CSCE420: Introduction to Artificial Intelligence
Prior Class Test Questions

The following questions are all questions from prior midterm examinations. They represent more than what is is feasible in 70 minutes under test conditions, but , for purposes of study and revision, providing more examples is preferable to fewer.

Important: Your exam assumes that you will have prepared 3 pages of notes. These can be US-letter, double-sided, typeset or written—to be brought with you on October 27th. These practice questions can give you a sense of what it might be useful to include in your notes.

Question 1	6
Question 2	5
Question 3	6
Question 4	8
Question 5	15
Question 6	14

Question 1. (6 points)

In the context of intelligent agents there is a notion of a *known* environment. What does this mean, and how would it relate to a web-bot (or web-crawler, web-spider) whose search space is the World Wide Web? (2 pts.)

"Known" in the context of environments relates to the level of knowledge of the designer, not the agent. (1) The search space being uncovered as new links are followed reflects ignorance until the state is opened.(.5) The states are pages (or, the page content) but they are unknown until actually downloaded, because two different URLs can get you to the same page. (.5)

What is the relationship of quadratic programming to linear programming? Be as precise as you can. (2 pts.)

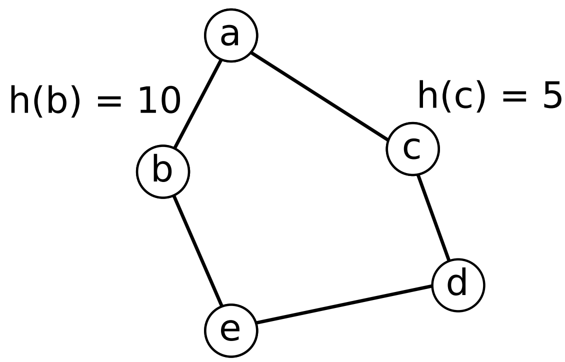
Commonality: Both quadratic programming and linear programming are optimization problems, wherein one seeks find a value that gives an extremal value of an objective subject to constraints. We encountered these when discussing local search (1).
Difference: Quadratic programming is a strict generalization of linear programs: all instances of LP problems are also QP problems, but not vice versa (1).

When is the least-constraining-value heuristic useless in a CSP search? (2 pts.)

When all values are equivalently constrained. (1)
When we want to enumerate all possible solutions. (1)

Question 2. (5 points)

Construct a map with 5 cities $\{A, B, C, D, E\}$ and 5 roads each of unit length, so that there are two routes from A to E. Give the values of an heuristic function so that depth-first search takes fewer steps to find the optimal path from A to E than A* search. (5 pts.)



Other heuristic values don't matter, because A* picks the wrong initial guess. However, they should be admissible, i.e., at least one.

Question 3. (6 points)

Assume that alpha-beta pruning reduces the branching factor from b to $\sqrt{b}$. How much deeper (in terms of plies) will an alpha–beta pruned search solve when compared with the standard minimax algorithm, given that both run for the same amount of time. Explain why mathematically. (3 pts.)

Alpha-beta pruning can solve a tree twice as deep as minimax in the same amount of time (pg 169.) = (1 point)
2 points for: $(\sqrt{b})^{m_1} = b^{(\frac{m_1}{2})} = b^{m_2}$ taking logs (base b) we get $m_1 = 2m_2$.

You are programming an agent to playing a complex game in which there is only a limited, fixed amount of time to search between moves. Your solution is to run minimax search with alpha–beta pruning until the time is nearly up, and then to make the best move found far. Suppose your agent does this and is about to pick the move. At that time the alpha and beta bounds for the current state have values 3 and 7 respectively. What can the difference between these two values be interpreted? (3 pts.)

You can interpret the value as representing, in a sense, the degree of uncertainty in the game outcome. Because we have not seen all possible states, there are unknowns in the outcome of the game, depending on the "fog" induced in requiring greater depth to solve. (1 point) We know, as a maximizing node, we can't do better than 7, although we will never do worst than 3.(2 points) When the difference is small, (and 3 is good enough) we may actually never care to get the full solution. (1 point, unless max of 3)

Question 4. (8 points)

Suppose you are given a screen-shot of a game of minesweeper in progress. You get a description of cells marked with flags as mines, some numbered with a count of neighboring mines (from zero to eight), and others yet to be opened. You cleverly decide to cast this as a CSP. You use a variable with domain $\{0, 1\}$ for each cell. Cells marked with mines are set to have value 1, those with known numbers are set of have value 0, and you introduce a summation constraint on the 8 neighboring cells. Suppose you use modify your favorite CSP algorithm to count the number of satisfying solutions.

If it tells you that there is only one solution, what do you conclude?

There is a risk free labeling with the information you have. (2)

It tells you that there are multiple satisfying solutions. What do you conclude?

You are unsure if there is a risk free solution. It is possible that a clever ordering would uncover new information that would solve this with without any risk. However, the CSP does not consider new information, nor the order of evaluation. Both are factors here. (4)

Suppose it tells you that there are no satisfying solutions. What do you conclude?

One of the blocks labeled mine isn't a mine. (2)

Question 5: Heuristics (15 points)

Consider heuristics where $h(n) = 0$ for all n for which $\textsc{Goal}(n) = \text{True}$, i.e., functions that have the value zero for states that are goals. Then the following statements are either true or false

Statement 1: Every admissible heuristic is consistent.
Statement 2: Every consistent heuristic is admissible.

For each statement: (1) identify whether it is true or false. (2) If it is true prove that it is so; otherwise, if it is false, provide a counterexample.

> *Reminder:* Using the notation from the textbook, with n' a successor of state n and a an action, a consistent heuristic is required to have $h(n) \leq c(n, a, n') + h(n')$.

Statement 1 is *false*. Here is a counter-example. Assume a linear graph between these cities, with the numbers appearing above as true costs
$A \xrightarrow{10} B \xrightarrow{10} C \xrightarrow{10} D \xrightarrow{10} Goal$.
Now consider the heuristic defined thusly:

$$h(D) = 9 < 10$$
$$h(C) = 19 < 20$$
$$h(B) = 2 < 30$$

Statement 2 is *true*. Here is a proof by contradiction (though, induction would work as well):
Assume the contrary and consider node n, the one of least distance from goal that violates admissibility. Thus,

$$\text{true\_cost\_to\_goal}(n) < h(n).$$

Now, n can't be a goal itself, since $h(n) = 0$ is admissible. So n must be connected, on its least cost path, to some other node *en route* the goal. Call that node n'. Then, since step costs are positive, we see

$$\text{true\_cost\_to\_goal}(n') < \text{true\_cost\_to\_goal}(n).$$

Also, $h(n') \leq \text{true\_cost\_to\_goal}(n)$, and accordingly,

$$\begin{aligned} h(n) &\leq c(n, a, n') + h(n') \\ &\leq c(n, a, n') + \text{true\_cost\_to\_goal}(n') = \text{true\_cost\_to\_goal}(n), \end{aligned}$$

but that means $h(n) < h(n)$, which is contradiction.

Question 6: Analysis of Games (14 points)

Consider a two-person, zero-sum game of perfect information we call G. Think of a game like checkers, or tic-tac-toe, but unlike those games, in G, more extreme scores are awarded for demolishing the opponent. Thus, the outcomes can be $O = \{-2, -1, 0, 1, 2\}$.
Suppose that agent $\mathcal{A}$ implements alpha-beta pruning correctly; it does a complete search down all the tiers reaching the leaves (i.e., it runs without cut-offs). Additionally, a fellow student has their own algorithm, $\mathcal{S}$, they've built using their own custom insights.
In the following, we report matches from playing G by pairs of players. When we say that the outcome of X *vs.* Y was (o_x, o_y), we mean that agent X played the first move as a maximizing player, that Y played second and was the minimizing player, and, after the game finished, X got outcome $o_x \in O$, and Y got $o_y \in O$. *(Answer each of these questions anew, discarding any inferences drawn from previous answers.)*

6.1 Is it possible to find that $\mathcal{A}$ *vs.* $\mathcal{S}$ yielded the outcome $(-1, 2)$, for the agents respectively? Explain/Interpret. (2 pts)

No. The reason it is a zero-sum game.

6.2 Is it possible for $\mathcal{A}$ *vs.* $\mathcal{S}$ to yield $(-1, 1)$, respectively, and for $\mathcal{S}$ *vs.* $\mathcal{A}$ to yield $(1, -1)$?
Explain/Interpret. (2 pts)

No, since $\mathcal{S}$ is outperforming $\mathcal{A}$, who is the optimal player.

6.3 Is it possible for $\mathcal{A}$ *vs.* $\mathcal{S}$ to yield $(1, -1)$, respectively, and for $\mathcal{S}$ *vs.* $\mathcal{A}$ to yield $(-1, 1)$?
Explain/Interpret. (2 pts)

Yes, here $\mathcal{S}$'s play is clearly sub-optimal. Reason: The first game's outcome might be attributed to first-move advantage, but the second match shows that not to be the deciding factor.

6.4 Is it possible for $\mathcal{A}$ *vs.* $\mathcal{A}$ to yield $(0, 0)$? Explain/Interpret. (2 pts)

Yes, and G is clearly a fair/balanced game.

6.5 Is it possible for $\mathcal{A}$ *vs.* $\mathcal{A}$ to yield $(2, -2)$ when $\mathcal{A}$ *vs.* $\mathcal{S}$ gave $(1, -1)$? (2 pts)

No, that means $\mathcal{A}$ playing the first position did worst against $\mathcal{S}$ than the optimal player (itself).

Both agents now play a third player $\mathcal{T}$.

6.6 Can $\mathcal{A}$ *vs.* $\mathcal{T}$ yield $(1,-1)$, and $\mathcal{T}$ *vs.* $\mathcal{A}$ give $(-1,1)$, when $\mathcal{S}$ *vs.* $\mathcal{T}$ yielded $(2,-2)$, and also $\mathcal{T}$ *vs.* $\mathcal{S}$ resulted in $(-2,2)$? Explain/Interpret. (4 pts)

Yes. Player $\mathcal{S}$ is sub-optimal, and so is $\mathcal{T}$. Because $\mathcal{A}$ assumes a worst-case opponent, it is conservative. Both $\mathcal{S}$ and $\mathcal{T}$ may be blind to certain plays at particular times, so play in reckless ways. But $\mathcal{A}$ would have to protect against the potential of those moves being played. In general, since $\mathcal{S}$ and $\mathcal{T}$ are sub-optimal, we can't really exclude any outcome for them.