CPE/CSC 480 ARTIFICIAL INTELLIGENCE

MIDTERM SECTION 1/2

FALL 2006

PROF. FRANZ J. KURFESS

CAL POLY, COMPUTER SCIENCE DEPARTMENT

This is the Fall 2006 midterm exam for the CPE/CSC 480 class. You may use textbooks, course notes, or other material, but you must formulate the text for your answers yourself. The use of calculators or computers is allowed for viewing documents or for numerical calculations, but not for the execution of algorithms or programs to compute solutions for exam questions. The exam time is 80 minutes.

Student Name:

Signature: **Date:**

PART 1: MULTIPLE CHOICE QUESTIONS

Mark the answer you think is correct. I tried to formulate the questions and answers so that there is only one correct answer. Each question is worth 3 points.

[30 points]

a) Which event is considered the "birth" of the field of Artificial Intelligence?

❑ The formulation of the Turing test by Alan Turing in 1950.
❑ A workshop in the summer of 1956 at Dartmouth.
❑ The development of the Lisp programming language in 1958.
❑ The victory of the Deep Blue computer system over the chess world champion, Gary Kasparov, in 1998.

b) What is a *simple reflex agent*?

❑ An agent that selects its actions based on rules describing common input-output associations.
❑ An agent that also considers relevant information from previous percepts when it makes a decision.
❑ An agent that tries to reach a desirable state.
❑ An agent that tries to reach the most desirable state, according to some metrics that measures the degree of desirability for states.

c) What is a *utility-based agent*?

❑ An agent that selects its actions based on rules describing common input-output associations.
❑ An agent that also considers relevant information from previous percepts when it makes a decision.
❑ An agent that tries to reach a desirable state.
❑ An agent that tries to reach the most desirable state, according to some metric that measures the degree of desirability for states.

d) Which of the following statements is the best characterization of a *partially observable* environment?

❑ an environment where some objects may be hidden behind other objects
❑ an environment where the sensory equipment of an agent can capture only some, but not all of the relevant aspects
❑ an environment whose properties can be described through a limited set of distinct values
❑ an environment where there is a “natural” way of grouping sequences of percepts and actions

e) In the context of search algorithms, which of the following issues is determined by the *number of steps* it takes to find a solution?

❑ completeness
❑ time complexity
❑ space complexity
❑ optimality

f) On the basis of the general search method as described in the textbook, which simple strategy results in *A* search*?

- □ append newly generated nodes at the beginning of the fringe (search queue)
- □ append newly generated nodes at the end of the fringe (search queue)
- □ insert newly generated nodes in the fringe (search queue) according to their path cost (lowest values first)
- □ insert newly generated nodes in the fringe (search queue) according to their f-cost function (lowest values first)

g) The A* algorithm sorts the nodes in the fringe according to the f-cost, which consists of two components, $f(n) = g(n) + h(n)$. Using $h(n) = 0$ results in another search method; which one is it?

- □ depth-first
- □ breadth-first
- □ uniform cost
- □ greedy best-first

h) Which of the following aspects distinguish the *IDA* algorithm* from the A* algorithm:

- □ the nodes within a given contour are explored in a depth-first manner, and the contour is expanded step by step
- □ in addition to the current path, it stores the information about the best alternative to explore in case the current path doesn't lead to the goal
- □ it drops the least promising nodes from the fringe (search queue) when it runs out of memory
- □ it utilizes contours to reduce the number of nodes to explore

i) Which of the following search methods has the worst (highest) time complexity (worst-case)?

- □ depth-first
- □ breadth-first.
- □ bi-directional
- □ A*

j) What is the most important difference between *local* and conventional (uninformed or informed) search methods?

- □ the search algorithm investigates only nodes that are reachable from the current node
- □ a solution has to be found, but the actual path to the solution is irrelevant
- □ it works better in continuous environments
- □ the analogy to landscapes with hills and valleys is more appealing than the one with trees

PART 2: SHORT QUESTIONS

In this part of the exam, you should answer the questions in about one paragraph per sub-question.

1. Compare the expected capabilities of different agent types for a *Spam classification agent*. The task of the agent is to examine all incoming email messages, and to classify them into good (ham) or bad (Spam). The minimal capability for the agent is to detect known Spam messages whose signatures are available in a database maintained by a computer security agency. If possible, the agent should also try to detect unknown Spam patterns. The most significant action of the agent is the labeling of each message; you may make assumptions about other available actions, but please indicate this if you do so. What is the least sophisticated type of agent (reflex-based, goal-based, etc.) that is reasonably capable of performing the minimal task (Spam classification of email messages)? How would this type of agent fare against unknown Spam patterns? Briefly explain your answer!

[15 points]

Agent Type	*Description: Main Properties, Advantages, Limitations*
Reflex Agent	**Main Properties:** **Advantages:** **Limitations:**
Model-Based Agent	**Main Properties:** **Advantages:** **Limitations:** □
Goal-Based Agent	**Main Properties:**

	Advantages: **Limitations:** □
Utility-Based Agent	**Main Properties:** **Advantages:** **Limitations:**
Learning Agent	**Main Properties:** **Advantages:** **Limitations:**

2. What is the basic principle of *bidirectional search*? What are some necessary conditions for its use? What are possible benefits over other search algorithms?

15 points

- Basic principle:

- Necessary conditions:

- Potential Benefits:

PART 3: PARCEL DELIVERY ROUTING (V2)

Companies like UPS, Federal Express, or the U.S. Postal Service are in the business of shipping parcels between many destinations. For each individual parcel, the problem is to find the best path from the sender's location to its destination. In the following sections, you need to apply different search algorithms to solve a simplified parcel routing problem. The graph with the cities (nodes) to be considered is shown below. The numbers on the edges indicate the actual driving distances in miles (based on AAA maps). A separate table contains the straight-line distances ("as the crow flies") between the cities for this graph (obtained in Oct. 2004 from http://www.indo.com/). This information was collected by a European collaborator, and is given in kilometers, not in miles.
For each of the algorithms, you need to do the following:

- Mark the sequence in which the nodes are visited. You can do this directly in the copy of the graph provided for each algorithm, e.g. by numbering the nodes in the order they are visited.
- Draw the corresponding search tree. It might be advisable to draw the search tree first on a separate sheet of paper, and then copy the relevant parts to the exam sheet.
- Fill in the table with the information about the search trace. Mark the newly added nodes in the fringe (frontier, queue) e.g. by underlining them. (Note: Some columns in the table are not relevant for some algorithms. The size of the table does not necessarily relate to the number of steps in the algorithm. Extend the table if necessary.)

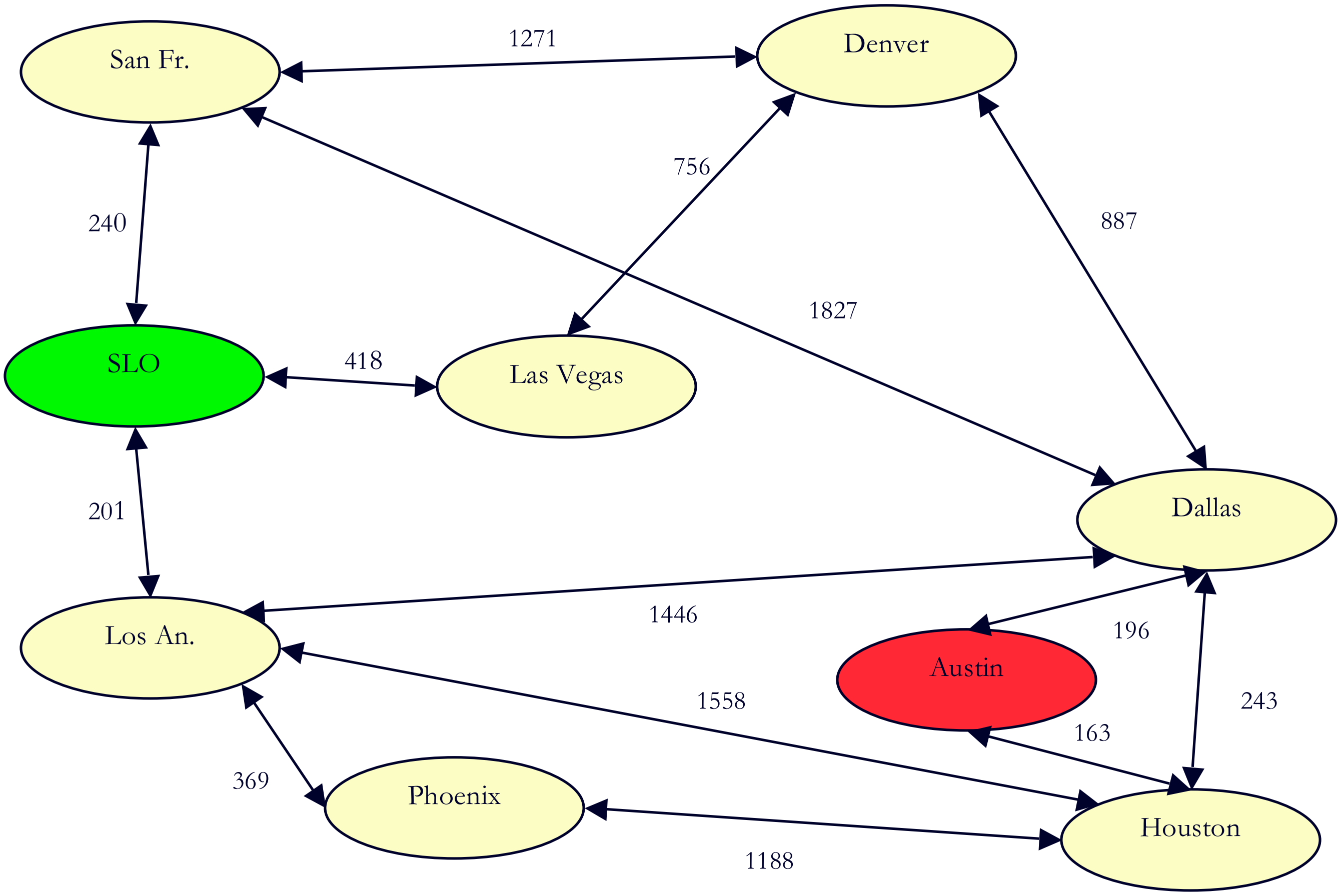
Use the alphabetical order on the full names of the nodes (not their abbreviations) to determine the order in which successor nodes are examined. You can assume that your algorithms avoid cycles by not re-visiting previously inspected nodes. For your convenience, the graph is repeated on the next pages. You can also use the off-line variation, and "jump" to the next node in the fringe.

Table: Distances "As the Crow Flies" (in kilometers)

km	*Austin*	*Dallas*	*Denver*	*Houston*	*Las Vegas*	*Los Angeles*	*Phoenix*	*San Francisco*	*San Luis Obispo*
Austin		291	1234	236	1751	1989	1400	2427	2210
Dallas	291		1062	360	1734	2014	1429	2402	2219
Denver	1234	1062		1407	991	1358	944	1549	1480
Houston	236	360	1407		1982	2225	1635	2658	2445
Las Vegas	1751	1734	991	1982		372	413	676	503
Los Angeles	1989	2014	1358	2225	372		590	554	243
Phoenix	1400	1429	944	1635	413	590		1059	813
San Francisco	2427	2402	1549	2658	676	554	1059		327
San Luis Obispo	2210	2219	1480	2445	503	243	813	327	

1. Traverse the graph from the starting point San Luis Obispo to the goal Austin according to the Uniform Cost Search method. Draw the tree and complete the table.

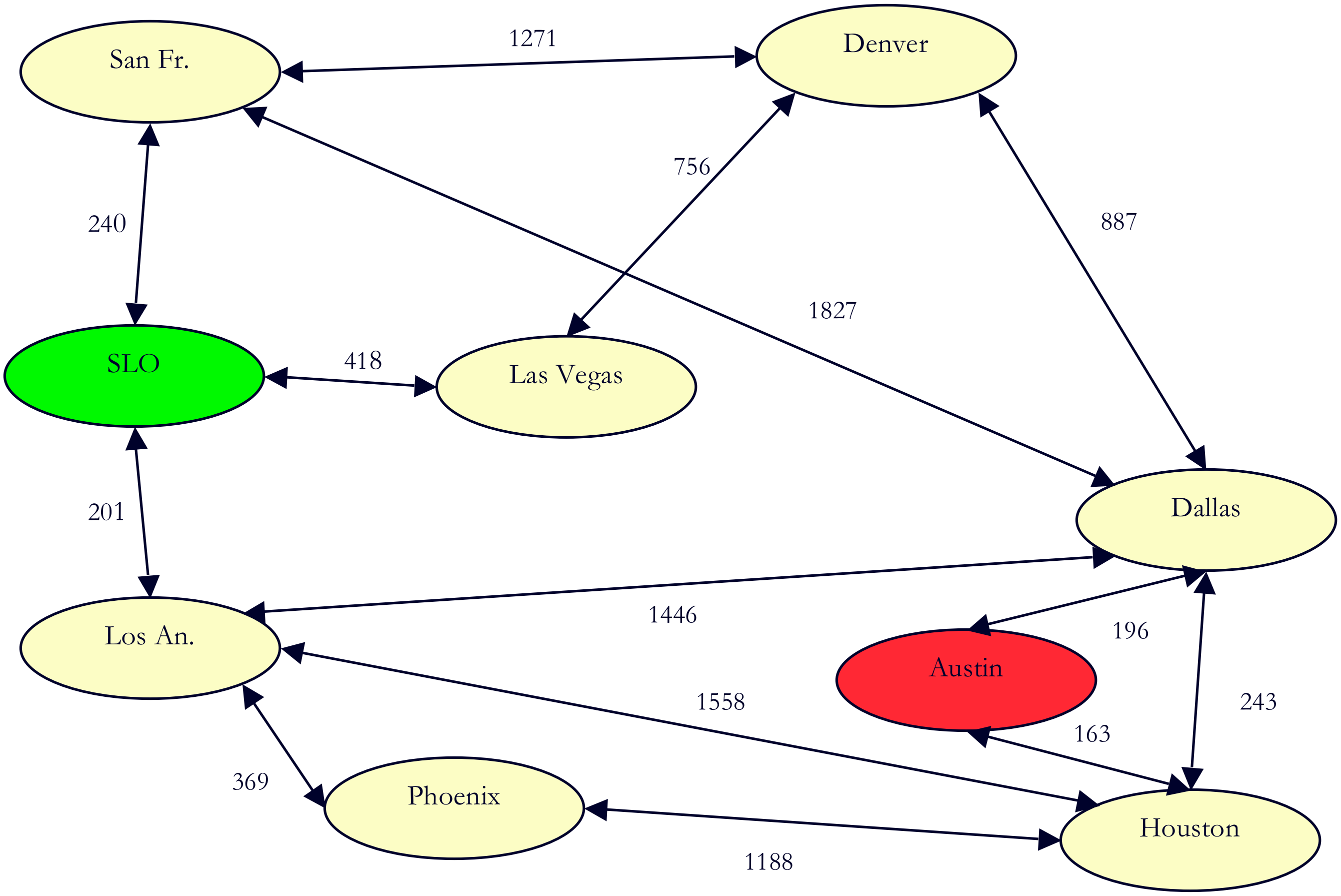
[15 points]



Step	*Current Node*	*Path Cost*	*Heuris-tic*	*F-Cost*	*Fringe (Frontier, Queue)*
0	**SLO**	0	-	-	
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

2. Traverse the graph from the starting point San Luis Obispo to the goal Austin according to the A* Search method with the straight-line distances as heuristic (see table above). Draw the tree and complete the table.

[20 points]



Step	*Current Node*	*Path Cost*	*Heuris-tic*	*F-Cost*	*Fringe (Frontier, Queue)*
0	**SLO**	0			
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

3. What is the effect of using miles for path costs and kilometers for the heuristic in the A* algorithm? Is this a serious problem (e.g. resulting in incorrect results)?

[5 points]

Total Points: