COMS21103: Data Structures & Algorithms
Guide to examinable material

January 11, 2016

The following is a guide to material from the lectures which is, and is not, examinable. Any material from lectures which is not specifically included in the "non-examinable" list should be assumed to be **examinable**.

For each algorithm mentioned in the list of examinable material, you should understand the algorithm and be able to reproduce it in pseudocode. You should also understand the proofs of correctness and be able to answer questions on them, but will not be expected to reproduce them from scratch. For each data structure, you should understand how it works and the operations it supports, and any associated proofs. Finally, you should also be confident with extending algorithms and data structures given in the course, and designing new algorithms of your own.

Examinable material

Concepts

1. Growth of functions and asymptotic notation.
2. Recurrences.
3. Asymptotic analysis of algorithms.
4. Properties of sorting algorithms, stability, etc.
5. Decision tree model for sorting.
6. Lower bounds for sorting.
7. Amortised analysis.
8. Shortest paths in graphs (directed and undirected, weighted and unweighted).
9. Minimum spanning tree problem.
10. Dynamic programming.
11. The line intersection problem.
12. Output sensitive algorithms.
13. String matching problem.
14. Stable matching problem.
15. Flow networks and the max-flow problem.
16. Polynomial multiplication using discrete Fourier transforms.
17. Linear programming.

Algorithms

1. Comparison-based sorting algorithms: Selection, Insertion, MergeSort, and QuickSort.
2. Counting and Radix Sort.

3. Graph traversal algorithms: Breadth-first search and Depth-first search.
4. Dijkstra's algorithm.
5. The Dynamic programming algorithm for Largest Empty Square.
6. The Dynamic programming algorithm for Weighted Interval Scheduling.
7. The Bellman-Ford algorithm.
8. Johnson's algorithm.
9. The line-segment intersection algorithm (the one seen in the lecture).
10. Kruskal's algorithm (excluding the proof of correctness).
11. The Gale-Shapley algorithm for stable matching.
12. Irving's algorithm for for the 'roommate' matching problem.
13. The Blum, Floyd, Pratt, Rivest, and Tarjan algorithm for order statistics.
14. Knuth-Morris-Pratt algorithm for string matching.
15. Boyer-Moore-Horspool algorithm for string matching.
16. Finite state machines for string matching.
17. The Ford-Fulkerson method for max-floow.
18. The Edmonds-Karp method for max-flow.
19. Fast-Fourier Transform (FFT) for polynomial multiplication.
20. The simplex algorithm.
21. The initialise-simplex algorithm.

Data structures

1. Binary trees.
2. Priority Queues implemented using Binary heaps.
3. 2-3-4 trees.
4. Disjoint-set data structures implemented using reverse trees.
5. Skip lists (excluding the proofs).
6. Fibonacci heaps (excluding the proofs).

Non-examinable material

1. Biographical and historical information from the end of lecture slides.
2. Max-flow for image segmentation.
3. Integer programming.
4. Red-Black trees.
5. Bloom filters.