**CSC373 Final Exam Study Plan**

**Unit 1: Basics, Divide and Conquer, Greedy, Dynamic Programming**

* Introduction
  + Concepts
    - Problem Specification (Input, Output)
    - Efficiency Measurements
    - Correctness Proof
    - Termination Proofs
* Divide and Conquer
  + Examples
    - Modular Exponentiation
    - Merge-Sort
    - Karatsuba Integer Multiplication
    - Closest Pair of Points
    - L-Tiling of Square Board
    - Inversion Counting
    - Matrix Multiplication
  + Concepts
    - Master Theorem
    - Generic Divide and Conquer Strategy
* Greedy
  + Examples
    - Single-Thread Interval Scheduling
    - Number-of-Machines-Minimizing Interval Scheduling
    - Huffman-Encoding
    - Horn-Clauses
    - Knapsack Problem
    - Making Change
  + Concepts
    - Greedy Pros and Cons
    - Generic Greedy Strategy
    - Generic Correctness Proof
    - Optimal Substructure Property
    - Overlapping Substructure Property
* Dynamic Programming
  + Examples
    - Longest Increasing Subsequence
    - Longest Common Substring/Subsequence
    - Chain Matrix Multiplication
    - Maximum Independent Set in Trees
    - Edit Distance
    - Weighted Interval Scheduling
    - 0/1 Knapsack Problem
    - Neat Printing
  + Concepts
    - Elements of Dynamic Programming Solution
      * Semantic Array
      * Computational Array
      * Pseudo-code
    - Correctness of Dynamic Programming Solution
      * Equivalence of Semantic and Computational Arrays
      * Analysis of Runtime
    - Dynamic Programming Sub-Paradigms
    - Recursive vs. Iterative (Memoization vs. Array/Table) Pros and Cons

**Unit 2: Graphs, Shortest Paths, Network Flow, Linear Programming**

* Graph Theory
  + Concepts
    - Trees are graphs
    - Subgraphs
    - Spanning Trees
    - Spanning Subgraphs
    - Strongly Connected Components
    - Bipartite Graphs
  + Graph Types
    - Directed/Undirected Graphs
    - Subgraphs
    - Weighted Graphs
  + Graph Representations
    - Edge list
    - Adjacency lists
    - Adjacency matrix
  + Algorithms
    - Breadth First Search
    - Depth First Search
    - Kruskal’s Algorithm
    - Prim’s Algorithm
* Shortest Path Algorithms
  + Concepts
    - Negative-weight cycles imply the “shortest” paths have infinite length – problem is ill-formed
    - Can “re-weight” a graph to eliminate negative edges
  + Single-Source Algorithms
    - Dijkstra’s Algorithm (can only handle positive weights)
    - Bellman-Ford Algorithm (can only handle negative weights, can be modified to detect negative cycles)
  + Multi-Source / All-Pairs Algorithms
    - Bellman-Ford on every vertex (not very efficient)
    - Floyd-Warshall (DP solution)
    - Multiplication of Adjacency Matrices
    - Johnson’s Algorithm (Dijkstra on every vertex of a re-weighted graph)
* Network Flow
  + Concepts
    - Flow Value (total amount of flow in a configuration)
    - Network Flow (flow configuration)
    - Min-Cut Theorem
    - Super-Sources / Super Terminals (can be used to easily solve multi-source/terminal problems)
    - Max-Flow Value of an Integral Network is also Integral
  + Algorithms
    - Ford-Fulkerson
    - Edmond- Karp
  + Applications
    - Maximum Bipartite Matching
* Linear Programming Introduction
  + Concepts
    - Linear Functions
    - Linear Constraints
    - Feasible Solutions
    - Feasible Regions
    - Matrix Representations
    - Polyhedrons
    - Polytopes
    - Vertex Degeneracy
  + Problem Types
    - Infeasible
    - Unbounded
  + Algorithms
    - Interior Points
    - Ellipsoid
    - **Simplex**
* Linear Programming Theory
  + Forms
    - Canonical / Standard Form
    - Slack Form
  + Conversion
    - Minimization / Maximization Conversions
    - Equality/Inequality Conversions
    - Standard/Slack Conversions
  + Duality
    - Duality Theorem
    - Farkas Lemma
    - Weak Duality
    - Strong Duality
* Simplex (in detail)
  + Method
    - High level (traversal along corners)
    - Low level (by hand)
  + Concepts
    - Basic Variable
    - Non-Basic Variable
    - Basic Solution
    - Pivoting
    - Bland’s Rule
    - Cycling
  + Running Time

**Unit 3: P vs. NP, Approximation Algorithms**

* P-Space Basics
  + Formal Language Equivalence (with encodings)
  + Problem Types
    - Search
    - Decision
  + Problem Classifications and Venn Diagram of Supposed P-Space Layout
    - P
    - NP
    - NP-Hard
    - NP-Complete (NPC)
    - Co-NP
    - Co-NPC
* Definitions
  + Canonical Search Problem of NP Decision Problems
  + Reductions
    - Poly-time reductions
    - Self-reductions
  + Circuit Satisfiability
    - Boolean Variable
    - Literal
    - Clause
    - CNF Formula
  + Graph Problems
    - Clique
    - Vertex Cover
    - Edge Cover
    - Hamiltonian Cycle
    - Hamiltonian Path
* Example Language Classifications
  + P
    - 2-CNF SAT (Search)
    - MST (Decision)
    - Shortest Path (Decision)
  + NP
    - K-CNF SAT (Search)
    - 3-CNF SAT / 3SAT (Search)
  + NPC
    - Circuits
      * 3SAT (search) (self-reducible)
      * Circuit Satisfiability (self-reducible)
    - Graphs
      * Vertex Cover (self-reducible)
      * Hamiltonian Path
      * Hamiltonian Cycle (self-reducible)
      * Travelling Salesman (self-reducible)
    - Sets
      * Subset Sum
      * Knapsack Problem (Decision)
      * Partitions
  + NP-Hard
    - k-Cliques (self-reducible)
  + Unknown (probably not in P)
    - 3SAT (Decision)
    - Longest Path (Decision)
    - Travelling Salesman (Decision)
* Proof Layouts
  + NP Proof: Polynomial Verification
    - Potential Solutions (Certificates) are Polynomial Sized
    - Verification Algorithm Runs in Polynomial Time
  + NP-Complete Proofs
    - Problem is in NP
    - Some NPC problem poly-time reduces to the target problem
* Approximation Algorithms
  + Definitions
    - Problem Types
      * Maximization
      * Minimization
    - Approximation Ratios
      * Maximization
      * Minimization
  + Examples
    - 2-approximation of Vertex Cover
    - 2-approximation of Triangular TSP
    - 2-approximation of Weighted Vertex Cover (use rounded linear program)
    - 2-approximation of Graph’s Steiner Tree
    - (8/7)-approximation of Max-3SAT
    - Constant-factor approximation of TSP DNE
    - (log m + 1)-approximation ratio of Set Cover
  + Proofs
    - Approximation Ratios