Exam 2 Review

CS 3358
Spring 2015

Jill Seaman

Exam 2

- Monday, April 20
- In class, closed book, closed notes, clean desk
- 20% of your final grade
- 80 minutes
- I recommend using a pencil (and eraser)
- All writing will be done on the test paper I will hand out.
- No calculators, cell phones, or head phones.

Exam Format

- 100 points total
  - Writing programs/functions/templates/code (about 50-60%), some recursive
  - Multiple choice
  - Fill-in-the-blank/short answer (analysis of sorting)
  - Demonstrating stack and queue operations (what is output), sorts and sort operations (show contents of the array
  - Finding errors in code (recursive functions)

Templates (w7)

- Why? What are they for?
  - Type independence, generic programming
- Templated Functions
- Templated Classes
  - Everything goes in the .h file
- Be familiar with the examples
  - swap, MemoryCell, (my) vector
- Be prepared to write code, convert existing function or class to template

Stack ADT (w8)

- Know the operations, how they work
  - O(1): push, pop, isFull, isEmpty
  - makeEmpty
- Be able to implement using an array or (singly) linked list or vector or . . .
- Be able to use a stack to solve a problem
- Be familiar with the sample code:
  - IntStack: the static stack class in the notes
  - stack_3358_LL.h: the linked-list implementation on the website
- Array vs Linked List implementations

Queue ADT (w8)

- Know the operations, how they work
  - O(1): enqueue, dequeue, isFull, isEmpty
  - makeEmpty
- Be able to implement using an array (with wraparound) or singly linked list or vector or . . .
- Be able to use a queue to solve a problem
- Be familiar with the sample code:
  - IntQueue: the static queue class in the notes
  - queue_3358_LL.h: the linked-list implementation on the website
- Array vs Linked List implementations

Recursion (w9)

- How to write recursive functions
  - Base case
  - Recursive case (smaller caller)
- Recursion over non-negative ints and lists
  - arrays, vectors, linked list, string (substr)
- Know what's wrong with recursive Fibonacci
- Binary Search: understand recursive version and run-time analysis
- You will be asked to write one or two recursive functions.

Sorting (w10)

- Understand the different sorts:
  - $O(N^2)$: selection, insertion, bubble
  - O(N log N): merge sort, quicksort
- Know the algorithms really well
  - Will not have to write code for merge or partition
  - Be able to write code for one sort (pick one)
  - **Will** be asked to show steps in the process (show result of a pass, or a merge, or a partitioning).
- Be familiar with runtime analyses and issues

Hash tables (w11)

- Hash tables and (good) hash functions
- Collisions and collision resolution
  - Linear probing
    - Lazy deletion
  - Quadratic probing
  - Separate chaining (pros+cons)
- Rehashing: how to expand the table
- Be able to hash a list of keys given a simple hashing function and collision strategy
  - Like the examples in the slides (18+23)

Exercises: stacks+queues

Given the ADT for the Stack_3358 at the end of the exam, implement the push, pop, isEmpty and isFull functions.

The class declaration would either:
a) include the private member variables or else
b) the question would state which implementation to use and you would provide the private member variables

Given a set of operations over an empty stack (or queue) show the contents of the stack (or queue) after the operations are completed (see Quiz 3 question 5).

Exercises: Sorting

Given the following array, what would be the contents
a) after the 3rd iteration of the insertion sort (4 elements are sorted)?
b) after the 4th iteration of the selection sort?
c) after one iteration of the bubble sort?

```
3  7  2  12 56  1  42  9
```

Given the array above, show the contents of the array after (any) valid partitioning method has been applied that uses the first element as the pivot. Indicate which element is the pivot in your result.

Exercises: Sorting

Give the code for the main merge sort (or quicksort) functions? Assume the merge (or partition) functions are already written.

Given the following two arrays, show the contents of the temp array and the positions of i and j after 3 passes of the merge algorithm.

```
 1 9 42 56
```

```
2 3  7 12
```

Given the following array, show the contents after a valid partitioning using the first element as the pivot point.

```
7  3  2  12 56  1  42  9
```

How to Study

- Review the slides for each lecture
  - understand all the concepts, quiz yourself!
- Use the book to help understand the slides
  - there will be no questions over material (or code) that is in the book but not on the slides
- Understand the code in the demo(s)
- Understand the homework assignment solutions
  - rewrite yours so it works
  - Practice, practice, practice
- Get some sleep