Syllabus for Day 1 of the Ph.D. Qualifying Exam in Computer Science Revised (4/29/2022)

Part I of the Ph.D. qualifying exam is for four hours and consists of two questions each from the following five areas: Computer Architecture, Operating Systems, Programming with Data Structures & Algorithms, Discrete Mathematics, and Algorithm Design & Analysis. A student is required to answer any eight out of the ten questions (if a student answers more than eight questions then an arbitrary eight questions will be counted). The exam is closed-book and closed-notes. The use of electronic devices is not allowed except for a scientific calculator.

This document describes the list of major topics from which the questions for each area will be chosen. Each major topic is accompanied by (i) suggested text(s) and (ii) a Computer Science course that covers the topic in sufficient depth.

Computer Architecture

Topics: basic computer organization and design, instruction set architecture, control unit design, arithmetic and logical (ALU) design, processor basics (including high-level languages, assembly code and machine code) input/output organization, memory organization and hierarchy, cache memory (cache operation and performance), virtual memory (pages, page swapping, page tables, translation lookaside buffer), parallelism (pipelines, multiprocessors, cache coherence, atomic instructions and synchronization), performance and performance metrics (CPU time, clocks per instruction, etc.)

Suggested Texts:
J. Hennessy and D. Patterson, *Computer Organization and Design RISC-V Edition*, 1st edition, Morgan Kaufmann, 2017

Course: CompSci 458 Computer Architecture

Operating Systems

Topics: Process management including process creation, switching, multithreading, scheduling, communication and concurrency control; memory management including paging, segmentation and virtual memory; Systems programming.

Suggested Text:

1. William Stallings. *Operating Systems: Internals and Design Principles*, 7th ed.
2. Kay A. Robbins, Steve Robbins, *UNIX Systems Programming: Communication, Concurrency and Threads*, Prentice Hall; 2nd Revised edition (June 27, 2003) (Available on Safari Online)
3. Allen Downey. *The Little Book of Semaphores* (Open Source Textbook)

Course: CompSci 537: Introduction to Operating Systems

Programming with Data Structures & Algorithms

Topics: objects, recursion, exceptions, container ADTs, dynamic arrays, iterators, linked lists, generic classes, stacks, queues, binary search trees, graphs (depth-first and breadth-first search), hashing, sorting, min-heaps.

Suggested Text: Michael Main. *Data Structures and Other Objects Using Java*, 4th ed., Addison-Wesley.
2011.

Course: CompSci 351: Data Structures and Algorithms

Discrete Mathematics

Topics: logic, sets, functions, proofs, mathematical induction, counting, binomial coefficients and identities, discrete probability, basic graph theory.

Suggested Text: Kenneth Rosen, *Discrete Mathematics and its Applications*, 7th ed., McGraw-Hill, 2012. Chapters 1, 2.1-2.3, 5.1, 5.2, 6.1-6.5, 7, 10.

Course: CompSci 317: Discrete Information Structures

Algorithm Design & Analysis

Topics: asymptotic notation, solving recurrence relations, stacks, queues, vectors, lists, trees, priority queues and heaps, hashing, binary search trees (including AVL trees and red black trees), sorting and selection, structures/algorithms for disjoint sets, algorithm design techniques (the greedy method, divide and conquer, recursion, dynamic programming), graph algorithms (including minimum spanning tree algorithms and shortest path algorithms).

Suggested Texts:

(i) Michael Goodrich and Roberto Tamassia, *Algorithm Design*, John Wiley and Sons Inc., 2002. Chapters 1.1-1.4, 2 to 7.
(ii) Sanjoy Dasgupta, Christos Papadimitriou and Umesh Vazirani, Algorithms, 2008. Chapters 0, 2 to 6.

Course: CompSci 535: Algorithm Design and Analysis