**Administrative**

**Today’s session**

Data structure

Stack

Queue

Priority queue

Homework 3

**Session Topics**

**Data structure**

● A **data structure** is an application-time construct used to store and retrieve data.

● A data structure that holds one value is **scalar**.

● A data structure that holds multiple values is **compound**.

● An **abstract data structure**, or abstract data type, defines the arrangement of the data and operations on the data from the perspective of the *user*.

● A **concrete** **data structure**, or data structure, defines the arrangement of the data and operations on the data from the perspective of the *developer*.

● There are several data structures including:

|  |  |  |  |
| --- | --- | --- | --- |
| Data structure | Scalar or compound? | Concrete or abstract? | Concrete data structures that may be used to implement |
| Variable | Scalar | Concrete | NA |
| Array1 | Compound | Concrete | NA |
| Linked list2 | Compound | Concrete | NA |
| Record | Compound | Concrete | NA |
| Class | Compound | Concrete | NA |
| Stack | Compound | Abstract | Array  Linked list |
| Queue | Compound | Abstract | Array  Linked list |
| Priority queue | Compound | Abstract | Array  Linked list |
| Tree3 | Compound | Abstract | Array  Linked list |
| Map | Compound | Abstract | Array  Linked list |
| Graph | Compound | Abstract | Array  Linked list |

1 In Java, this includes the ArrayList.

2 This includes linking from one node to one or more other nodes.

3 This includes n-ary trees and heaps.

**Stack**

● A **stack** is a data structure that stores data as a series of linked nodes.

● A stack may contain zero or more linked nodes.

● A node is a packet of information containing:

✓ At least one datum.

✓ A link to the next node in the stack, or **null** if the node is the bottom of the stack.

● The link to the first node of the stack is called the **top**.

● The top is the only connection between the stack and the outside world.

● A stack is considered last-in first-out (LIFO).

● There are two possible operations on a stack:

✓ Push – add a node to the top of the stack.

✓ Pop – if the stack is not empty, remove a node from the top of the stack.

● Stack applications include:

✓ In an operating system, for holding recursive and non-recursive method calls.

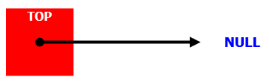
✓ In a travel application, for holding GPS coordinates for a route.

✓ In a maze application, for holding previous moves.

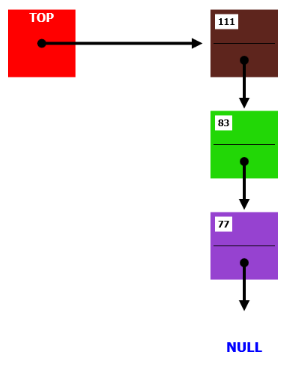
✓ In a calculator application, for holding lower-precedence expressions while evaluating higher-precedence expressions.

✓ In a compiler, for parsing statements.

● Here is an empty stack:



● Here is a stack with three nodes:



● A visualization of the stack is available at [www.cs.usfca.edu/~galles/visualization/StackLL.html](http://www.cs.usfca.edu/~galles/visualization/StackLL.html)

● See **Stack using array** sample application on Blackboard.

**Queue**

● A **queue** is a data structure that stores data as a series of linked nodes.

● A queue may contain zero or more linked nodes.

● A node is a packet of information containing:

✓ At least one datum.

✓ A link to the next node in the queue, or **null** if the node is the end of the queue.

● The link to the first node of the queue is called the **head**.

● The link to the last node of the queue is called the **tail**.

● The head and the tail are the only connections between the queue and the outside world.

● A queue is considered first-in first-out (FIFO).

● There are two possible operations on a queue:

✓ Insert – add a node to the tail of the queue.

✓ Remove – if the queue is not empty, remove a node from the head of the queue.

● Queue applications include:

✓ In an operating system, for holding the series of tasks awaiting a time slice with the CPU.

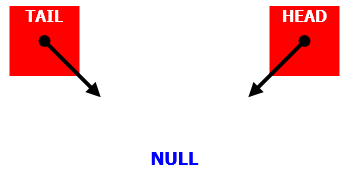
✓ In an airline application, for holding a series of flights ready for takeoff.

✓ In a UPS application, for holding a series of deliveries.

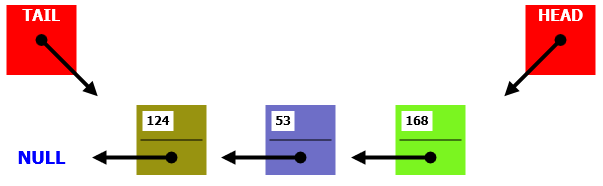
✓ In a keyboard buffer, for holding a series of characters.

✓ In a media player, for holding a series of buffered sound.

● Here is an empty queue:



● Here is a queue with three nodes:



● A visualization of the stack is available at

[www.cs.usfca.edu/~galles/visualization/QueueLL.html](http://www.cs.usfca.edu/~galles/visualization/QueueLL.html)

● See **Queue using array** sample application on Blackboard.

**Priority queue**

● A **priority queue** is a data structure that stores data as a series of linked nodes.

● A priority queue may contain zero or more linked nodes.

● A priority queue is similar to a regular queue but the node to be removed next is the one assigned the highest priority.

● The priority is determined by the developer and/or user and assigned based on the value(s) stored in a node.

● There are two possible operations on a priority queue:

✓ Insert – add a node to the priority queue while optionally satisfying the priority requirement.

✓ Remove – if the priority queue is not empty, remove a node from the head of the priority queue.

● There are several ways to implement a priority queue including:

|  |  |  |
| --- | --- | --- |
| Method | Insert | Remove |
| Ordered array or linked list | Search for a spot to insert the node such that the priority requirement is maintained. | Remove the node at the head (same as regular queue). |
| Unordered array or linked list | Add the node at the tail (same as regular queue). | Search for the node with the highest priority and remove it. |
| Heap | Add the node to the bottom of the tree and filter it up until its correctly positioned. | Remove the node at the root. Move the node from the bottom of the tree to the root and filter it down until its correctly positioned. |

● Priority queue applications include:

✓ In an operating system, for scheduling jobs based on their priority.

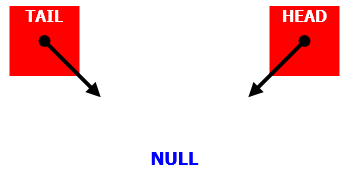
✓ In an operating system, for handling interrupts.

✓ In a database application, for processing synchronous and asynchronous requests.

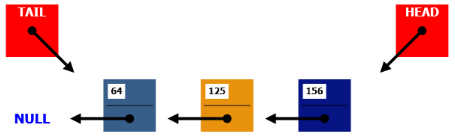
✓ In a UPS application, for holding a series of deliveries in geographic order.

✓ In an online application, for processing preferred and standard customers.

● Here is an empty priority queue using the ordered-linked-list method:



● Here is a priority queue using the ordered-linked-list method with three nodes:



● See **Priority queue using array** sample application on Blackboard.

**Homework 3**

● Assigned today.

● Available on Blackboard.

● Due in one week.