Abstract Data Type
Maps (dictionaries)

COMP128 Data Structures

Maps

- A collection designed to translate (i.e. map) keys to their corresponding values
  - Both keys and values must be Objects
- No ordering of keys in the collection
- Usually O(1) access by key to retrieve its corresponding value
  - If implemented cleverly

Examples

- Contacts
  - Key: Last, First Name; Value: Phone Number
- Glossary
  - Key: Word; Value: Definition
- Index of Book
  - Key: Word; Value: Page Numbers
- Web Server
  - Key: URL; Value: Machine and File Name

Typical Operations

Operation	Pre-Condition	Post-Condition
`isEmpty`	none	same map
`containsKey(Object item)`	none	same map
`containsValue(Object item)`	none	Same map

Typical Operations

Operation	Pre-Condition	Post-Condition
`isEmpty`	none	same map
`containsKey(Object item)`	none	same map
`containsValue(Object item)`	none	same map
`put(K key, V value)`	none	map has 1 more key-value pair
`get(Object key)`	none	same map
`remove(Object key)`	none	map has 1 fewer key-value pair

Iterating over a Map #1

```
for ( Map.Entry<String,Integer> entry : testMap.entrySet()){

entry.getKey();
entry.getValue();

}
```

Iterating over a Map #2

```
Iterator<Map.Entry<String,Integer>> itr1 =
        testMap.entrySet().iterator();


while(itr1.hasNext()) {
    Map.Entry<String,Integer> entry = itr1.next();
    entry.getKey();
    entry.getValue();
}
```

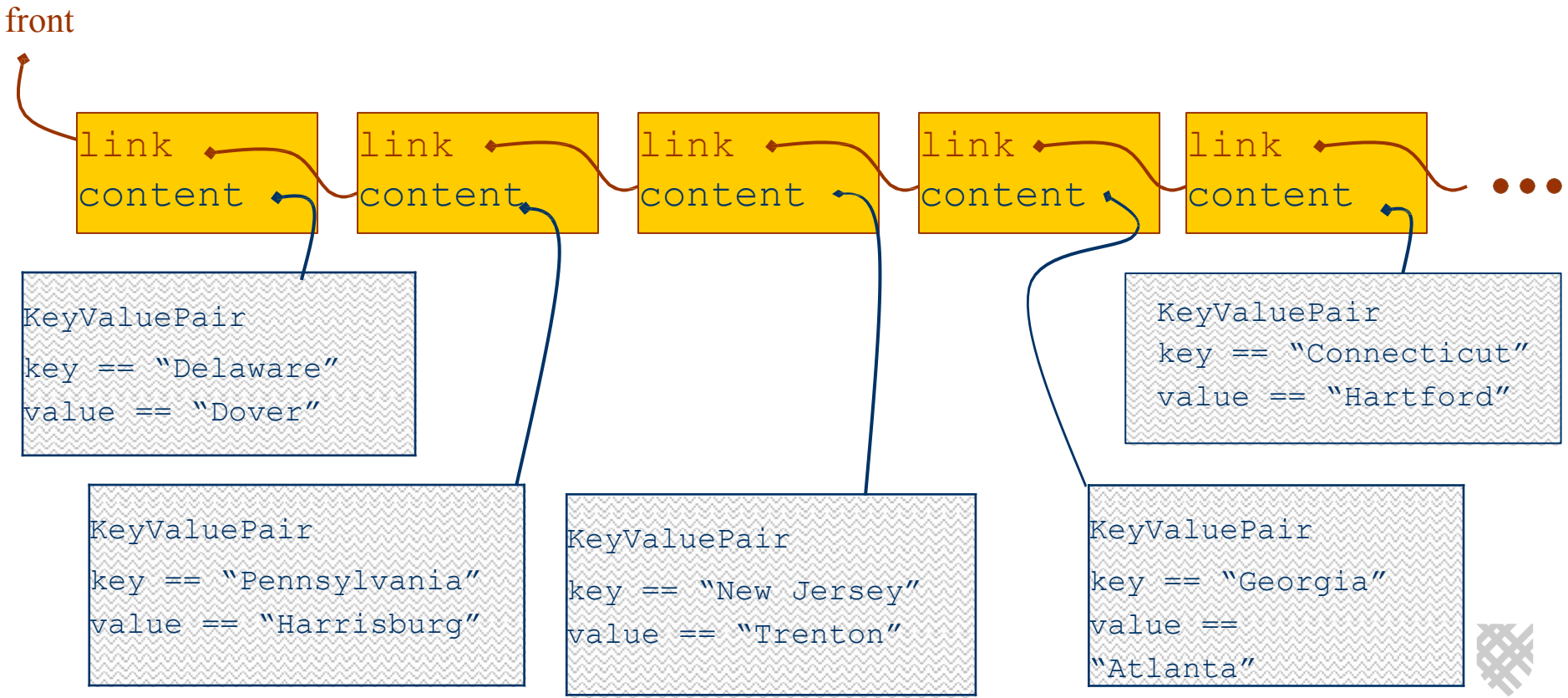
Iterating over a Map #3

```
for (String key : testMap.keySet()) {

    testMap.get(key);

}
```

In-class Activity

Word Counter Activity

Using List for Implementation



Using List for Implementation

- is an array of KeyValuePair items
- an item is placed in the array at index key.hashCode() % length (a hash function)

Hash Table Example (1st letter as hash value)

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	...
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

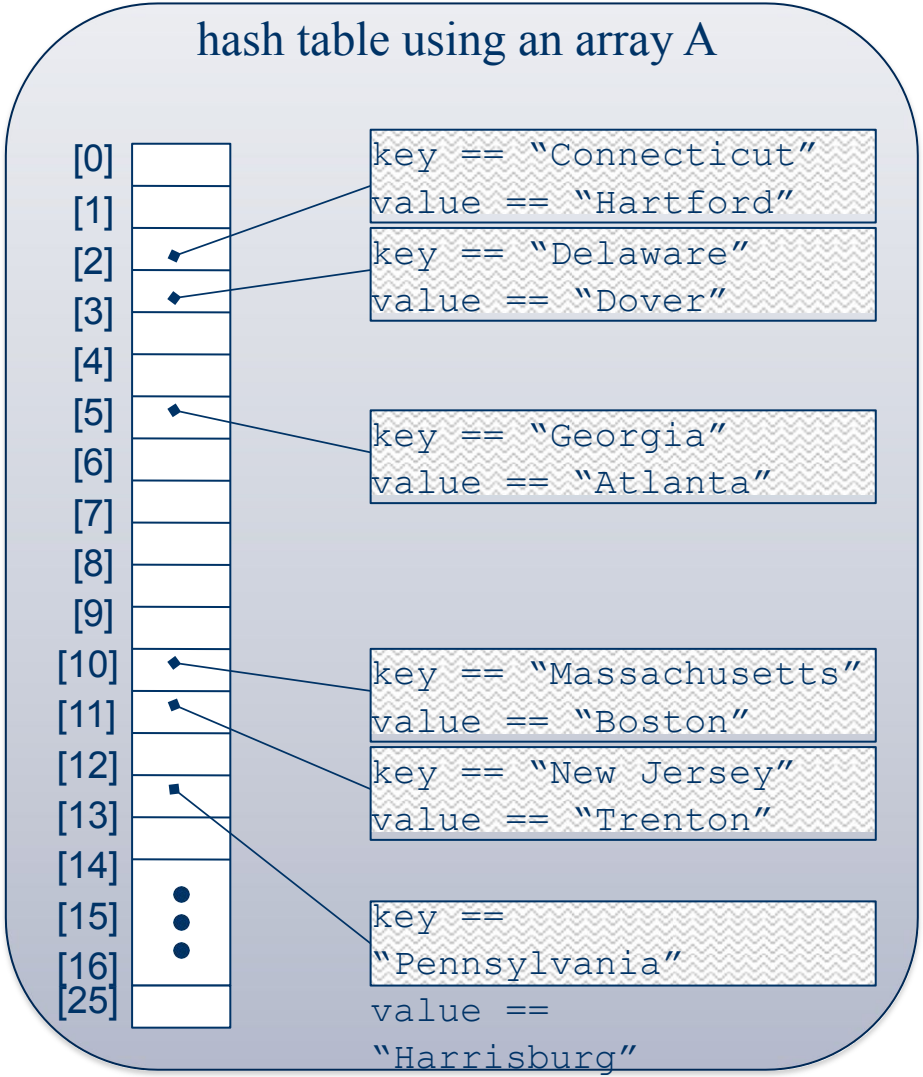
`key == "Delaware"` hashCode() → 3

`key == "New Jersey"` hashCode() → 13

`key == "Georgia"` hashCode() → 6

Representation

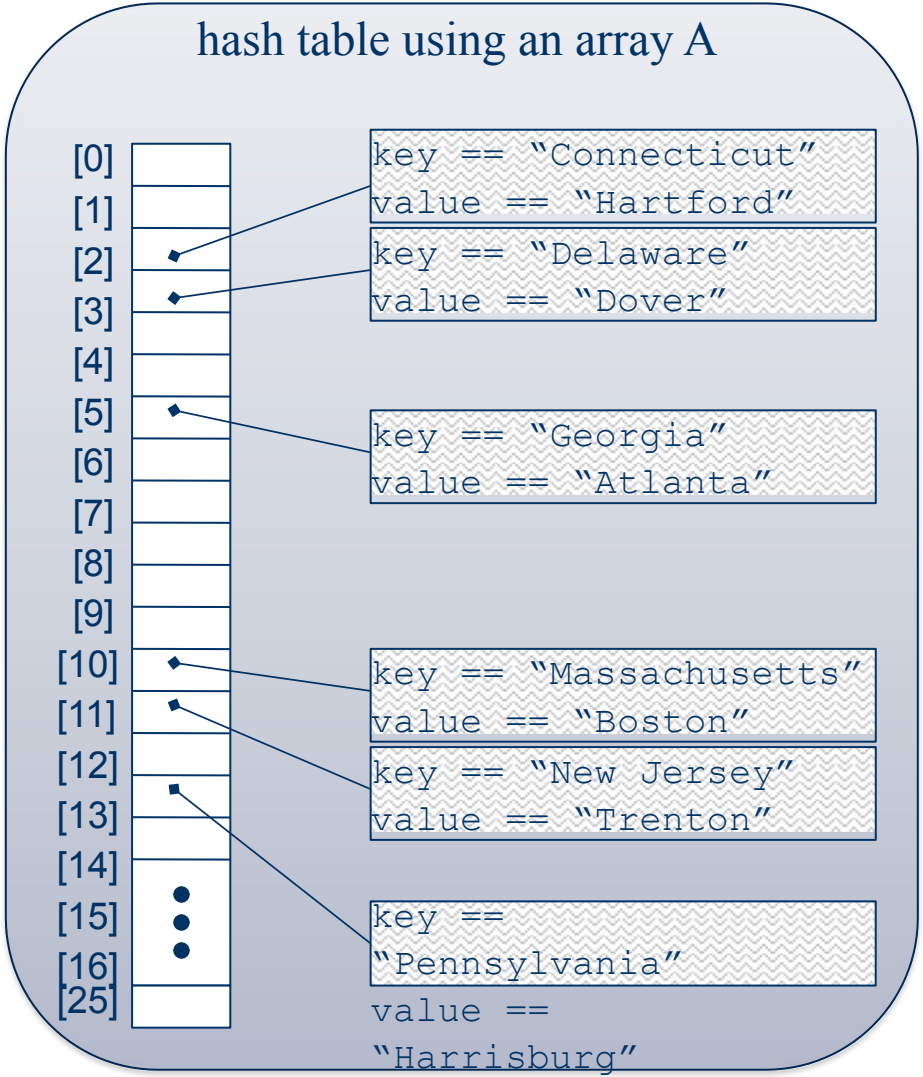
A[key.hashCode()] contains reference to KeyValuePair object



Representation

A[key.hashCode()] contains reference to KeyValuePair object

What is the most efficient algorithm to lookup the capital of Pennsylvania using this hash table?

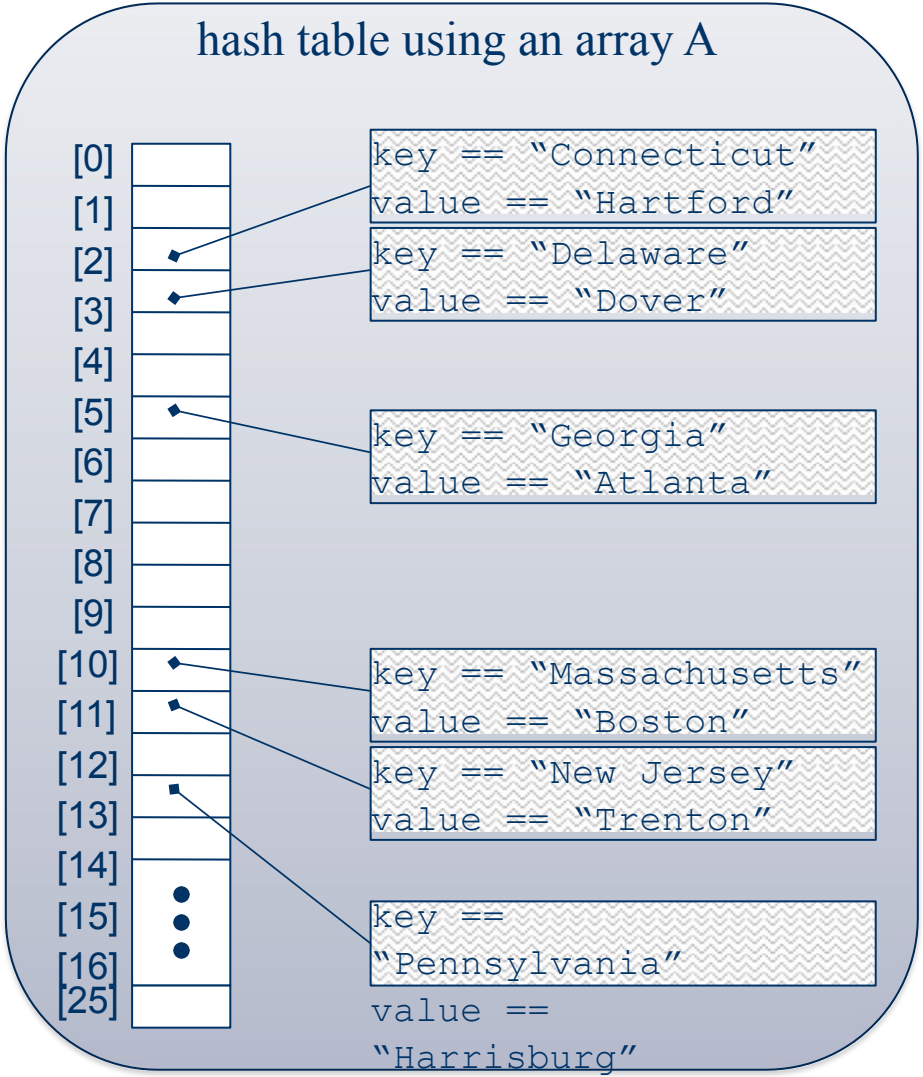


Representation

A[key.hashCode()] contains reference to KeyValuePair object

What is the most efficient algorithm to lookup the capital of Pennsylvania using this hash table?

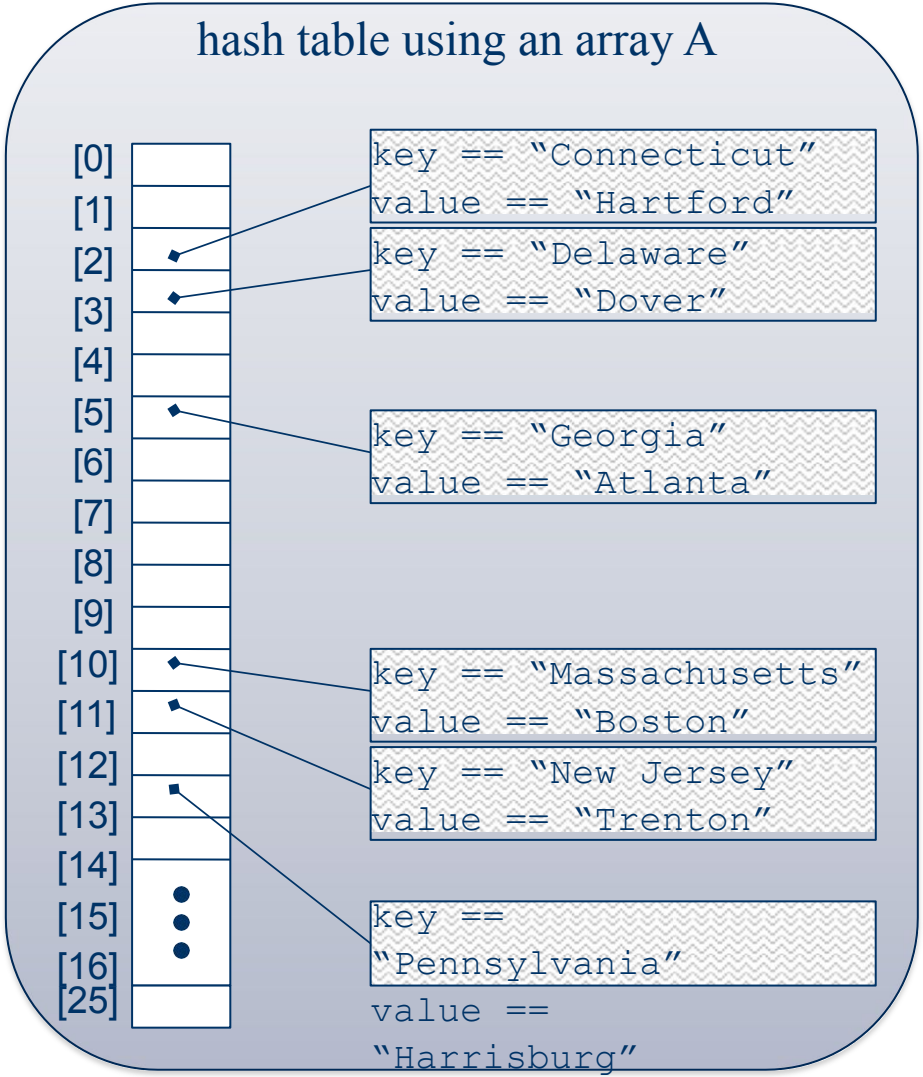
What about looking up the capital of Florida at this time?



Collisions

When Maryland is inserted into this table, a collision occurs.

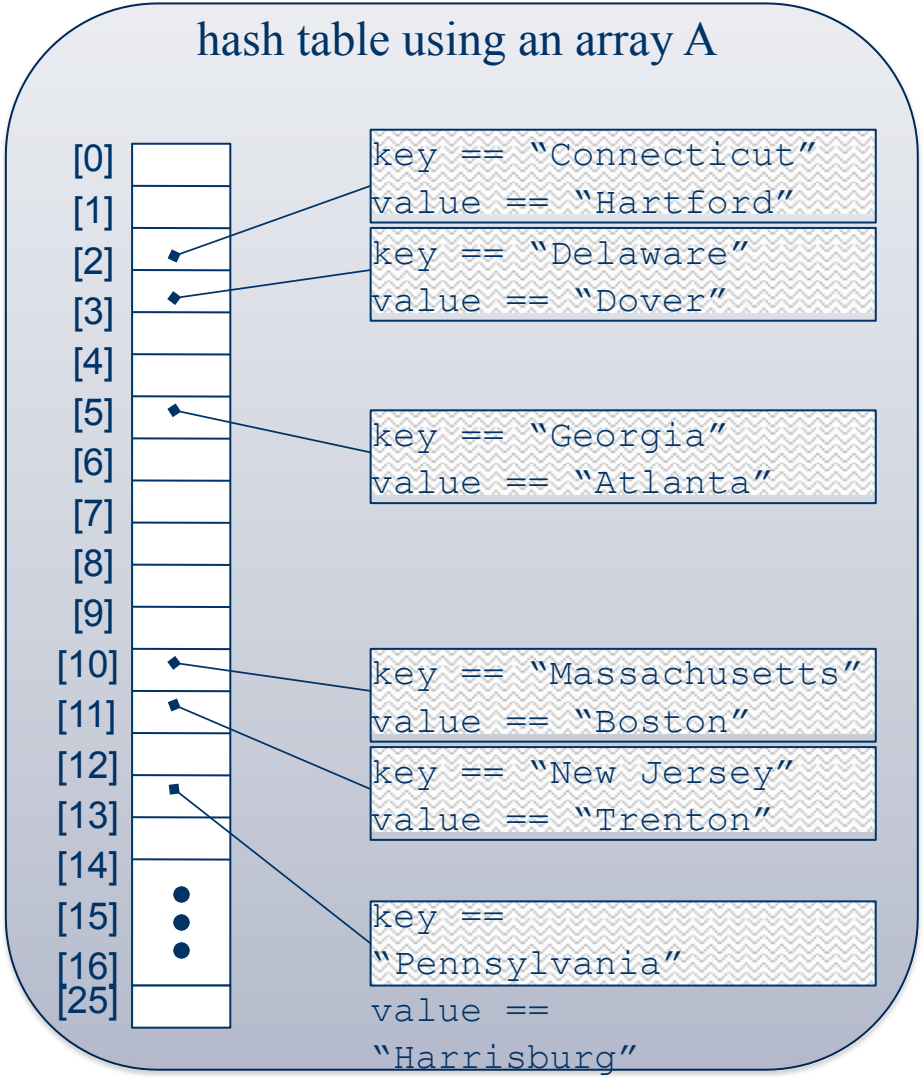
With 50 states and only 26 cells in the hash table, there are other collisions.



Collisions

Question:
Where do you put the following?

```
key == "Maryland"
value == "Annapolis"
```

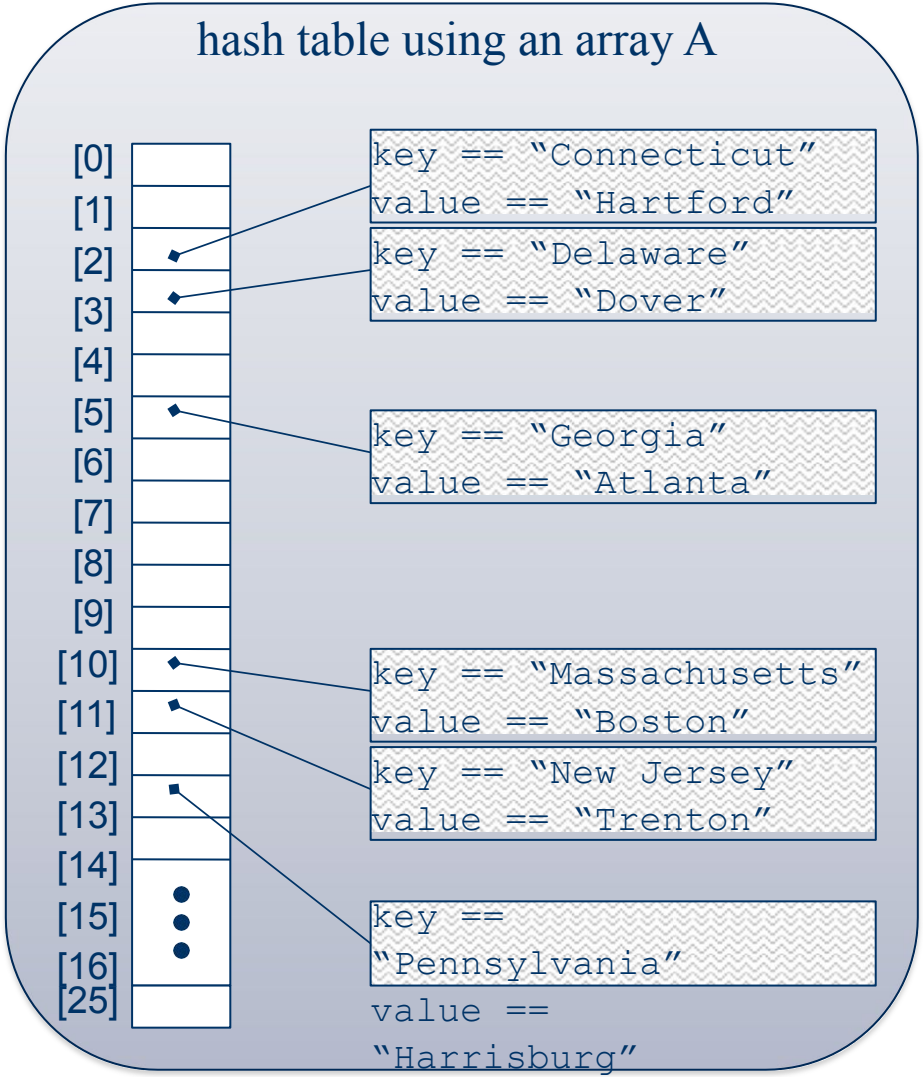


Collisions

Question:
Where do you put the following?

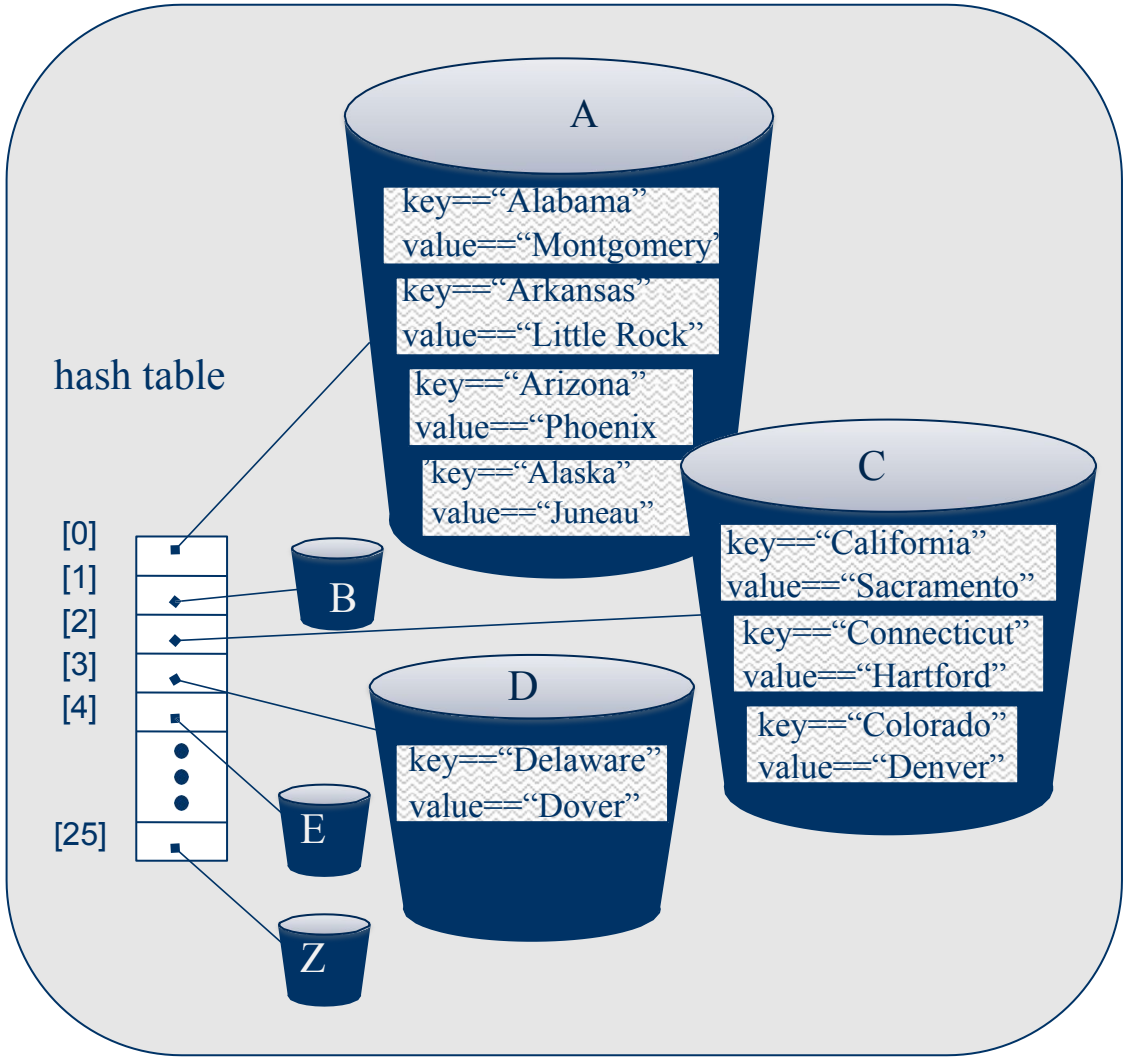
```
key == "Maryland"
value == "Annapolis"
```

Answer:
Think of each hash table cell as a bucket (container).



Hash Buckets

Note that some buckets have many collisions, and others are empty.



Collision Resolution

There are alternative collision resolution strategies.

- **Open hashing (Chaining)** - implement each bucket as a separate container (perhaps a linked list)
- **Closed hashing (Open Addressing)** - force every object into its own cell of the hash table. This requires another function(s) to calculate a different index in the event of a collision.

In-class Activity

Hash Code Activity

Hash Code In-class Activity

- f(Student object) = 17
- f(Student object)= studentName.hashCode()
- f(Student object)= ((Integer)studentID).hashCode()

Selecting a Hash Function

Characteristics of a Good Hash Function

- It must yield a valid hash table index.
  - use h(x) == expressionOfX % tableSize
- It must be efficient to calculate.
- It must minimize collisions.
  - How many U.S. state names begin with the letter “N”? How many begin with the letter “Z”?

Selecting a Hash Function

Characteristics of a Good Hash Function

- It must yield a valid hash table index.
  - use h(x) == expressionOfX % tableSize
- It must be efficient to calculate.
- It must minimize collisions.
  - How many U.S. state names begin with the letter “N”? How many begin with the letter “Z”?
  - **Ideas for better hash functions:**
    - folding - combine multiple parts of the key
    - use the middle of the key

Selecting a Hash Function

A different hash function for the state capitals:

hashCode(s) = (1st letter of state name for s
+ last letter of state name for s
+ middle letter of state name for s) % 26

Why is this an improvement?

Load Factor

At some point the array will get too many collisions and need to expand. This happens when the load becomes greater than the load factor.

- load = numElements / arrayLength
- 0.75 is a common load factor

Expanding causes the array to double in size and the items must be rehashed

Java Implementation

In Java, several classes in the java.util package implement the Map interface (e.g HashMap, TreeMap)

For a class to work as a key, you must override equals() and hashCode()

In-class Activity

Recommender Activity