CS 367 - Introduction to Data Structures
Tuesday, October 11, 2016

Midterm Exam 1
- Tuesday, October 18, 5:00 pm – 7:00 pm
- Lec 1: room 105 Psychology
- Lec 2: room 272 Bascom Hall
- Lec 3: room 132 Noland Hall
- UW ID required
- See posted Exams page for more information (and link to sample questions doc)

Homework 5 due 10 pm Friday, October 14th
Homework 1 & 2 Grades available – contact grader by 10/25 if you have questions
Program 2 due 10 pm Sunday, October 23rd – SHOULD BE TESTING PARTS

Last Time
Complexity
- best/worst cases
- analyzing Java code (from last time)
- practice analyzing Java code (from last time)
- significance of scaling
Comparing ArrayList vs LinkedList

Today
Comparing ArrayList vs LinkedList (from last time)
Complexity Caveats (from last time)
Shadow Array - improving array resizing
Stack ADT
- concept
- array implementations
- chain of nodes implementations
Queue ADT
- concept
- chain of nodes implementations

Next Time
Read: finish Stacks and Queues, Trees Intro., start Priority Queues
Circular Array Data Structure
Tree Terms
Priority Queue ADT
- concept
- operations
- implementation options
Comparing ListADT Implementations

Space Requirements → Problem size N is?

Array:

Singly-Linked List:

Circular Singly-Linked List:

Doubly-Linked List:

Circular Doubly-Linked List:

Ease of Implementation

Array:

Singly-Linked List:

Circular Singly-Linked List:

Doubly-Linked List:

Circular Doubly-Linked List:
Returning N Papers to N Students

problem size (N) =
dominant operation =

→ What is the complexity of each algorithm below?

Algorithm 1:
call out each name,
have student come forward & pick up

  best-case:
  worst-case:

Algorithm 2:
hand pile to first student,
student linearly searches through papers & takes hers/his,
pass pile to next student who does likewise

  best-case:
  worst-case:

Algorithm 3:
sort the papers alphabetically,
hand pile to first student who does binary search,
pass to next student who does likewise
Shadow Array – Improving Array Resizing

"Naïve" Approach

"Shadow Array" Improvement
Stack ADT

Concept

Operations

Implementing using an Array

Where should the top be located in the array?

Implementing using a Chain of Nodes

Where should the top be located in the chain of nodes?

Complexities
Queue ADT

Concept

Operations

Implementing a using a Chain of Nodes

→ Is one option better than the other?

Option 1: front of queue is at head, rear of queue is at tail

Option 2: front of queue is at tail, rear of queue is at head
Implementing a Queue ADT using an Array

Assume a shadow array is used so that expand is O(1).

**Option 1:** front of queue is at _______________, rear of queue is at _______________

- \[ \begin{array}{cccccc} \text{ } & \text{ } & \text{ } & \text{ } & \text{ } & \text{ } \\ \text{ } & \text{ } & \text{ } & \text{ } & \text{ } & \text{ } \\ \end{array} \]

**Option 2:** front of queue is at _______________, rear of queue is at _______________

- \[ \begin{array}{cccccc} \text{ } & \text{ } & \text{ } & \text{ } & \text{ } & \text{ } \\ \text{ } & \text{ } & \text{ } & \text{ } & \text{ } & \text{ } \\ \end{array} \]

**Option 3:** front of queue is at _______________, rear of queue is at _______________

- \[ \begin{array}{cccccc} \text{ } & \text{ } & \text{ } & \text{ } & \text{ } & \text{ } \\ \text{ } & \text{ } & \text{ } & \text{ } & \text{ } & \text{ } \\ \end{array} \]