600.226: Data Structures
Midterm 2

Peter H. Fröhlich
phf@cs.jhu.edu

November 11, 2005
Time: 40 Minutes

Start here: Please fill in the following important information using a **permanent pen** before you do **anything** else! Your exam will **not** be graded if you use a pencil or erasable ink on this page.

Name (print): ______________________________

Login (print): ______________________________

Ethics Pledge: With your signature you **certify** the information above and you also **affirm** the following: "*I agree to complete this exam without unauthorized assistance from any person, materials, or device.*"

Signature: ______________________________

Date: ______________________________

Instructions: Please read these instructions carefully before you start. **Switch off** your phones, pagers, and other noisy gadgets! You are **not** allowed to have anything but a pen (pencil, eraser) and this exam on your desk. You are **not** allowed to talk to anyone during the exam. If you have a question, please raise your hand **quietly**. You must **remain seated quietly** until all exams have been collected. Remember that you can **not** claim grading errors if you do not use a **permanent** pen for your answers.

Do not open before you are told to do so!

You got ________ out of 40 points.

1 Binary Warmup (10 points)

For each of the following statements, determine whether it is either **true** or **false**. **(1 point each)**

1. A splay tree of n items can have a height of $O(n)$ for some sequences of operations.

2. Algebraic specifications of abstract data types provide test cases for implementations.

3. In JAVA generic types, i.e. the `T` in `Iterator<T>`, must be reference types.

4. Recoloring in a red-black tree can be understood as a promotion in the corresponding (2,4) tree.

5. Specifications for abstract data types are not concerned with the efficiency of implementations.

6. For linked lists, the transpose heuristic requires less data movement than the move-to-front heuristic.

7. A graph with n nodes has at most $O(2n)$ edges.

8. The height of a red-black tree with n items is always $O(\log n)$.

9. Object-oriented polymorphism means that the sending object decides which method to invoke.

10. Experimental analysis compares concrete implementations of the same abstract algorithm.

2 Tough Choices (8 points)

For each of the following questions, circle **one** answer out of the choices given. **(2 points each)**

1. What is the **worst-case** asymptotic complexity of **binary** search (in an array)?
   (a) $O(n)$
   (b) $O(1)$
   (c) $O(\log n)$
   (d) Binary search cannot be used for arrays.
   (e) None of the above.

2. Consider the level of **detail** available to clients in the form of **implementations** (e.g. source code for classes), **interfaces** (e.g. fully abstract classes), and **specifications** (e.g. algebraic ones). Which of the following seems most appropriate (A $<$ B means "A provides less detail than B")?
   (a) Interface $<$ Implementation $<$ Specification
   (b) Interface $<$ Specification $<$ Implementation
   (c) Implementation $<$ Interface $<$ Specification
   (d) Specification $<$ Interface $<$ Implementation
   (e) None of the above.

3. You need an ORDEREDSET implementation. Your application requires extracting the minimum and maximum **most frequently**; it also needs to check membership **a lot**; but it **rarely** inserts or removes elements. Which of the following data structures seems **most** appropriate?
   (a) Sorted, cyclic linked list.
   (b) Sorted, expandable, cyclic array.
   (c) Double-ended heap ("Deap") on an expandable, cyclic array.
   (d) Balanced binary search tree (e.g. red-black tree).
   (e) None of the above.

4. Consider the "order" of complexity classes from "fastest" to "slowest" in the intuitive sense. Which of the following seems most appropriate ($A < B$ means that if $f \in A$ holds then $f \in B$ also holds)?
   (a) $O(n) < O(\log n) < O(n \log n) < O(2^n)$
   (b) $O(1) < O(n) < O(\log n) < O(n^2)$
   (c) $O(1) < O(n) < O(n \log n) < O(2^n)$
   (d) $O(1) < O(\log n) < O(n^2) < O(n \log n)$
   (e) None of the above.

3 Short Answer (8 points)

For each of the following questions, answer in **one to three** sentences, the shorter the better. **(2 points each)**

1. We spent quite some time on designing the interfaces for `Stack<T>`, `BoundedStack<T>`, and `UnboundedStack<T>`. Why did we settle on this peculiar design?

2. In the **very populous** banana republic of FALSADICHO, people (all of whom have unique ID tattoos) are either for the unelected president or they are terrorist sympathizers. The secret police needs to check whether an ID is "good" or "bad" **very quickly**; no "bad guys" should get away, but the occasional "good guy" getting hanged is alright. Propose a "patriotic" data structure.

3. Contrast **depth-first search** and **breadth-first search** on a graph of your choice. Which one would you pick to determine whether a graph is connected or not? Why?

4. You have a binary search tree T. Give a pseudo-code implementation of `IORP(T, f, t)`, an operation that prints all keys k in T for which $f \leq k \leq t$ holds in order. (**I**n **O**rder **R**ange **P**rint)

4 Fun with (2,3) Trees (7 points)

Consider a (2,3) tree, i.e. a multiway search tree in which each internal node has either 2 or 3 children. **Again, this is a (2,3) tree, *not a (2,4) tree*!** Starting with an empty tree, sketch the evolution of the (2,3) tree as insertions of the keys 1, 2, 3, 4, and 5 are performed, in that order. Be sure to show the **"before" and "after"** in cases where the structure of the (2,3) tree is violated temporarily.

5 Fun with AVL Trees (7 points)

Consider an AVL tree, i.e. a height-balanced binary search tree in which each internal node has balance $b \in \{-1, 0, +1\}$. Starting with an empty tree, sketch the evolution of the AVL tree as insertions of the keys 1, 3, 2, 7, and 8 are performed, in that order. Be sure to show the **"before" and "after"** in cases where the balance of the AVL tree is violated temporarily; also give the balance of each node after each insertion.

This page is intentionally **mostly** blank in case you run out of space elsewhere. If you ended up here early, please go over **everything** again and remain seated **quietly**! Make sure that the title page is filled out correctly and in **permanent** pen. Maybe you want to "rewrite" your **answers** in permanent pen as well?