CS170 First Midterm Review Sheet

March 11, 2013

DISCLAIMER: This document is intended as a study material for your first midterm; *it does not necessarily accurately reflect the content of your exam!* Think of this as a not necessarily comprehensive list of terms and ideas that your GSI's would study if they wanted to review what you have learned so far.

1 Preliminaries, Numbers, Notation.

- Asymptotic runtime notation– $O(t), \Theta(t), \Omega(t)$
- Binary representation of integers
- Expressing runtime with a recurrence relation
- Solving recurrence relations–tree method, Master Theorem
- Multiplying matrices
- Reductions (i.e. arguing that computing a^2 cannot take less time than computing ab)
- Hashing
- RSA
- Euclid's algorithm

1.1 Practice Problems.

Section 1 worksheet, the first homework, and the following book problems may be useful:

0.1 For practicing with big-O, Θ, and Ω notation.
1.33 For a refresher of Euclid's algorithm and some runtime practice.
1.43 For RSA.

2 Divide and Conquer.

- Multiplying integers
- Strassen's Algorithm and matrix decomposition
- Binary Search
- Finding the median or kth-smallest element

2.1 Sorting.

- Mergesort
- Lower bound on comparison-based sorting

2.2 Fast Fourier Transform.

- Roots of unity
- Coefficient Representation of polynomials
- Value Representation of polynomials
- FFT Algorithm:
  - Evaluation
  - Interpolation

2.3 Practice Problems.

Section 2 worksheet, homework 2, and the following book problems may be useful:

2.1 For binary integer multiplication.
2.5 Practice solving recurrence relations.
2.12 Practice setting up recurrence relations.
2.9 Practice with FFT. If you feel shaky about the algorithm, trying to do this by hand might help.

3 Graph Search.

3.1 Definitions.

- Graph, Vertices, Edges, Weighted Edges, Directed Edges (Arcs)
- Directed (Acyclic) Graph
- Connected Component
- Strongly Connected Component
- Source, Sink
- Source Component, Sink Component
- `post` and `pre` numbers, and lemmas involving their relative values after depth-first search
- Back, Cross, Tree, and Forward edges
- Shortest Path
- Negative cost cycle

3.2 Algorithms.

- Depth-Firsth Search
  - Implementations
  - Runtime(s)
  - Limitations
  - Applications
- Breadth-First Search
  - Implementations
  - Runtime(s)
  - Limitations
  - Applications

- DAG Linearization (Topological Sort)
  - Implementations
  - Runtime(s)
  - Limitations
  - Applications
- Djikstra's Algorithm
  - Implementations
  - Runtime(s)
  - Proof of correctness
  - Limitations
  - Applications
- Bellman-Ford Algorithm
  - Implementations
  - Runtime(s)
  - Proof of correctness
  - Limitations
  - Applications

3.3 Practice Problems.

Section 3,4 & 5 worksheets, homeworks 3 & 4, and the following book problems may be useful:

3.2 Practice depth-first search and related vocabulary.
3.5 Practice applications.
3.13 Practice with understanding connectivity in directed and undirected graphs.
4.1 Practice with Djikstra.
4.2 Practice with Bellman-Ford .
4.14 Applications.
4.15 Applications.
4.21 Applications.

The following non-book problems might also be useful:

1. Explain the relationship between Djikstra's algorithm and BFS.
2. Explain why Bellman-Ford is different from Djikstra's algorithm (why it is able to find shortest paths in graphs with negative edges?)
3. Find an example that shows that Djikstra's algorithm cannot even handle negative-cost edges, not just negative-cost cycles.

4 Greedy Algorithms.

- Scheduleing
- Huffman Encoding
- Disjoint Sets (a.k.a. Union-Find) data structure

4.1 Minimum Spanning Trees

- Kruskal's Algorithm
- Prim's Algorithm
- Borouvka's Algorithm
- The Cut Property
- Variations of MST
  - Max Spanning Tree
  - Unique edge weights
  - etc.

4.2 Practice Problems.

Section 5 & 6 worksheets, homeworks 5 & 6, and the following book problems may be useful:

5.2 Practice with MST algorithms.
5.3 Connectivity and trees.
5.5 MSTs.
5.13 Huffman encoding.
5.17 Huffman encoding.
5.23 MSTs.