

CSSE 220 Day 27

Data Structures Practice

Checkout *DataStructures* project from SVN

Questions

Common ADTs

- ▶ Array List
- ▶ Linked List
- ▶ Stack
- ▶ Queue
- ▶ Set
- ▶ Map

Implementations for all of these are provided by the **Java Collections Framework** in the **java.util** package. We'll practice using them.

Array Lists and Linked Lists

Operations Provided	Array List Efficiency	Linked List Efficiency
Random access	$O(1)$	$O(n)$
Add/remove item	$O(n)$	$O(1)$

Finish ListIterator **delete** together now

Stacks

- ▶ A last-in, first-out (LIFO) data structure
- ▶ Real-world stacks
 - Plate dispensers in the cafeteria
 - Pancakes!
- ▶ Some uses:
 - Tracking paths through a maze
 - Providing “unlimited undo” in an application

Operations Provided	Efficiency
Push item	$O(1)$
Pop item	$O(1)$

Implemented by
Stack, **LinkedList**,
and **ArrayDeque** in
Java

Queues

- ▶ A first-in, first-out (FIFO) data structure
- ▶ Real-world queues
 - Waiting line at the BMV
 - Character on Star Trek TNG
- ▶ Some uses:
 - Scheduling access to shared resource (e.g., printer)

Operations Provided	Efficiency
Enqueue item	$O(1)$
Dequeue item	$O(1)$

Implemented by
LinkedList and
ArrayDeque in Java

Sets

- ▶ Collections **without duplicates**
- ▶ Real-world sets
 - Students
 - Collectibles
- ▶ Some uses:
 - Quickly checking if an item is in a collection

Operations	HashSet	TreeSet
Add/remove item	$O(1)$	$O(\lg n)$
Contains?	$O(1)$	$O(\lg n)$

Can hog space

Sorts items!

Q3

Maps

- ▶ Associate **unique keys** with **values**
- ▶ Real-world “maps”
 - Dictionary
 - Phone book
- ▶ Some uses:
 - Associating student ID with transcript
 - Associating name with high scores

Operations	HashMap	TreeMap
Insert key-value pair	$O(1)$	$O(\lg n)$
Look up value for key	$O(1)$	$O(\lg n)$

Can hog space

Sorts items by key!

Markov Chaining

»» Demonstration

Markov Chain Program

- ▶ 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

- ▶ Output: a randomly generated list of words that is “like” the original input in a well-defined way

Markov Chain Process

- ▶ Gather statistics on word patterns by building an appropriate data structure
 - ▶ Use the data structure to generate random text that follows the discovered patterns
- 

Work Time

- ▶ Review HW description
- ▶ What questions do you have?

- ▶ Use the remainder of the class to work on Vector Graphics with your team.