# CPE/CSC 480 ARTIFICIAL INTELLIGENCE

# MIDTERM EXAM

# FALL 2004


**PROF. FRANZ J. KURFESS**

CAL POLY, COMPUTER SCIENCE DEPARTMENT


This is the Fall 2004 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. There is only one correct answer. Each question is worth 3 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 the main difference between an intelligent *program* and an intelligent *agent*?
- ❑ A program must be compiled first, whereas an agent is directly executable.
- ❑ An agent is mobile, whereas programs are stationary.
- ❑ An intelligent agent is autonomous, and can perform tasks largely independently.
- ❑ A program is a syntactical entity, whereas an agent is a semantic entity.

c) Which statement is the best characterization of *rational agents*?
- ❑ The formulation of algorithmic descriptions of building blocks for intelligent systems.
- ❑ The formal specification of abstract reasoning mechanisms for systems that represent and manipulate knowledge.
- ❑ The construction of systems that exhibit behaviors necessary for solving tasks requiring intelligence.
- ❑ An attempt to describe the way the human mind functions.

d) Which term is also used to characterize an agent with an *internal state*?
- ❑ Rational agent.
- ❑ Model-based agent.
- ❑ Omniscient agent.
- ❑ Intelligent agent.

e) Which of the following statements is the best characterization of a *static* environment?
- ❑ an environment that does not change while the agent is deliberating
- ❑ an environment that does not change through the lifetime of an agent
- ❑ the agent does not perceive any changes during a certain time, although changes may occur that the agent is not capable of picking up with its sensors
- ❑ an environment that can only be changed through actions of the agent itself

f) In general, which of the following type of environment is the *most challenging* for agents?
- ❑ fully observable, discrete, episodic, static
- ❑ static, episodic, continuous, partially observable
- ❑ partially observable, non-deterministic, non-episodic, dynamic, continuous
- ❑ deterministic, sequential (non-episodic), dynamic, continuous

g) What is *completeness* with respect to search algorithms?
- ❑ An estimate of how many search steps it takes to find a solution.
- ❑ An estimate of how much information the algorithm needs to store for finding a solution.
- ❑ Provided that a solution exists, the algorithm will find it.
- ❑ If there is a distinction between the quality of goal states, the agent will find the best one.

h) On the basis of the general search method as described in the text book, which simple strategy results in *best-first 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 evaluation function (lowest values first)

i) Which of the following conditions guarantees the *completeness* of the *uniform-cost search algorithm*?
- ❑ the number of branches is finite, and there are no zero-cost actions
- ❑ the number of branches is finite, and the cost of every step (action) is greater than or equal to a small positive constant ε
- ❑ the path cost from the initial node to any node in the tree must be greater than zero
- ❑ the number of branches is finite

j) The *IDA* algorithm* is a variation of A* with the following properties:
- ❑ 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

## PART 2: SHORT QUESTIONS

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

1. At a first glance, the iterative deepening search method seems to involve a substantial amount of repeat work by re-constructing the search tree from scratch each time.
   a. Why is this not as bad as it seems?

   **10 points**

   b. Under which circumstances will *iterative deepening* perform much worse than *depth-first* search?

   **5 points**

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

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 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 from http://www.indo.com/), measured in kilometers.

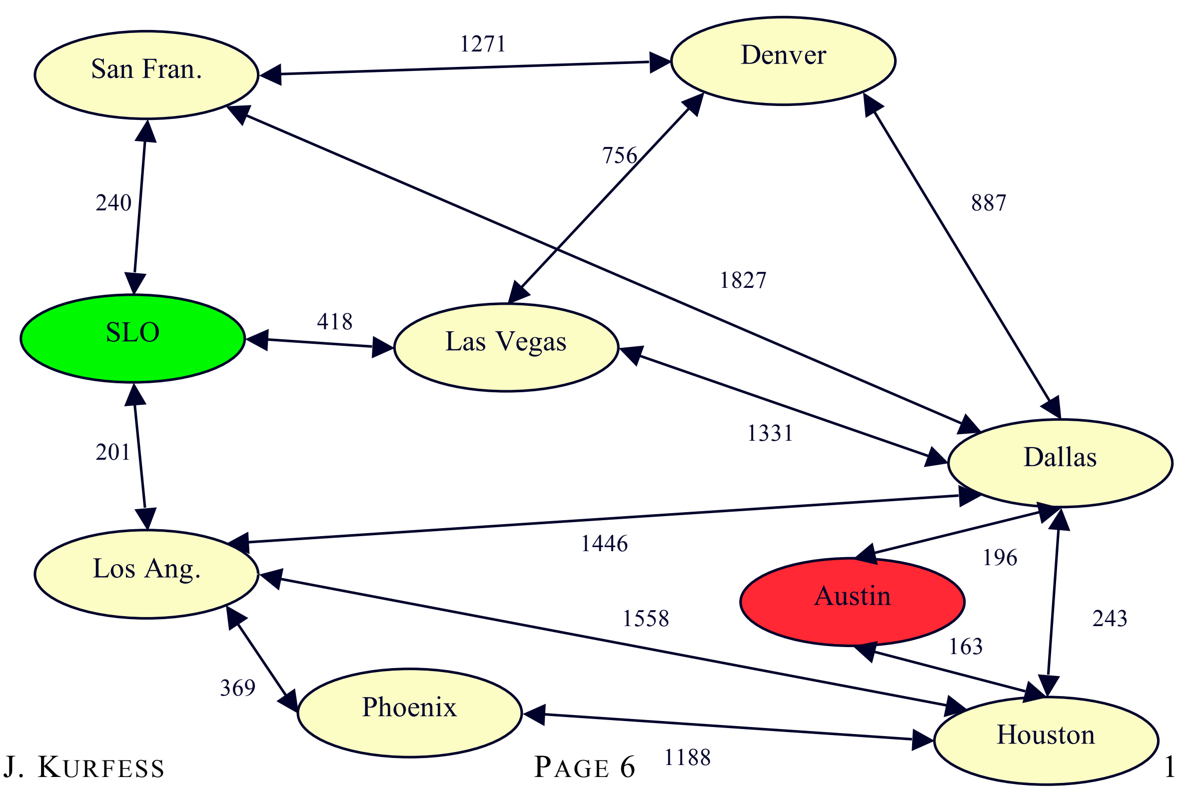
For each of the algorithms, you need to do the following:

- 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 frontier, e.g. by underlining them. (Note: Some columns in the table are 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.

a. Traverse the graph from the starting point San Luis Obispo to the goal Austin according to the Depth-First Search method. Draw the tree and complete the table.

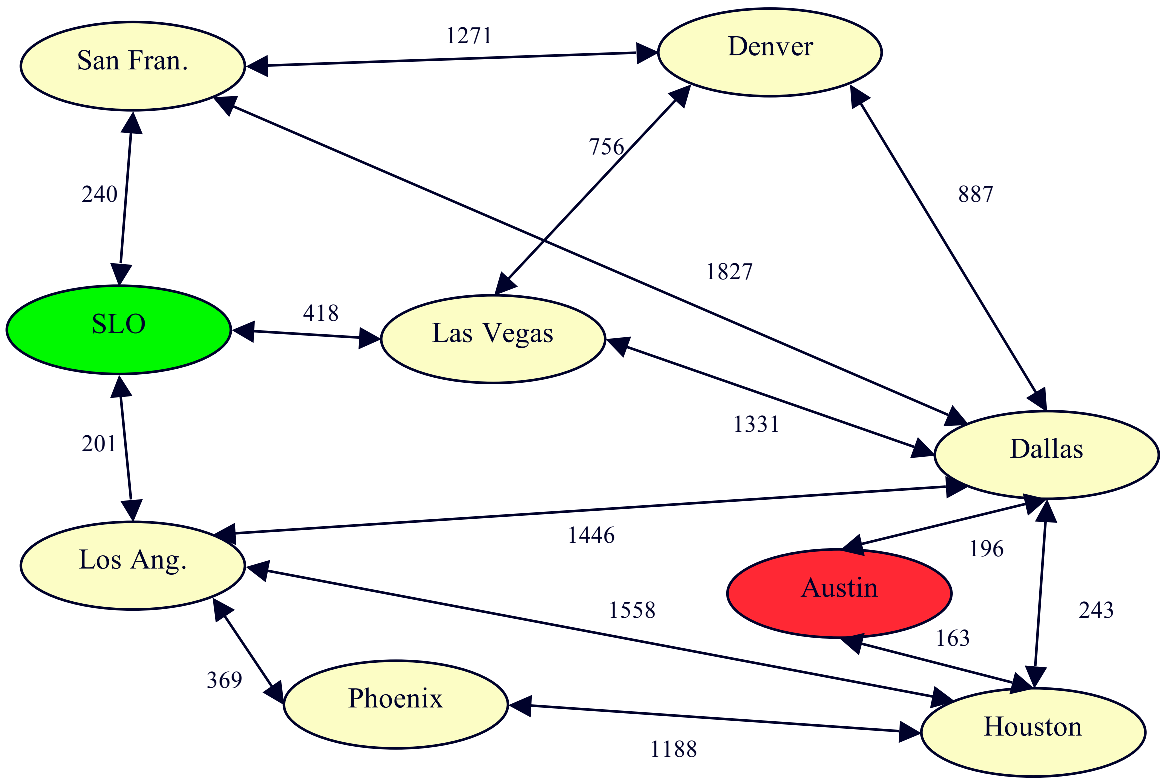
[15 points]



| *Step* | *Current Node* | *Path Cost* | *Heuris-tic* | *F-Cost* | *Frontier (Queue)* |
|---|---|---|---|---|---|
| 0 | **SLO** | | | | |
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |
| 6 | | | | | |
| 7 | | | | | |
| 8 | | | | | |
| 9 | | | | | |
| 10 | | | | | |
| 11 | | | | | |
| 12 | | | | | |
| 13 | | | | | |
| 14 | | | | | |
| 15 | | | | | |
| 16 | | | | | |
| 17 | | | | | |
| 18 | | | | | |
| 19 | | | | | |
| 20 | | | | | |

b. 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 on p. 9). Draw the tree and complete the table.

**[20 points]**



| Step | Current Node | Path Cost | Heuris-tic | F-Cost | Frontier (Queue) |
|---|---|---|---|---|---|
| 0 | **SLO** | | | | |
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |
| 6 | | | | | |
| 7 | | | | | |
| 8 | | | | | |
| 9 | | | | | |
| 10 | | | | | |
| 11 | | | | | |
| 12 | | | | | |
| 13 | | | | | |
| 14 | | | | | |
| 15 | | | | | |
| 16 | | | | | |
| 17 | | | | | |
| 18 | | | | | |
| 19 | | | | | |
| 20 | | | | | |

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* | 175 | 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 | |

c. 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:**