

## Abstract Data Types

- Boil down data types (e.g., lists) to their essential operations
- Choosing a data structure for a project then becomes:
  - Identify the operations needed
  - Identify the abstract data type that most efficiently supports those operations
- Goal: that you understand several basic abstract data types and when to use them

### **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 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)

### **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, Li nkedLi st, 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)

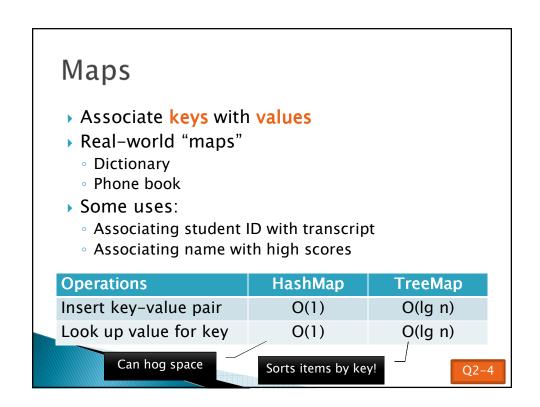
Operations Provided	Efficiency
Enqueue item	O(1)
Dequeue item	O(1)

Implemented by Li nkedLi st and ArrayDeque in Java

#### Sets Unordered collections without duplicates Real-world sets Students Collectibles Some uses: · Quickly checking if an item is in a collection HashSet **Operations** TreeSet Add/remove item 0(1) O(lg n) Contains? $O(lg_n)$ O(1)

Sorts items!

Can hog space





**Team** URL (Individuals use individual repositories) http://svn.csse.rose-hulman.edu/repos/csse220-201130-markov-teamXX

- Curt's section teams:
  - 11,caijy,filhobc
  - 12,hirtjd,spurrme
  - 13,luok,shanx
- Curt's section individuals:
  - addantnb, chena1, cornetcl, eckertzs, elswicwj, hopwoocp, lyonska, nelsonca, taos, wilsonam

**Team** URL (Individuals use individual repositories) http://svn.csse.rose-hulman.edu/repos/csse220-201130-markov-teamXX

- Delvin's section teams:
  - 21, solorzaa, whitemrj
  - 22, hazelrtj, tilleraj
  - 23, haydr, lawrener
  - 24, myersem, rybickcb
  - 25, mehrinla, vassardmzhenw
  - 26, cooperdl,fengk

- Delvin's section Individuals:
  - amesen
  - finnelhn
- oliverr
- senatwj

# 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

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
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

 $\rightarrow$  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

### Fixed-Length Queue and Markov

- FixedLengthQueue: a specialized data structure, useful for Markov problem
- Implement FLQ in the next 25 minutes or so
- When you finish, read the (long) Markov description and start working on it
- We will only do milestone 1 (so no text justification)

Check out FixedLengthQueue from your Markov team or individual repo