CSSE 220 Day 19

Continue Data Structures Grand Tour

FixedLengthQueue

Markov Orientation and kickoff

CSSE 220 Day 19

- Turn in the written problem from HW 18.
- Before next class
  - Reading assignment from Weiss
  - Read and understand the Markov assignment
  - Take the short ANGEL Quiz about the contents of the Markov assignment documents.
- Questions?
- Continue the Data Structures Tour
- FixedLengthQueue Program
- Markov start-up

Some basic data structures

What is "special" about each data type?

What is each used for?

What can you say about time required for
- adding an element?
- removing an element?
- finding an element?

- Array (1D, 2D, …)
- Stack
- Queue
- List
  - ArrayList
  - LinkedList
- Set
- MultiSet
- Map (a.k.a. table, dictionary)
  - HashMap
  - TreeMap

You should be able to answer all of these by the end of this course.

Map

- A Table of key-value pairs.
- Insert and look up things by key.
- Implementations:
  - TreeMap
  - HashMap
- Same running time as the corresponding sets.

Java Map Example – HashMap

```java
HashMap<String, Integer> hm = new HashMap<String, Integer>();
hm.put("Mitt", 20);
hm.put("Mike", 85);
hm.put("John", 20);
hm.put("Rudy",  0);
hm.put("Alan", 95);
hm.put("Fred", 50);
int mikeValue = hm.get("Mike");
System.out.println("Value for Mike: " + mikeValue );
System.out.println("All entries in the HashMap:");
System.out.println(hm);
Collection values = hm.values();
System.out.println("Values: " + values);
Set keys = hm.keySet();
System.out.println("Keys: " + keys);
```

Note that the elements are not in Comparable order.

Output:

```
Value for Mike: 85
All entries in the HashMap:
{Mitt=20, Alan=95, Fred=50, John=20, Mike=85, Rudy=0}
Values: [20, 95, 50, 20, 85, 0]
Keys: [Mitt, Alan, Fred, John, Mike, Rudy]
```

Priority Queue:

- **Priority Queue:** Each item has an associated priority
  - Only the item with minimum priority is accessible.
  - Operations:
    `insert(add)`
    `findMin(peek)`
    `deleteMin(poll)`
  - Useful for simulations and for scheduling in an OS
  - Also in a famous Data Compression algorithm (230)
  - You will explore some implementations in the homework exercises later this week
  - Efficient implementation: binary heap (230)
    - findMin is O(1), the others are O(log N)

Graph

- A collection of vertices and edges
- Each edge joins two nodes (the two nodes may be allowed to be the same)
- Directed or undirected
- Graph Theory has been a subject of mathematical study for almost 3 centuries
- Example: Road map
- Example Diagram of links between web pages
- Find is O(N). Add, remove depend on implementation O(1), O(N), O(N^2)

Network

- A network is a graph whose edges have numeric labels
- Examples:
  - Road map (mileage)
  - Airline's flight map (flying time)
  - Plumbing system (gallons per minute)
  - Computer network (bits/second)
- Famous problems:
  - Shortest path
  - Maximum flow
  - Traveling salesman

Some basic data structures

What is "special" about each data type?

What is each used for?

What can you say about time required for
- adding an element?
- removing an element?
- finding an element?

- Array (1D, 2D, …)
- Stack
- Queue
- List
  - ArrayList
  - LinkedList
- Set
- MultiSet
- Map (a.k.a. table, dictionary)
  - HashMap
  - TreeMap
- PriorityQueue
- Tree
- Graph
- Network

You should be able to answer all of these by the end of this course.

Markov Chain Progam

- Input: a text file

the skunk jumped over the stump
the stump jumped over the skunk
the skunk said the stump stunk
and the stump said the skunk stunk

- Processing:
  - Gather word pattern statistics
  - Store them in an appropriate data structure
  - Output text that follows the patterns

- Output: a randomly-generated text file with many of the same properties as the original file
  - Fully justified, of course ☺

Markov

- Input: a text file

the skunk jumped over the stump
the stump jumped over the skunk
the skunk said the stump stunk
and the stump said the skunk stunk

Statistics (n=1):

NONWORD	the
the	skunk (4), stump (4)
skunk	jumped, said, stunk, the
jumped	over (2)
over	the (2)
stump	jumped, said, stunk, the
said	the (2)
stunk	and, NONWORD
and	the

Markov

- Input: a text file

the skunk jumped over the stump
the stump jumped over the skunk
the skunk said the stump stunk
and the stump said the skunk stunk

Statistics (n=2):

NW NW	the
NW the	skunk
the skunk	jumped, said, the, stunk
skunk jumped	over
jumped over	the
over the	stump, skunk
the stump	the, jumped, stunk, said
…	

Output

- n=1:

```
the skunk the skunk jumped over the skunk stunk

the skunk stunk
```

- n=2:

```
the skunk said the stump stunk and the stump jumped over the skunk jumped over the skunk stunk
```

- Note: it's also possible to hit the max before you hit the last nonword.

Full Justification

- Do this step LAST
- Output needs to be full-justified (as on the Output slide)
- You are required to use lists (Array and Linked) to hold the output line and to make it easier to modify the line (by adding extra spaces) before you print it

- Demo

Markov Data structures

For the prefixes?

For the set of suffixes?

To relate them?

Statistics (n=2):

NW NW	
NW the	
the skunk	
skunk jumped	
jumped over	
over the	
the stump	
…	

Fixed-length Queue and Markov

- FixedLengthQueue: a specialized data structure.
- Useful for Markov problem.
- You and your Markov partner should implement it in the next 25 minutes or so.
- Put both people's names in a comment at the top of your program file. Submit to one person's repository.
- **Then** read (twice) and begin digesting the Markov assignment.
- Discuss it with your partner.
- Plan when you will meet today to continue the discussion and get started on the program.