Part I of the Ph.D. qualifying exam is for four hours and consists of two questions each from the following areas: Computer Architecture, Operating Systems, Programming with Data Structures & Algorithms, Discrete Mathematics and Algorithm Design & Analysis. A student is required to answer any eight questions. The exam is closed-book and closed-notes. The use of electronic devices is not allowed.

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: register transfer language, basic computer organization and design, instruction set design, control unit design, arithmetic and logical (ALU) design, pipelining and vector processing, input/output organization, memory organization.


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:

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 (insertion sort, merge sort).


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.


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:

Course: CompSci 535: Algorithm Design and Analysis