

CSSE 220 Day 27

Data Structures Overview/Practice
FixedLengthQueue
Introduction to Markov Chaining

Checkout *DataStructures* project from SVN

Questions

Juggling Mini-project and Markov

▶ It's nothing new – Curt did it last term.

25 Mon Nov 3 Details	<ul style="list-style-type: none"> • Big Java, Ch. 14 • Bubble Sort • Selection Sort 	• HW 22	<ul style="list-style-type: none"> • Sorting and searching • Algorithm analysis • Introduce mini-project • Mini-project work time 	<ul style="list-style-type: none"> • Slides • Dam Beavers Spec. 	Mini-project	
26 Wed Nov 5 Details	• Big Java, §15.1–15.3	• HW 23	<ul style="list-style-type: none"> • Sorting wrap-up • Function objects • Intro. to linked lists • Vector graphics demos 	• Slides	Mini-project	
27 Thu Nov 6 Details	• Big Java, §15.4, §16.1–16.4	• HW 24	<ul style="list-style-type: none"> • Linked list implementation • Abstract data types • Data-structure-palooza: stacks, queues, sets, maps 	• Slides	Mini-project	
28 Mon Nov 10 Details	• Big Java, §16.5–16.7	• HW 25	<ul style="list-style-type: none"> • Data structure implementation sketches • Intro. to Markov Chaining 	<ul style="list-style-type: none"> • Slides • FixedLengthQueue • Markov 	Mini-project	Markov Part 1
29 Wed Nov 12 Details	• Big Java, Ch. 17	• HW 26	• Project presentations	• Slides	Mini-project	Markov Part 1
30 Thu Nov 13 Details	• None	<ul style="list-style-type: none"> • HW 27 • (Markov Milestone 1 due <i>Friday</i> at midnight) 	<ul style="list-style-type: none"> • Generic programming • Type variables • Generic methods • Course evaluations • Exam review 	• Slides	Markov Part 1	Final Exam Monday 6pm in G308

25 Mon Feb 9 Details	<ul style="list-style-type: none"> • Big Java, §15.2–15.3 • Selection Sort • Insertion Sort 	• HW 23	<ul style="list-style-type: none"> • Exam 2 problem on Vector Graphics • Vector Graphics Demos • Student Questions • Sorting • Searching 	• Slides	Mini-project	
26 Wed Feb 11 Details	• Big Java, §15.4, §16.1–16.2	• HW 25	<ul style="list-style-type: none"> • Sorting wrap-up • Data Structures and Abstract Data Types • Array implementation (1D, 2D) 	<ul style="list-style-type: none"> • Slides • BinaryInteger assignment 	Mini-project	
27 Thu Feb 12 Details	• Big Java, §16.3–16.5	• HW 26	<ul style="list-style-type: none"> • Java Collections Framework • Collection and List Interfaces; iterators • ArrayList implementation • Linked List Implementation, part 1. • Linked list implementation • Abstract data types • Data-structure-palooza: stacks, queues, sets, maps 	• Slides	Mini-project	Markov Part 1
28 Mon Feb 16 Details	• Big Java, §16.6–16.8	• HW 27	<ul style="list-style-type: none"> • Data structure implementation sketches • Intro. to Markov Chaining 	<ul style="list-style-type: none"> • Slides • FixedLengthQueue • Markov 	Mini-project	Markov Part 1
29 Wed Feb 18 Details		• HW 28	• Project presentations	• Slides	Mini-project	Markov Part 1
30 Thu Feb 19 Details	<ul style="list-style-type: none"> • HW 29 • (Markov due <i>Friday</i> at midnight) 		<ul style="list-style-type: none"> • Generic programming • Type variables • Generic methods • Course evaluations • Exam review 	• Slides	Markov Part 1	Final Exam Wed 6 PM

Most Minesweeper work will be out of class.

Most Markov work in-class

Markov vs. Minesweeper teams

- ▶ Markov has usually been a pair assignment (and that's probably ideal).
- ▶ But I don't want you to have to flit between two teams, so I am having you do Markov with your same team.
- ▶ This is potentially dangerous:
 - Be sure that you and the rest of your team understand everything that is going on.

More details on Markov later today.



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.

Array List and Linked List

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

Next Week we will discuss implementations.

Stack

- ▶ A last-in, first-out (LIFO) data structure
- ▶ Real-world stacks
 - Plate dispensers in the cafeteria
 - Pez dispenser
- ▶ 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

Queue

- ▶ A first-in, first-out (FIFO) data structure
- ▶ Real-world queues
 - Waiting line at the BMV
 - Waiting for customer service at Dell.com
- ▶ Some uses:
 - Scheduling access to shared resource (e.g., printer)

Can implement as a (growable) "circular" array
<http://maven.smith.edu/~streinu/Teaching/Courses/112/Applets/Queue/myApplet.html>

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

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

Set

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

But the Java **TreeSet** implementation of the **Set** interface does keep its items ordered.

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

Can hog space

Sorts items!

Q1

Map

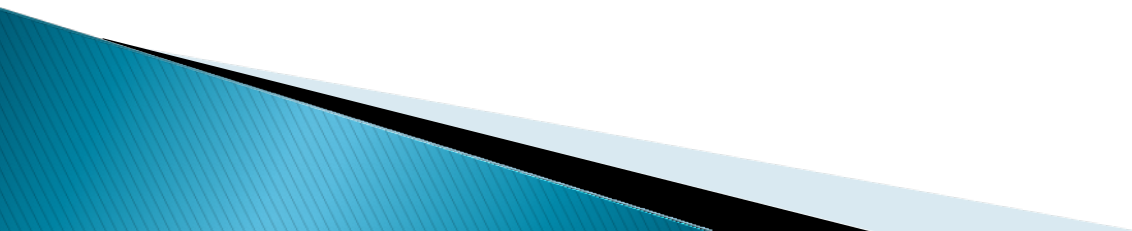
- ▶ Associate **keys** with **values**
- ▶ Real-world “maps”
 - Dictionary
 - Phone book
- ▶ Some uses:
 - Associating student ID with transcript
 - Associating name with high scores

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!

Data Structures Exercise

- ▶ In the DataStructures Project
 - ▶ Do the ToDo items
 - ▶ Talk with someone else
 - ▶ Get help from instructor/TA
 - ▶ If you don't finish now, finish for HW.
- 

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

Markov Example, $n = 1$

► 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

Prefix	Suffixes
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 Example, $n = 2$

▶ 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

Prefix	Suffixes
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

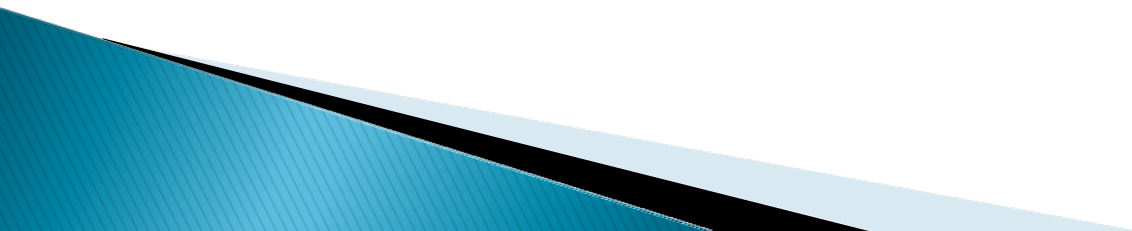
▶ Note: it's also possible to hit the max before you hit the last nonword.

Markov Data structures

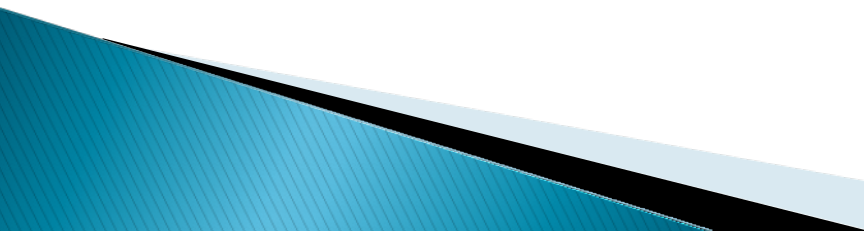
- ▶ For the prefixes?
- ▶ For the set of suffixes?
- ▶ To relate them?

Prefix	Suffixes
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
...	

Markov Pairs

- ▶ The Markov and FixedLengthQueue projects are in your team repository:
 - ▶ Check them out.
- 

Fixed-Length Queue and Markov

- ▶ FixedLengthQueue: a specialized data structure, useful for Markov problem
 - ▶ Work with your team to implement it in the next 25 minutes or so.
 - ▶ Then read (twice) and begin digesting the Markov assignment
 - ▶ It's in several inter-linked documents
 - ▶ Discuss it with your team
- 

Work Time

- »» Review HW description,
Teams work on
FixedLengthQueue/Markov for
rest of class