CS 312 Study Guide for Final Exam

The exam is closed book and will have a time limit of three hours, you will be allowed one page of notes that you must create yourself (the purpose of the page of notes is two fold: 1) to enhance your preparation through your deciding what to include and how to represent it and 2) to obviate the need for memorizing details.) This page of notes should be an ancillary study aid and should not be used as an attempt to take the place of core knowledge and understanding but rather to enhance it. Note that for all algorithms you should be able to figure out time and space complexity.

The exam will consist of problems similar to those encountered on the homework. It is implicitly comprehensive in that you are responsible for material throughout the course, but it is explicitly weighted towards material from the second half of the semester. You should be prepared to answer questions over any of the material in the reading or that we covered in class. The following topic list gives a good, high-level coverage of the material for which you should be prepared to answer questions:

- Dynamic Programming
  - Philosophy and how to use it
  - Longest increasing subsequence, binomial coefficient, edit distance, knapsack with and without repetition, Recursion and Memoization
- Linear Programming
  - Setting up an LP from an objective and constraints, putting an LP in standard form, basic idea of Simplex algorithm (you should be able to a simple LP)
- Intelligent Search
  - Backtracking, Branch and Bound, beam search, bounding functions
  - TSP with B&B, reduced cost matrix, State space search with partial path
- Complexity Classes
  - P, NP, NP-complete
  - Bounded approximation algorithms, approximation factor
- Local Search
  - Basic algorithm and properties, local minima
- Advanced Algorithms
  - randomized algorithms
    - Monte Carlo vs. Las Vegas
    - Amplification of stochastic advantage
  - quantum computing
  - genetic algorithms
  - perceptron learning algorithm
- Analysis
  - Empirical analysis, Average case analysis (high level)
- Overall summary
  - When to use which the basic 312 paradigms
    - Divide and Conquer

        - Problem has natural hierarchy with independent branches
        - Speed-up happens when we can find short-cuts during partition/merge that can be taken because of the divide and conquer paradigm
- Graph Algorithms
    - When finding reachability, paths, and properties of tasks represented as graphs, often fall under other approaches
- Greedy
    - Common simple and fast approximation approach, occasionally optimal
- Dynamic Programming
    - Overlapping subproblems (given by a recursive definition) that are only slightly (constant factor) smaller than the original problem, solved with the proper ordering
- Linear Programming
    - Any optimization with linear objective and constraints
- Intelligent Search
    - Effective when we have some heuristic knowledge of the search space to allow pruning
- Local Search
    - Simple optimization technique for many complex search spaces – local optima issues
- Stochastic Algorithms
    - Sampling problems, amplification of stochastic advantage, takes advantage of fast computers, etc.