Practice Problems for Midterm Exam

Uninformed Search

1. Which attributes are typically stored on the nodes of a search tree for uninformed search? What additional information would you need to store for informed search?
2. What is the difference between TREE-SEARCH and GRAPH-SEARCH? Related questions may include:
   - Which type of search do we use when we are searching for a path on a grid? Why?
   - Which type of search do we use when we are searching for a solution to the 8-queens problem? Why?
   - Which type of search do we use when we are searching for a path between cities on a road network? Why?
3. What are the criteria used to compare search algorithms?
4. What are the properties of breadth-first (or depth-first, or uniform-first, or iterative-deepening depth-first, or bidirectional breadth-first) search? Justify the complexity bounds you report.
5. Which uninformed search technique has the best time and space complexity? What is the time and space complexity?
6. Which algorithm would you use if you knew that the depth of the search tree is bounded and why?
7. Prove that uniform-cost search and breadth-first search with constant edge costs are optimal when used in GRAPH-SEARCH.
8. Show a state space with varying step costs in which GRAPH-SEARCH using depth-first (or iterative-deepening depth first) finds a suboptimal solution.
9. Under which circumstances can you apply bidirectional search?
10. You need an additional data structure in order to be able to solve GRAPH-SEARCH problems. How is this data structure called and how can you implement it?
11. Using one or more of the criteria used to compare search algorithms, provide arguments to support the following statements:
    - Iterative deepening search is preferred over breadth-first search.
    - Bidirectional search is preferred over breadth-first search.

Informed Search

1. What is the name of the overall search strategy that selects the fringe node which maximizes an evaluation function $f(n)$? Name two approaches that implement this search strategy? How do they define the evaluation function? What does a heuristic represent?
2. What are the properties of greedy best-first search? How does it work? What are the properties of A* in TREE-SEARCH and GRAPH-SEARCH? How does it work?
3. True or False?

- Greedy Best-First search with $h(n) = 0$ for all nodes n is guaranteed to find an optimal solution if all arc costs are 1.
- In a finite search space containing no goal state, the A* algorithm will always explore all states.

4. Prove the following statement: "If G_2 is a node that corresponds to a goal state but a suboptimal path, A^* will select a reachable node n along the optimal path before selecting G_2 for expansion".
5. What is the definition of an admissible heuristic? Prove that A^* always finds the optimal path in TREE-SEARCH if the heuristic is admissible.
6. Is the "Manhattan distance" an admissible heuristic when planning on a grid? Is it true that the "Manhattan distance" is an admissible heuristic for finding the shortest path through the real Manhattan in New York City? [Hint: Go to Google maps and search for: "Broadway & W 33rd St, New York, 10001". Consider the distance between Madison Square Park and Times square.]
7. What is the definition of a consistent heuristic? Prove that a consistent heuristic is also admissible.
8. Show that if a heuristic $h(n)$ is consistent, then the evaluation function used by the A* algorithm $f(n) = h(n) + g(n)$ is non-decreasing along any search path. Show that A* expands nodes with a non-decreasing order of evaluation function.
9. Show that the fist goal node expanded by the A* algorithm also corresponds to the optimal goal node. Show that A* expands all nodes with $f(n) < C*$, where $C*$ the optimal solution cost.
10. Under which circumstances is the A* algorithm complete?
11. Which optimal informed search algorithm expands fewer nodes than A* for the same heuristic? Ignore the effects of ties.
12. You might be provided with a graph and asked to show the sequence of nodes expanded by A*, as well as the g, h and f values. You might be asked if the heuristic was admissible.
13. Under which conditions will A* search expand the same nodes as Breadth-First search?
14. Which of the following are admissible, given admissible heuristics h_1, h_2? Which of the following are consistent, given consistent heuristics h_1, h_2?
    - $h(n) = min\{h_1(n), h_2(n)\}$
    - $h(n) = wh_1(n) + (1 - w)h_2(n)$, where $0 \leq w \leq 1$
    - $h(n) = max\{h_1(n), h_2(n)\}$
15. What is the definition of the effective branching factor and how is it used to compare heuristics?
16. Suggest possible directions for automatically designing heuristics.
17. Give examples of relaxed versions of the 15-puzzle?
18. What is the definition of a dominant heuristic?

Adversarial Search

1. What is the definition of a two-player zero sum game?
2. What are the differences between a search tree used for adversarial search versus a tree for classical single-agent search problems?
3. Describe the operation of the minimax algorithm. What properties does it have in terms of time and space complexity?
4. Why do we use depth-first search to solve adversarial search problems?
5. Describe the operation of alpha-beta pruning. What is the time complexity assuming perfect ordering? What is the time complexity on average?
6. You might be provided a search tree that corresponds to a game and asked to identify which nodes will not be checked by the minimax algorithm if we apply alpha-beta pruning. You might also be asked to compute the value at each node as computed by the minimax algorithm. You might be asked if a different ordering of the terminal nodes would result in increased pruning.
7. Are the following statements true or false?
   - A player using minimax is guaranteed to play optimally against any opponent.
   - A player using alpha-beta pruning is guaranteed to play optimally against an optimal opponent.
8. Do you know any other way to accelerate the performance of adversarial search other than alpha-beta pruning?
9. What requirements should a heuristic function satisfy in heuristic adversarial search?
10. Give examples of problems that might arise in heuristic adversarial search.
11. What are the differences between a search tree used for games with chance versus search trees for deterministic games?
12. Describe the operation of the expectiminimax algorithm.
13. What problem arises in heuristic adversarial search for games with chance in the definition of the heuristic evaluation function that does not exist in deterministic games?
14. What are the difference between a search tree used for *non zero-sum games* with *more* than 2 players versus search trees for two-player, zero-sum games?

Local Heuristic Search

1. When is it appropriate to use local search instead of classical search? What are the advantages and the disadvantages of local search?
2. Say we have a search space that is very large with a very large branching factor at most nodes, there may be infinite paths in the search space, and we have no heuristic function. What search method would be good to use in this situation and why?
3. Describe the basic operation of a hill-climbing approach.
4. How can you avoid the issue of plateuax in hill-climbing?
5. What does “probabilistic completeness” mean and how can you achieve a hill-climbing approach that is probabilistically complete?

6. You might be provided with a toy-problem (e.g., the n-queens problem) and asked to define a heuristic evaluation function that will allow you to apply hill-climbing. Given your heuristic function, you might be asked to solve a specific instance of such a toy problem with hill-climbing.
7. Describe the operation of simulated annealing. What is the motivation behind the development of this local search technique? How does the algorithm behave at very high "temperatures", and how it behaves at very low "temperatures"? When is simulated annealing probabilistically complete?
8. For what types of problems will hill climbing work better than simulated annealing? In other words, when is the random part of simulated annealing not necessary? Is Simulated Annealing with a constant, positive temperature at all times the same as Hill-Climbing?
9. What is the main principle behind local beam search? How is it different from traditional hill climbing? What are the properties of local beam search?
10. What are the main principles behind genetic algorithms? How are genetic algorithms different from local beam search? What are the properties of genetic algorithms?
11. You might be asked to design a solution based on genetic algorithms for a specific problem
12. Consider a genetic algorithm using only selection and mutation operators, that is without crossover. Is this equivalent to any other search algorithm? Justify your answer.

Constraint Satisfaction Problems (CSPs)

1. What is the definition of a constraint satisfaction problem?
2. What are two ways to approach the solution of constraint satisfaction problems? What techniques are more appropriate for each approach?
3. You might be provided a specific constraint satisfaction problem and asked to devise an incremental or local-search approach for solving it.
4. Why do we use a variation of depth-first search to solve constraint-satisfaction problems? How is this variation called?
5. What heuristics do you know for ordering variables in backtracking search? What heuristic would you use to order value assignments in backtracking search?
6. Describe the forward checking heuristic.
7. Describe the arc consistency heuristic.
8. What is the difference between the forward checking heuristic and the arc consistency heuristic? In other words, when will each heuristic detect an inconsistent assignment?
9. What is the idea behind intelligent backtracking? How does backjumping operate?
10. You might be provided an example of a constraint satisfaction problem (e.g., a map coloring problem) and asked to predict the effects of certain heuristics in backtracking search. (e.g., if we assign this variable a certain value, what will be the result of Forward Checking, etc.)

Logic-based Inference and Satisfiability

1. What is the conjunctive normal form (CNF)? Given an example of a logical sentence in CNF.
2. Use logical equivalences to show that the statement $\alpha \Leftrightarrow \beta$ (or a different logical statement) can be turned into conjunctive normal form.
3. Show that the clause $(\neg P_1 \vee \ldots \vee P_m \vee Q)$ is logically equivalent to the implication sentence $(P_1 \wedge \ldots \wedge P_m) \Rightarrow Q$.
4. What is the resolution rule and why is it important for logical inference?
5. Describe the steps of the satisfiability algorithm?
6. You might be provided with a knowledge base KB and asked to use the satisfiability algorithm in order to prove that it entails a new sentence α: $KB \models \alpha$.
7. What is the Davis-Putnam algorithm? What strategy does it follow to solve logic-based inference problems?
8. Describe heuristics for the Davis-Putnam algorithm.
9. What are three alternative methodologies to logic-based inference problems.

Classical Planning

1. Describe the version of the Planning Domain Definition Language (PDDL) that was covered in class. Given examples of challenges that can be described by PDDL.
2. You may be provided a problem and asked to build its PDDL specification.
3. What is the idea in forward state search for classical planning? What are the challenges of forward state search?
4. What is the idea in backward state-space search for classical planning? What are the challenges in backward state-space search?
5. How can you define heuristics for classical planning?

Intro to Decision Theoretic Agents and Probability Theory

1. What is the focus of decision theory? What are the basic steps of a decision theoretic agent?
2. You might be asked to provide any of the rules of probability theory discussed in the class:
   - Kolmogorov's axioms,
   - Product rule - with and w/o normalization,
   - Bayes rule - with and w/o normalization,
   - marginalization and conditioning,
   - independence and conditional independence properties.
3. You might be asked to compare the value of $P(\alpha|b)$ and $P(\neg\alpha|b)$ in a way that minimizes the number of probabilities that we have to be aware of.
4. Assume two random *binary* variables α, β. What is the minimum number of distinct probabilities that you need to know to compare $P(\alpha|\beta)$ and $P(\neg\alpha|\beta)$ using the Bayes rule? [Clarification of "distinct": if you know $P(\gamma)$ then you also know $P(\neg\gamma) = 1 - P(\gamma)$.]

Bayesian Networks: Properties and Exact Inference

1. What is the definition of a Bayesian network?

2. Why can we answer any query in a probabilistic domain given the Bayesian network that involves all the variables in this domain and their independence properties?

3. How can we compute the joint probability distribution $P(X_1, \ldots, X_n)$ from the conditional probabilities $P(X_i|Parents(X_i))$ stored on the Bayesian network that involves variables $X_1, \ldots, X_n$?

4. You may be provided with a Bayesian network and asked to compute the joint probability distribution. You can also then be asked to compute conditional probabilities that involve the variables of the Bayesian network. The conditional probability might:

    - Involve all the variables in the Bayesian network
    - Involve a subset of the variables in the Bayesian network

5. How can we compute exactly a conditional probability of the form $P(A|B)$ (without a normalization factor) from full joint probability distributions of the form: $P(A, B)$ and $P(\neg A, B)$.

6. How does exact inference by enumeration work in Bayesian networks? You can be given a Bayesian network (e.g., such as the Burglary-Alarm problem) and asked to compute a conditional probability by applying exact inference.

7. What are the basic ideas behind Variable Elimination? How is it advantageous over Inference by enumeration? You might be provided a Bayesian network and asked to compute a conditional probability by applying Variable Elimination.

Approximate Inference in Bayesian networks

1. Which techniques do you know for approximate inference in Bayesian networks? What is the basic idea/methodology behind approximate methods for inference in Bayesian networks?

2. How does direct sampling work? You might be provided with a Bayesian network and a series of "random" numbers and asked to sample an atomic event by employing direct sampling.

3. What direct sampling algorithms do you know to compute conditional probabilities? What are the basic ideas behind them? How do they work? You might also be provided a network and "random" numbers and asked to imitate the operation of these algorithms?

4. What is the Markov Blanket of a random variable in a Bayesian network? Why is it important?

Dynamic Bayesian Networks: Temporal State Estimation

1. What assumptions do we typically employ in a Dynamic Bayesian network? Give a graphical representation of a network that involves state and evidence variables.

2. What problems can we answer by employing temporal models (Dynamic Bayesian Networks)? Provide the mathematical representation of these problems.

3. What information/input should we have available in order to solve a problem with a Dynamic Bayesian Network?

4. Derive the filtering equation in Dynamic Bayesian Networks.

5. What is the filtering equation? Explain its various elements. You can be provided with a small temporal state estimation problem and asked to apply filtering for one or two steps in order to update the belief distribution. You will be provided a transition, an observation model and an initial probability distribution. For example, consider the rain-umbrella example provided in the notes, or a robot moving in a grid world with obstacles.
6. What is the backwards message used for solving the smoothing problem in temporal models?
7. Which state estimate do you expect to be more accurate, a filtering estimate, a smoothing estimate or a predictive estimate?
8. Assume that for a specific system the predictive estimate is more accurate than the filtering estimate? What can you infer about the properties of this system for which we are trying to estimate its state?
9. What is the difference between the “most likely explanation” problem formulation in temporal models compared to filtering, smoothing and prediction?
10. What is the advantage of the “most likely explanation” formulation over executing smoothing for all the states visited so far?