CSE 415: FINAL EXAM REVIEW LIST
OPEN BOOK, OPEN NOTES, IN-CLASS EXAM

1. Search (Ch 3)
    - Be able to give a formal state-space model for a problem expressed in English. Formal means to specify S, s, A, f, g and c as sets or functions as appropriate.
    - Be able to specify what would be the dead states for a given problem.
    - Be able to generate part of a search tree for a given model, either depth-first or breadth-first.
    - Be able to answer questions about the completeness and complexity of the various search variants given in Chapter 3.
2. Informed Search (Ch 4)
    - Be able to explain the use of a heuristic function in a search or to give an example of one for a stated problem.
    - Be able to apply any of the following search methods to a well-stated problem and show a portion of the search.
        - A* algorithm
        - steepest-ascent hill climbing
    - Be able to answer questions about complexity, completeness, and optimality for the algorithms of Ch 4.
3. Constraint Satisfaction Problems (Ch 5)
    - Be able to formalize a constraint satisfaction problem by specifying the sets of variables, possible values, and constraints.
    - Be able to explain or illustrate how a backtracking tree search for a constraint satisfaction problem would work: alone, with forward checking, or with arc consistency checks.
    - Be able to compare how a general heuristic search would compare with a constraint satisfaction search when both are applicable to a given problem.
4. Game Playing (Ch 6)
    - Be able to develop a utility function for a given game or show how a given one works.
    - Be able to show how a basic minimax search works for some given example.
    - Be able to show how the alpha-beta procedure works for some given example.
    - Be able to show how shallow search might be used to improve the alpha-beta procedure.
    - Be able to answer questions about Samuels' checker player.
    - Be able to answer questions about programming the Kalah game.

5. Logic (from the handout of my lecture)
    - Be able to interpret predicate calculus formulas in English.
    - Be able to answer questions about the normalization done by resolution theorem provers for predicate calculus in order to get the formulas into conjunctive normal form.
    - Be able to show how to produce a resolvent on a SMALL set of SIMPLE formulas.
    - Be able to explain what is going on in a given small resolution proof.
6. Learning (Ch 18 for decision trees and ensembles; Ch 20 for Bayesian learning and neural nets)
    - Be able to use a given decision tree to classify a test vector.
    - Be able to construct the best decision tree for a given training set by
        (a) yourself, given the criteria for best
        (b) information gain
    - Be able to answer questions about overfitting in decision trees and what can be done about it.
    - Be able to answer questions about the ensembles: bagging, boosting, stacking, and Chou's variant.
    - Be able to show how a given perceptron classifies a test vector.
    - Be able to answer questions about how perceptrons learn their weights.
7. Vision (from lecture notes only)
    - Be able to answer questions about the low, mid, and high-level aspects of vision.
    - Be able to apply a mask to a (portion of an) input image to produce an output image.
    - Be able to apply an edge operator, such as the Sobel or Prewitt operators, to a portion of an image to find gradient magnitude and/or angle.
    - Be able to answer questions about the difference between these simple operators and the Canny edge operator.
    - Be able to show how K-means clustering works on a very small data set. can be used to find regions in a gray-tone or color image.
    - Be able to show how to reduce a high-level computer vision problem to graph matching.
    - Be able to show how relational indexing would work on a given problem.
    - Be able to answer conceptual questions about the use of neural nets in Rowley's face detector.