Intro to Data Structures

Lecture #23 – Sets & Maps
November 18, 2014

Mark Stehlik

Outline for Today

- HW 5 grading (3 points back)!
- HW 6 questions? Due tonight…
- HW 7 out by tomorrow evening
- Quiz Thursday (sorting, stable sorts, trees, BST)
- Sets & Maps

Sets & Maps

- Introducing two new Abstract Data Types:
  - Set
    - an unordered collection of elements/objects
    - no duplicates allowed
  - Map (also called a dictionary or associative array)
    - a collection of key-value pairs; associates keys with values
    - keys must be unique, but different keys can map to the same value
    - a single key can also map to multiple values (how?)
    - you saw these in python (python dictionaries are maps)

Maps

- Examples of maps
  - Dictionary: maps words to definitions
  - Phone book: maps names to phone numbers
  - Advisor list: maps academic advisor to students
  - Class list: maps graduation year to list of students
  - Student schedule: maps student to list of courses
  - iTunes playlist: maps category to list of songs
  - Google: maps words to (a set of) web pages that contain the words

Distinguishing Lists, Sets, Map ADTs

- List
  - sequence of elements (not necessarily sorted), i.e., there's a first element, a second, etc. (an index)
  - allows duplicates
- Set
  - unordered, i.e., no notion of first, second
  - no duplicates allowed (what method enforces this?)
- Map
  - maps (unique) keys to values (but value can be a collection)

What operations defined for Sets/Maps?

- To the API…
- Set and Map are Interfaces (which implies?)

Set operations

- A set is just another collection, so…
  - isEmpty()
  - size()
  - add(value)
  - contains(value)
  - remove(value)
  - iterator (and iterable, so allows enhanced for loop)

Map operations

- A little more specialized…
  - isEmpty()
  - size() - number of key/value pairs
  - put(Key, value)
  - get(key)
  - containsKey(key)
  - remove(key)
  - keySet() - returns a set w/all the keys
  - iterator

Sets/Map implementations

- Two flavors of each…
  - TreeSet, HashSet
  - TreeMap, HashMap
- Tree implementations used when order (of keys) matters (and what we'll use today)
- Hash implementations for performance (more about hashing in the next lecture)
- To the demo…