Syllabus for Algorithmics Component of PhD Qualifying Exam I (Theory)

May 1999

1 Description

The new Theory exam, Exam I in the Computer Science PhD written qualifier program, has two equally weighted parts, theoretical foundations and algorithmics. Students who entered under the previous system of exams, and who have completed just one of the Foundations or Algorithmics exams, have the option of doing just one of the two parts in half the total allotted time.

This algorithmics portion of the exam tests your knowledge of fundamental concepts in the design and analysis of algorithms and data structures. The exam is written with the understanding that you had strong data structures and algorithms courses in your undergraduate training or that you have taken the first-year graduate course in algorithms at Indiana University.

The new exam will focus on design and analysis of algorithms. It will continue to assume some knowledge of data structures (such as presented in references [2,4]). References [1,3,5,6] cover the main material of the exam. Each book covers most of the field. Below we have tried to indicate which book has the best coverage of each topic.

2 Topics

2.1 Algorithmics background

1. **Basic data structures (lists, stacks, queues and trees):** [4] Chapters 3, 4.
2. **Data structures for sets:**
   (a) Hashing (including extendible hashing): [4] Chapter 8.
   (b) Search trees (including AVL trees and B-trees) [2] Chapters 4 and 5.
   (c) Self-organizing lists and trees: [4] Chapters 6.2 and 7.3.
3. **String matching:** [4] Chapter 5.5.
4. **Graphs and graph algorithms:** [3] Chapters 23–27.
5. **Memory management:** [4] Chapter 10.

2.2 Algorithms analysis

1. **Mathematical notation and techniques:**
   (a) Asymptotic notation: [5] Chapter 1.10, 4.2.
   (b) Summing series, Products and Binomials: [5] Chapter 2, 3.1-2, 3.4-6.
   (c) Recurrence relations: [5] Chapter 5.

2. **Algorithmic analysis techniques:**
   (a) Worst case asymptotic analysis
   (b) Average case analysis

3. **Algorithm design techniques:**
   (a) Divide-and-conquer: [2] Chapter 10.1; [5] Chapter 5.2, 5.5.
   (b) Backtracking: [2] Chapter 10.4; [5] Chapter 4.3.1.
   (c) Dynamic programming: [3] Chapter 16.
   (d) Greedy: [3] Chapter 17.1–17.3.

4. **Basic lower-bound techniques:**
   (a) Decision trees [6].
   (b) Adversary arguments [6].

5. **Algorithms for searching, selecting, and sorting:**
   (a) Searching: [6] Chapter 2.
   (b) Selecting: [6] Chapter 3.
   (c) Sorting: [6] Chapter 4.

References

[**1**] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley Publishing Company, 1974.

[**2**] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *Data Structures and Algorithms*. Addison-Wesley Publishing Company, 1983.

[**3**] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms.* MIT and McGraw-Hill, 1990.

[**4**] H.R. Lewis and L. Denenberg. *Data Structures and Their Algorithms*. Harper-Collins Publishers, 1991.

[**5**] P.W. Purdom and C.A. Brown. *The Analysis of Algorithms*. Holt, Rinehart and Winston, 1985.

[**6**] G.J.E. Rawlins. *Compared to What?* Computer Science Press, 1991.