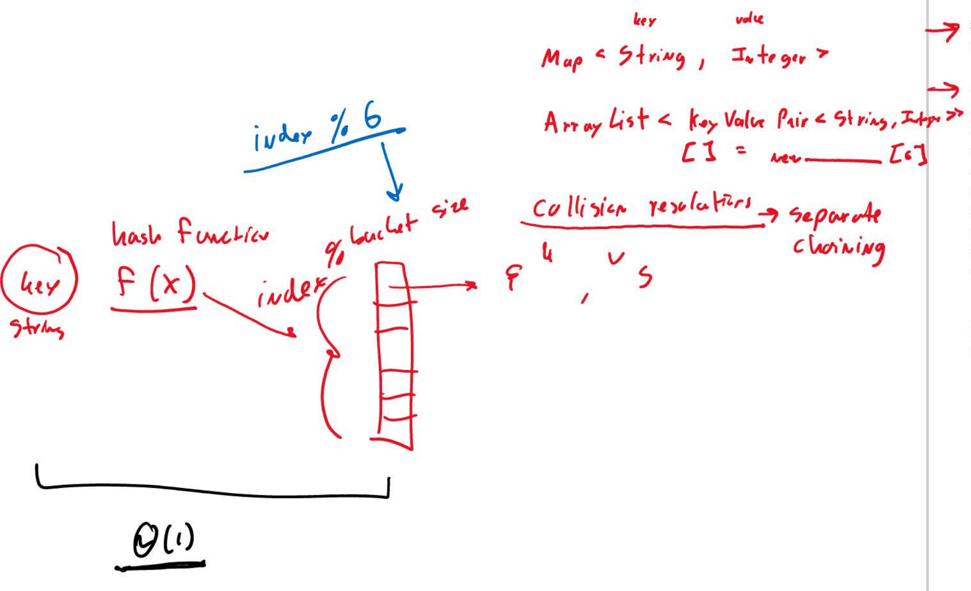
PA5 due Today
PA6 out tomorrow
Exam 2 → Fri @ 8am
due Sun @ 8am

run time
sorting
hash tables/maps



Map and HashTable

Mapping keys to values

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

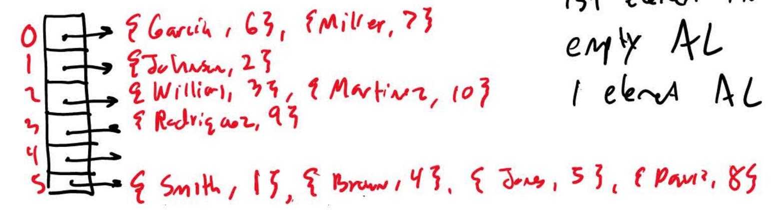
Hash Function

```
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)


0 → {Garcia, 6}, {Miller, 7}
1 → {Johnson, 2}
2 → {Williams, 3}, {Martinez, 10}
3 → {Rodriguez, 9}
4
5 → {Smith, 1}, {Brown, 4}, {Jones, 5}, {Davis, 8}

N=10
set → add more 5 letter names
→ $\boxed{\Theta(N)}$ LL add
$\Theta(1)$ AL add
[$\Theta(N)$ AL expand capacity

get ("Davis")
4 comparisons
get ("Gregs")
4 comparisons

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

set()
Worst Case $\Theta(N)$ expand capacity
Best Case: $\Theta(1)$

What conditions make up the best case for set()?
empty array list (no collisions)

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

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

Hash Function (same as previous)

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

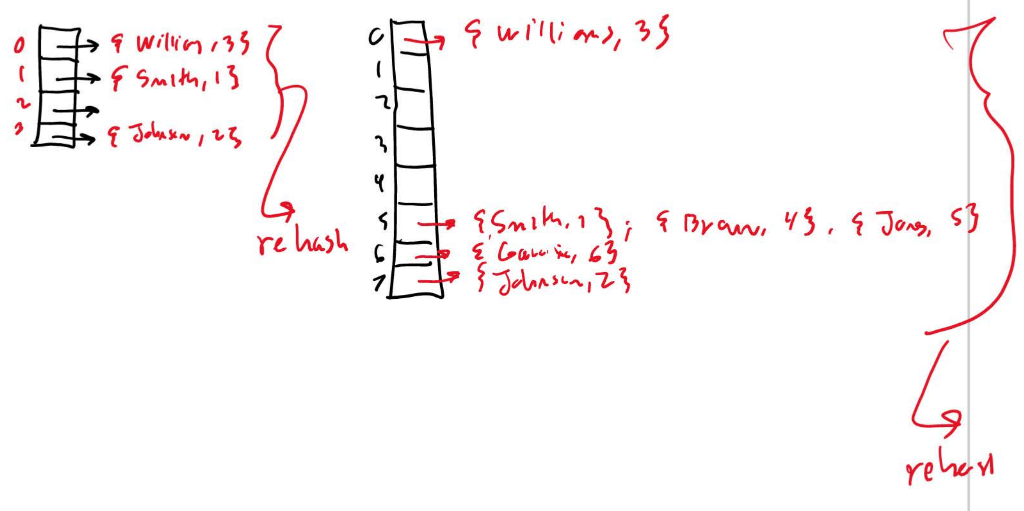
# of buckets – 4
(i.e. the size of the array)

expandCapacity() called in set()

>= LoadFactor – 0.75 $\frac{\text{size}}{\text{capacity}}$

	hash	index			LF	
set("Smith", 1);	5	~~1~~	~~5~~	5	0	
set("Johnson", 2);	7	~~3~~	~~7~~	7	1/4	
set("Williams", 3);	8	~~0~~	~~0~~	0	2/4	
set("Brown", 4);	5	~~5~~		5	3/4	3/8
set("Jones", 5);	5	~~5~~		5	4/8	
set("Garcia", 6);	6	~~6~~		6	5/8	
set("Miller", 7);	6	6			~~6/8~~	6/16
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 Separate Chaining (using expandCapacity)



Does the run-time change with expandCapacity()?

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

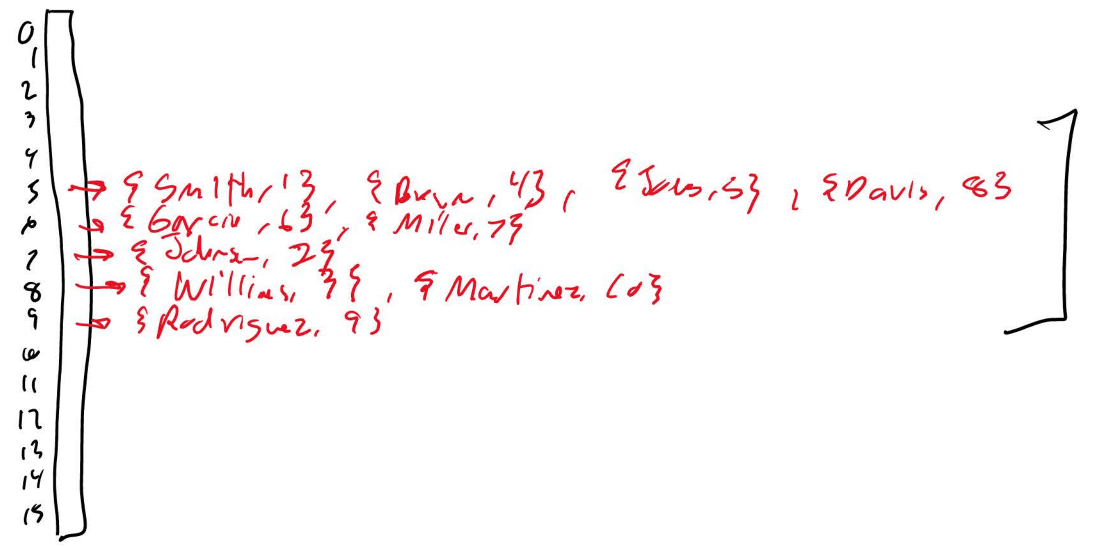
set()
Worst Case $\Theta(N^2)$
Best Case: $\Theta(1)$

What conditions make up the best case for set()?

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

What conditions make up the best case for get()?

Why is the hash function important?


0
1
2
3
4
5 → {Smith, 1}, {Brown, 4}, {Jones, 5}, {Davis, 8}
6 → {Garcia, 6}, {Miller, 7}
7 → {Johnson, 2}
8 → {Williams, 3}, {Martinez, 10}
9 → {Rodriguez, 9}
10
11
12
13
14
15

clustering
↓
bad hash function