**CS 310 Final Exam Study Notes**

Last Modified: 28 April 2022

This will be a comprehensive final exam, with about ⅓ of the material coming from earlier work in the semester and ⅔ from material since the last test. For the material from earlier in the semester, the focus will be on the concepts of algorithm analysis: counting basic operations, big-Oh, -Omega, and -Theta, and use of the master theorem, and not as much on specific algorithms.

For the new material, you should know or be able to do:

* the classification of problems into tractable, intractable, and unsolvable, and examples of each
* the NP problem-solving strategy and examples of problems that can be solved using it
* the definitions of the sets P, NP, and NP-Complete, and examples of problems in each set
* polynomial reducibility and examples of reductions of one problem to another
* the four main strategies of coping with NPC (and NP-Hard) problems, and examples of problems for which each is suited
* the general form of backtracking algorithms, what they are used for, and examples
* the general form of branch-and-bound algorithms, what they are used for, and examples
* the general form of heuristic algorithms, what they are used for, and examples

Some problems to look at:

1. Given a set of *n* jobs and *n* workers, manually trace the branch-and-bound algorithm to find an optimal assignment of jobs to workers.
2. Given a set of *n* items each with a weight and a value, and a knapsack of a given weight capacity *W*, manually trace the branch-and-bound algorithm to find an optimal selection of items to place in the knapsack.
3. Give a polynomial algorithm to determine if a particular assignment of colors to vertices of a graph is a valid *k*-coloring of the graph.
4. A certain problem can be solved by an algorithm whose number of operations is in *O*(*n*lg*n*). Is the problem tractable or intractable, or is there not enough information to know?
5. Generate all permutations of the set {a, b, c, d} using backtracking.
6. Give a heuristic algorithm for the graph coloring problem.
7. Find a planar graph that does not contain a triangle that requires at least three colors for a valid coloring.