# Linked Lists and Priority Queues – A Taste of 214

CSE 114 INTRODUCTION TO OBJECT-ORIENTED PROGRAMMING

1

## Announcements

Today: Some 'data structures'



2

## Data Structures

Efficient solutions to problems regard organizing data in ways that make it easy to access and easy to process

The study of data structures helps you design data organizations that allow for efficient solutions.

You have already seen a type of data structure! Arrays are a very simple data structure



3

## Linked Lists

Linked lists hold a series of data items like an array

Instead of sequential storage, they use individual objects each holding a 'reference' to the next object in the list

This structure has a 'head' and a 'tail' reference to the first and last items in the list



4

# Nodes

Linked Lists, queues, trees, etc need component objects to hold data and references to other items.

These are called 'Nodes'. - See Node.java

Simple Node:

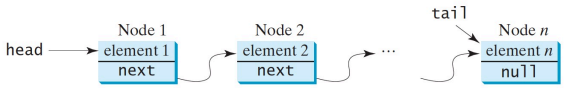
```
public class Node {

   public String data;
   public Node next;   // Reference to next node

  Node (String s) {
     data = s;
     next = null;
   }
}
```

5

5

# Linked Lists

See LinkedList.java

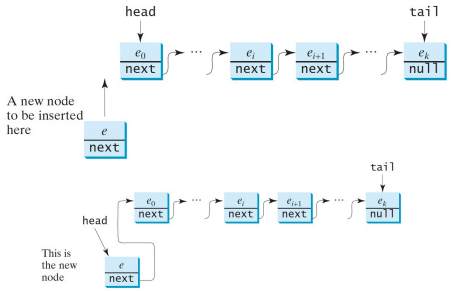



6

6

# Linked List Operations

Insert (First Element):

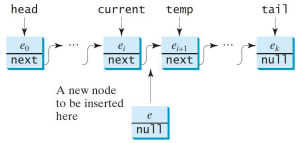



7

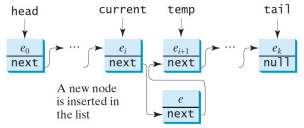
7

# Linked List Operations

Insert (Arbitrary Element):




Order of Operations is important!

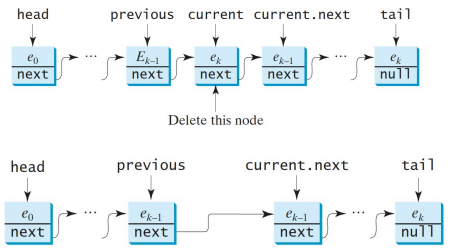



8

8

## Linked List Operations

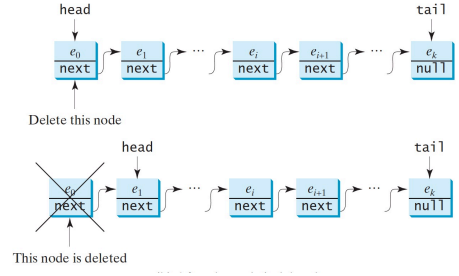
Delete (Arbitrary Node):




9

## Linked List Operations

Delete (Head Node):




10

## Priority Queue

Priority Queues can hold data in a 'priority based' order

We can decide what the criteria is

Example: Keep a list of words in a sorted order

See PQueue.java

11

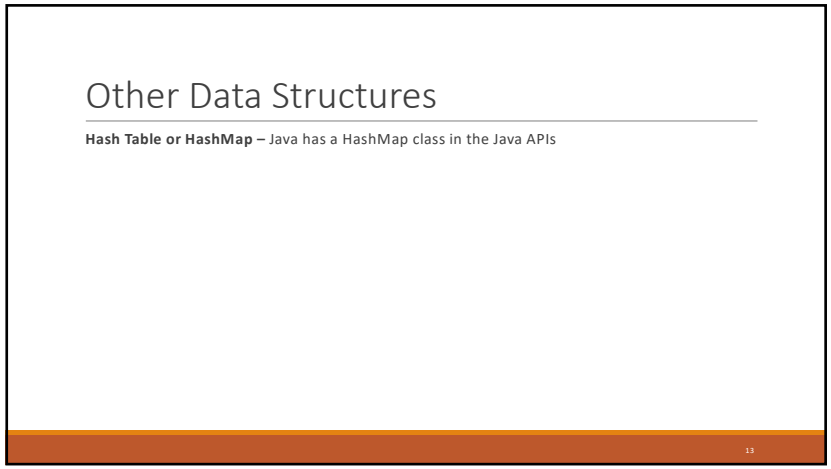
## Other Data Structures

**Stack –** Collection of data organized so that the last item added is the first to be removed
- 'LIFO' – Last-in, First-out
- Operations – (Push, Pop)

**Queue –** Collection of data organized so that the first item added is the first to be removed
- **'FIFO' –** First-in, First-out
- Operations – Insert (Adds Node at the 'tail' of the queue), Remove (Removes Node from the 'head' of the queue)

**Binary Tree** – Keeps 'Nodes' organized based on a particular requirement
- Sort Order
- Position in an arithmetic expression
- Many other options…

12


Other Data Structures
Hash Table or HashMap – Java has a HashMap class in the Java APIs
13

13