CS 445: Data Structures
Final Examination: Study Guide

• Java prerequisites
   Topics:
   – Classes, objects, and references
   – Access modifiers
   – Arguments and parameters
   – Garbage collection
   Self-test questions: Appendix C

• Designing classes
   Topics:
   – Composition and inheritance
   – Access modifiers
   – static keyword
   – Overriding methods
   – Dynamic binding and method polymorphism
   – Reference type vs. object type
   – Interfaces
   – Typecasting
   – Generic interfaces, classes, and methods
   – Generic type declarations, including bounded types, type wildcards, and bounded wildcards
   – Assertions

   Self-test questions: Appendix D; Prelude; Java Interludes JI1, JI3
   Exercises:
   – Prelude: 1, 2
• Bag

Note: Since this was our first data structure, we covered several topics in this section even though they weren’t specific only to bags.

Topics:

– ADTs, collections, data structures and their relation to interfaces and classes
– Client vs. implementer
– Considering corner cases
– Test methods
– Bag ADT and interface
– Bag array vs. linked implementations
– Resizing arrays
– Inner classes, static and non-static

Self-test questions: Chapters 1–3
Exercises:

– Chapter 1: 1, 5
– Chapter 2: 1, 5, 6, 11
– Chapter 3: 1, 5, 12, 14

• Algorithm analysis

Topics:

– Asymptotic analysis
– Big-O notation
– Growth rates
– Amortized analysis
– Sum of the first $n$ integers
– Analysis of bag implementations

Self-test questions: Chapter 4
Exercises:

– Chapter 4: 1–6, 10–12, 17
• Stack
  Topics:
  – Stack interface
  – Using stacks to match brackets
  – Using stacks to evaluate postfix
  – Using stacks to convert infix to postfix
  – Array vs. linked implementations of stack (runtime, memory usage)
  – Program stack / run-time stack

Self-test questions: Chapters 5, 6
Exercises:
  – Chapter 5: 1, 3, 6–8
  – Chapter 6: 1, 3, 5, 8, 9

• Recursion
  Topics:
  – Breaking problems into subproblems
  – Requirements for recursion to work
  – Activation records
  – Divide & conquer vs. general recursion
  – Tail recursion
  – Easy vs. hard recursive algorithms to make iterative
  – Overheads of recursion
  – Recursive backtracking
    * General goals
    * The specific structure we used (next, extend, isFullSoln, reject)
  – Analyzing recursive methods with recursion trees
  – Processing arrays recursively by specifying bounds of subarray

Self-test questions: Chapters 7, 18
Exercises:
  – Chapter 7: 1, 2, 5, 8, 16
  – Chapter 18: 4, 6, 7

3
• Sorting
Topics:
  – Simple sorts
    * Selection sort
    * Bubble sort
    * Insertion sort
  – Shell sort
  – Divide & conquer for sorting
    * Merge sort
    * Quicksort (including effect of pivot, pivot selection techniques)
    * Base cases
  – Sorting in-place vs. memory overhead
  – Stable sorting
  – Runtime analysis of sort methods

Self-test questions: Chapters 8, 9
Exercises:
  – Chapter 8: 1–3, 11, 13
  – Chapter 9: 1–4, 9

• List
Topics:
  – List ADT and uses
  – Array vs. linked implementations of List (runtime, memory usage)
  – Circular linked list
  – Doubly-linked list
  – Average case analysis

Self-test questions: Chapters 12–14
Exercises:
  – Chapter 12: 1, 2, 4, 5, 7, 8
  – Chapter 13: 1, 8, 9, 11
  – Chapter 14: 1, 2, 4, 15
• Iterator
  
  Topics:
  
  – Reasons for iterators
  – Iterator<T>, which classes should implement it, and why
  – Iterable<T>

  Self-test questions: Java Interlude 5, Chapter 15

  Exercises:
  
  – Chapter 15: 1–3, 10

• Tree and Binary Search Tree
  
  Topics:
  
  – Motivations for trees
  – Tree terminology
  – Recursive definitions of tree, binary tree, and BST
  – Developing tree operations recursively
  – Full, complete, balanced trees
  – Tree node classes
  – Traversals: Depth-first and breadth-first
  – Tree iterators, general strategy, reasons for the approach
  – BST operations and how they work
  – Analysis of BST operations

  Self-test questions: Chapters 23–25

  Exercises:
  
  – Chapter 23: 2–4, 6–8
  – Chapter 24: 4, 6, 7, 9
  – Chapter 25: 1, 2, 3, 5, 6, 7, 15
• Queue and Priority Queue

Topics:

– Queue ADT
– Linked Queue implementation
– Two-part circular linked Queue implementation
– Logically-circular array Queue implementation
– Analysis of Queue implementations
– PQ ADT
– Array implementations of PQ
– Heap implementations of PQ
– Analysis of PQ implementations

Self-test questions: Chapters 10, 11, 26

Exercises:

– Chapter 10: 1, 3, 5
– Chapter 11: 3–6, 10
– Chapter 26: 1, 2, 5