1 Exam Policy

- Wednesday, 03/16/16, 10:15am - 12:15 am. Total length: 2 hrs, start promptly at 10:15am. Please be on time.

- Format: Total Score 60 (final takes 30% of your total score). 10 problems in total. 8 basic questions (44 points). 1 algorithm design problem (6 points). 1 analysis problem (10 points).

- Note: Two pages of handwritten¹ or typed (with font size at least 10) notes (8.5 by 11 inches, front and back) allowed.

- The solution of the final will be posted right after the exam on 03/16/16. The grades should be posted by the mid-night. You can check your exam and report any grading issues by the noon of 03/18/16 (Friday).

2 General Comments

Do your best to get as many points as you can: answer the questions you know first, and briefly show your thinking or write down a partial answer even when you’re not complete sure.

Most of the questions will be of the form "prove X" or "show Y," with some guidance as to what methods you should use and how formal you should be. A few questions could be creative or qualitative, e.g., "Describe how to modify data structure X in order to efficiently implement operation Y" or "Which data structure would be most appropriate for problem Z?"

3 What to Study

- Solutions to assignment 1, 2, 3, 4, 5, mid-term.

- Slides and lecture notes posted online.

- Textbook Chapter 1, Chapter 2.1-2.4, Chapter 3.1-3.3, Chapter 4.1, 4.3, 4.4, 4.5.

- Your own notes from the lecture.

- Discussions on piazza. Ask questions and make good use of office hours.

4 Topics & Sample Problems

The scope of the final exam is from Week 1 to Week 9.

¹If extensive handwriting is a problem for you, please let me know as soon as possible and accommodations can be made.
Topics before mid-term

• Big-O, big-Omega, and big-Theta. For each of these, you should know the formal definition and be able to use it to prove or disprove statements.

Sample Problems: R-1.15 to R-1.20

• Loop invariants. Sample Problems: lecture notes and assignment 1.

• Stacks, queues, vectors, and lists: Basic definitions and common usage.

Sample Problems: R-2.1, slides and assignment 2.

• Amortized analysis: accounting and potential method. You will be free to use either method in your answer, but make sure you are comfortable with at least one.

Sample Problems: C-1.1, C-1.2, lecture notes and assignment 2.

• Trees: binary trees, non-binary trees, array-based implementation of binary trees. Questions may involve traversals, insert/delete, and recursive and iterative algorithms, properties about trees.

Sample Problems: R-2.3, R-2.4, C-2.9, C-2.10, C-2.19, C-2.20, assignment 2.

• Heap and priority queue: insertion and removal operations, how to make use of heap in algorithms.

Sample Problems: R-2.11, R-2.12, R-2.16, R-2.17, assignment 3.

Topics after mid-term

• Binary Search Tree: definition and properties of of binary search trees, insertion and removal operations.

Sample Problems: R-3.1, R-3.2, assignment 3.

• AVL tree: the height balance property, the relation between the height and the number of items, the rotation operations after insertion or removal.

Sample Problems: R-3.3, R-3.4, assignment 3, 4.

• (2,4) tree and red-black tree: definitions of both trees, insertion and removal operations of (2,4) trees, the correspondence between (2,4) trees and red-black trees.

Sample Problems: R-3.8, R-3.9, R-3.10, assignment 4.

• Comparison-based and non-comparison-based sorting algorithms, the decision tree method: implementation and complexity of various sorting algorithms, using the decision tree method to prove simple lower bounds.

Sample Problems: R-4.14, C-4.15, assignment 5.

Attention: we have mentioned a few alternatives for the implementation of operations of data structures during the term. For exam purpose, please stick to the implementation in the slides and the textbook.