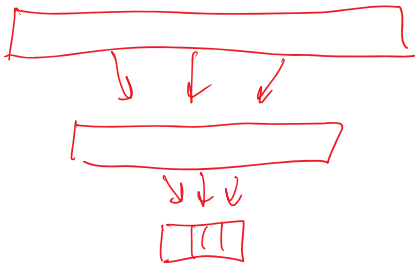
Why is the hash function important?

even distribution → low # of collisions

String

int

array index



> 4 billion

~ 4 billion

8
16
32
64

LP → 1.67

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

Worst Case $\Theta(1) * \underline{O(N)}$

Best Case: $\Theta(1)$

Average Case: $\Theta(1)$

2 4 8 16 32 64 128 256 512

2 4 8 16

1000

IIII IIII → ec (9) times

1024 spots

512 elements
256
128
64
32
16
8
4
2