CS151 Intro to Data Structures

Maps

Announcements

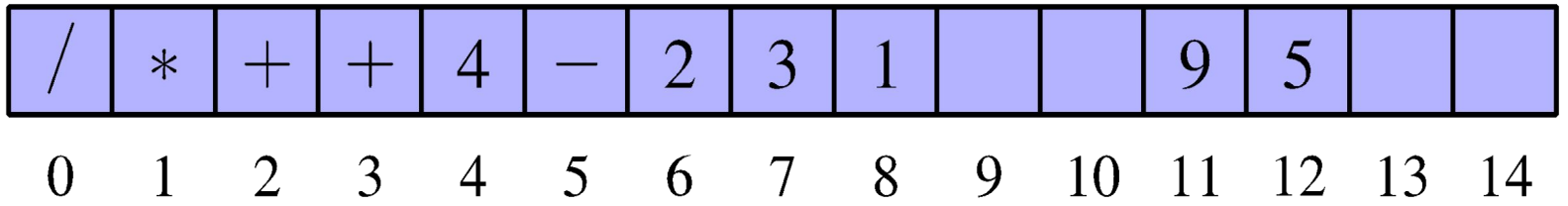
No lab today
Lab8 and HW06 due thursday

Outline

- Maps
  - Operations:
    - get
    - put (insert)
    - remove
- ArrayMap Implementation
- Intro to Hash Maps

Array/ArrayList

How do we access items in an array?



Maps

- Also called “dictionaries” or “associative arrays”

- Similar syntax to an array:
  - `m[key]` retrieves a value
  - `m[key] = value` assigns a value
  - keys need not be ints

- data structure that stores a collection of key-value pairs

Key-Value Pairs

- Each element in a map consists of (K, V)

- The key is used to identify the value
  - In an array, this would be the index

- **Examples**: what are the keys and values here?
  - Dictionary
  - Phone Book
  - Student grades

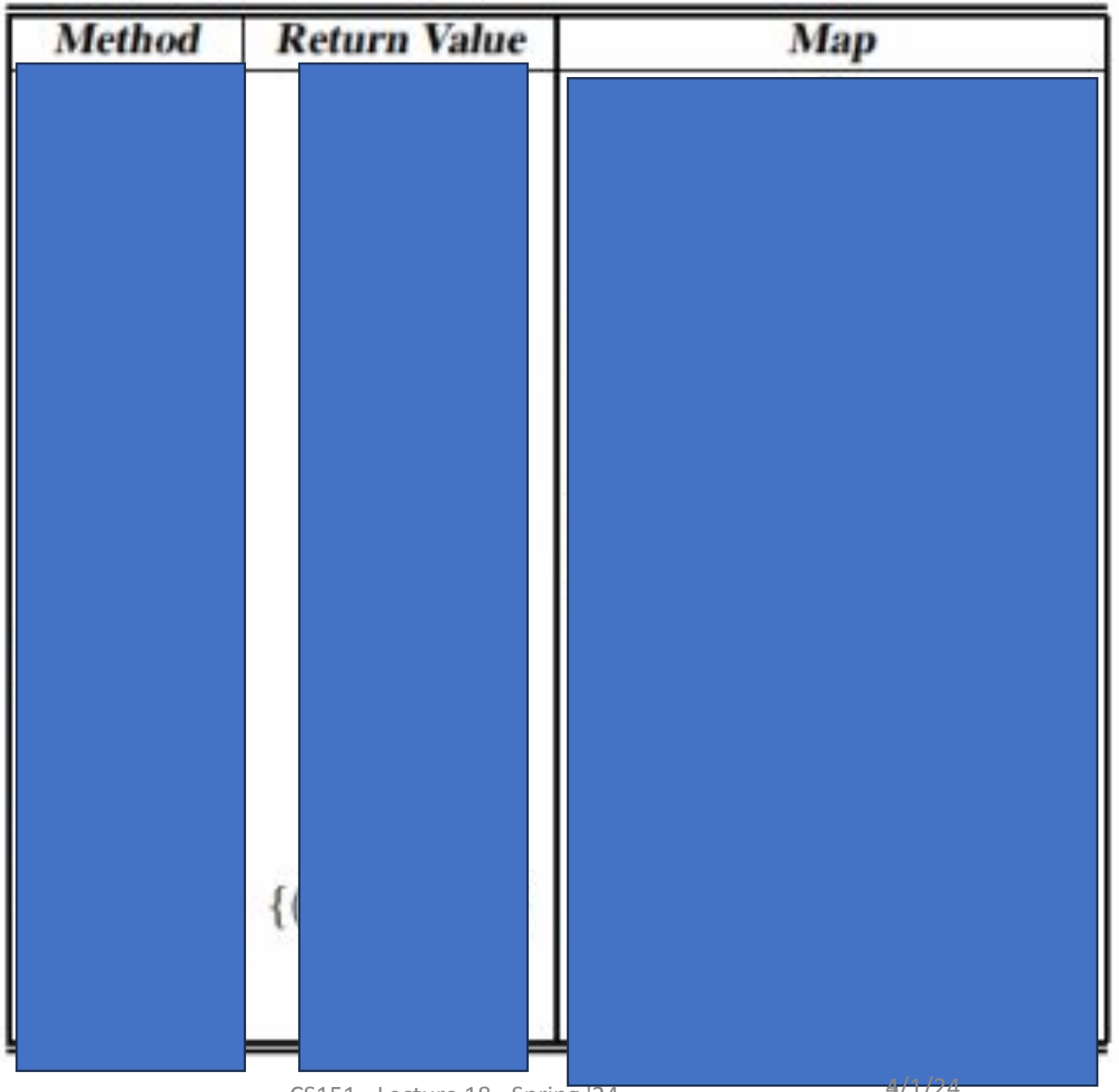
Map

- A indexable collection of key-value pairs
- Multiple entries with the same key are not allowed

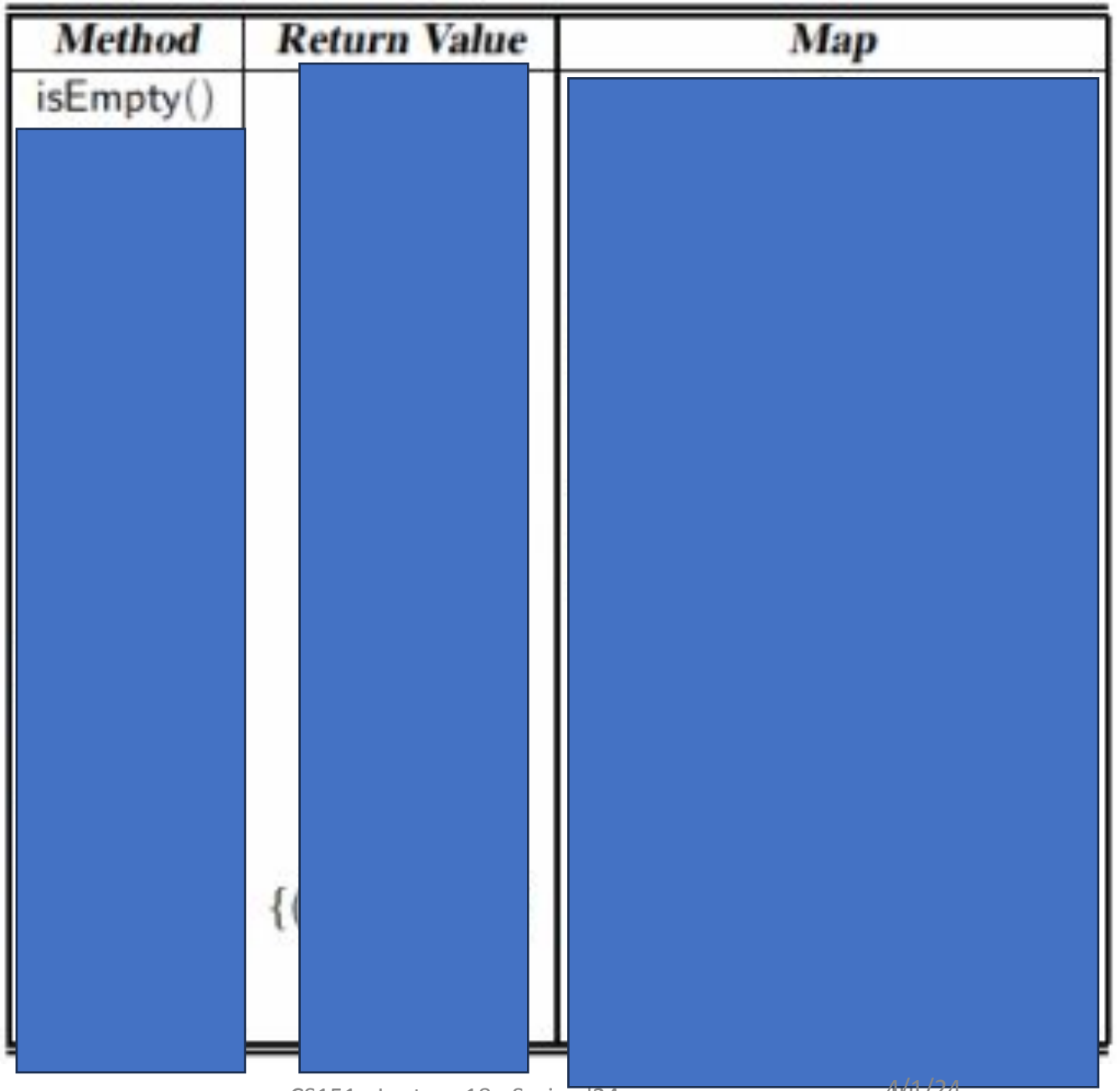
Map ADT

- `get(k)`: if the map `M` has an entry with key `k`, return its associated value; else, return null
- `put(k, v)`: insert entry `(k, v)` into the map `M`; if key `k` is not already in M, then return null; else, replace old value with `v` and return old value associated with `k`
- `remove(k)`: if the map `M` has an entry with key `k`, remove it from `M` and return its associated value; else, return null
- `size()`, `isEmpty()`
- `keySet()`: return an iterable collection of the keys in `M`
- `values()`: return an iterator of the values in `M`
- `entrySet()`: return an iterable collection of the entries in `M`

Example



Example



Example

Method	Return Value	Map
isEmpty()	true	

Example

Method	*Return Value*	*Map*
isEmpty()	true	{}

Example

Method	*Return Value*	*Map*
isEmpty()	true	{}
put(5,*A*)		

Example

Method	*Return Value*	*Map*
isEmpty()	true	{}
put(5,*A*)	null	

Example

Method	*Return Value*	*Map*
isEmpty()	true	{}
put(5,*A*)	null	{(5,*A*)}

Example

Method	*Return Value*	*Map*
isEmpty()	true	{}
put(5,*A*)	null	{(5,*A*)}
put(7,*B*)		
put(2,*C*)		
put(8,*D*)		
put(2,*E*)		
get(7)		
get(4)		
get(2)		
size()		
remove(5)		
remove(2)		
get(2)		
remove(2)		
isEmpty()		
entrySet()		
keySet()		
values()		

Example

Method	Return Value	Map
isEmpty()	true	{}
put(5,A)	null	{(5,A)}
put(7,B)	null	{(5,A),(7,B)}
put(2,C)	null	{(5,A),(7,B),(2,C)}
put(8,D)	null	{(5,A),(7,B),(2,C),(8,D)}
put(2,E)	C	{(5,A),(7,B),(2,E),(8,D)}
get(7)	B	{(5,A),(7,B),(2,E),(8,D)}
get(4)	null	{(5,A),(7,B),(2,E),(8,D)}
get(2)	E	{(5,A),(7,B),(2,E),(8,D)}
size()	4	{(5,A),(7,B),(2,E),(8,D)}
remove(5)	A	{(7,B),(2,E),(8,D)}
remove(2)	E	{(7,B),(8,D)}
get(2)	null	{(7,B),(8,D)}
remove(2)	null	{(7,B),(8,D)}
isEmpty()	false	{(7,B),(8,D)}
entrySet()	{(7,B),(8,D)}	{(7,B),(8,D)}
keySet()	{7,8}	{(7,B),(8,D)}
values()	{B,D}	{(7,B),(8,D)}

Map ADT

- Map class is abstract
- Concrete Implementations of Map:
  - `UnsortedTableMap`
  - `HashMap`

Map

- How can we implement a map?
  - Array !

Map.Entry Interface

- A (Key, Value) pair
- Keys and Values can be any reference type
- Methods:
  - `getKey()`
  - `getValue()`
  - `setValue(V val)`
- Implementation: `SimpleEntry`

ArrayMap

Let’s implement a `Map` as an array of `SimpleEntry`s

LinkedList Map

- `get(K key)`
- `put(K key, V value)`
  - If k is not in the map add it. If it is in the map, replace with the new value.
- `remove(K key)`

Performance Analysis

	Array	LinkedList
get		
put		
remove		

Performance Analysis

	Array	LinkedList
get	O(n)	O(n)
put	O(n)	O(n)
remove	O(n)	O(n)

HashMaps

Hash Functions

- A *hash function* maps an arbitrary length input to a fixed length *unique* output
- https://emn178.github.io/online-tools/sha256.html
- Applications
  - data structures
  - encryption / digital signatures
  - blockchain
- Properties of a good hash function:
  - one way
  - **collision resistant**
  - **uniformity**

Another Simple Hash Function

Given an int x...

h(x) = last 4 digits of x

- one way?
- collision resistant?
- uniform?

Another Simple Hash Function

h(x) = x % N

- one way?
- collision resistant?
- uniform?

HashMaps

- How can we use hash functions to improve the performance of our ArrayMap implementation?