Practice Midterm Examination

CSCI 561 FALL2014: Artificial Intelligence

Student ID: ______________________________

Last Name: ______________________________

First Name: ______________________________

USC email: ______________________________

Instructions:

1. Date: **9/29/2014 from 5:00pm – 6:20 pm**
2. Maximum credits/points for this midterm: 100 points.
3. Credits/points for each question is indicated in the brackets [ ] before the question.
4. **<u>No books</u>** (or any other material) are allowed.
5. Attach extra sheets (available upon request) if required (write full name on each extra sheet).
6. **Write down name, student ID and usc email address.**
7. No questions during the exam.
8. **Be brief: a few words are often enough if they are precise and use the correct vocabulary studied in class.**
9. When finished raise completed exam sheets until approached by proctor.
10. **Adhere to the Academic Integrity code.**

1. [15%] General AI knowledge.

For each of the statements below, write **T** if the statement is always and unconditionally true, and write **F** if it is always false, sometimes false, or just does not make sense:

(i) [1%] __ F _ The Turing test defines the conditions under which a machine can be said to be "intelligent".

(ii) [1%] __T My office is not an accessible/observable environment.

(iii) [1%] __ F _ A contingency problem involves a nondeterministic and accessible/ observable environment.

(iv) [1%] __F_ During search, one usually applies the goal test onto newly expanded children, before queuing-up these children.

(v) [1%] __T_ If the cost of applying an operator once is always 1, then BFS is optimal.

(vi) [1%] __F_ A* is an admissible algorithm.

(vii) [1%] __F_ DFS is faster than BFS.

(viii) [1%] __T_ DFS has lower asymptotic space complexity than BFS.

(ix) [1%] __F_ When using the correct temperature decrease schedule, simulated annealing is guaranteed to find the global optimum in finite time.

(x) [1%] __ F _ Alpha-beta pruning accelerates game playing at the cost of being an approximation to full minimax.

(xi) [1%] __F_ Genetic algorithms use a step called "failover".

(xii) [1%] __F_ Hill-climbing is an entirely deterministic algorithm.

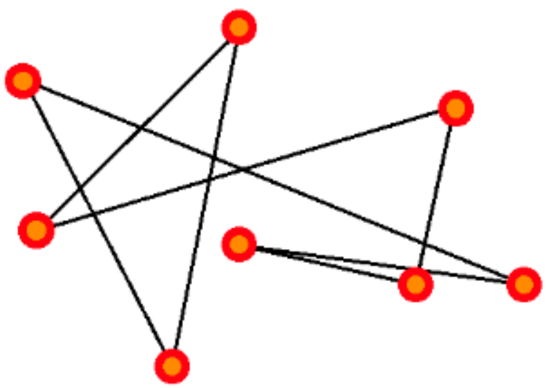
(xiii) [1%] __T_ The exact evaluation function values do not affect minimax decision as long as the ordering of these values is maintained.

(xiv) [1%] __ F _ A perfectly rational backgammon-playing agent never loses

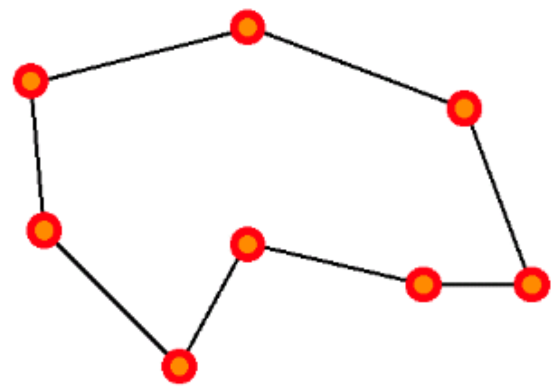
(xv) [1%] __T_ Hill climbing search is best used for problem domains with densely packed goals

2. [15%] Search Algorithms Concepts.

Let's formalize the traveling salesman problem (TSP) as a search problem. Remember that the goal in this problem is to find the shortest possible route that visits every city on a map exactly once, as exemplified below:



Suboptimal solution (long path)



Optimal solution

Assume that we have *n* cities forming a set C=$\{c_1, \ldots, c_n\}$. Also assume that you can travel from any city in that set to any other city, and that the distance between any two cities c_i and c_j is given by $d(c_i, c_j)$. Please be concise but precise in your answers. Define:

(a) [5%] A suitable representation for states:

For solution 1: permutation of all cities in the tour
Example:
For five cities keep track of the order in which they are visited
<A, B, C, D, E> can permute order to < A, B, D, C, E>

or

For solution 2: *city name (similar to Romania domain)*

(b) [2%] The initial state of the problem:

For solution 1: random permutation of all cities

or

For solution 2: randomly chosen city c_i

(c) [3%] A good goal test to use in this problem:

For solution 1: minimize the distance travelled by the tour

Or

For solution 2: minimize the distance of a complete round trip back to start city c_i after visiting all other cities once

(d) **[3%]** Good operators to use for search:

For solution 1: an example action is permute 2 cities:
<A, B, C, D, E> can permute order to < A, B, D, C, E>

or

For solution 2: example actions are travel to a city not in the path so far or
if at Nth city, then add start city again

(e) **[2%]** Which search algorithm would be the most appropriate to use here if we want to minimize the distance of the tour found?
For solution 1: Local Search - GA/SA/hill climbing, etc…

Or

For solution 2: UCS or A*

3. [30%] Comparing Search Strategies

Consider the search space on the following page, where S is the start node and G1, G2 and G3 satisfy the goal test. Arcs are labeled with the cost of traversing them and the estimated cost to a goal is reported inside nodes.

For each of the following search strategies (next page), indicate which goal state is reached (if any) and list, in order, all the states of the nodes **popped off** of the OPEN queue. When all else is equal, nodes should be removed from OPEN in alphabetical order.

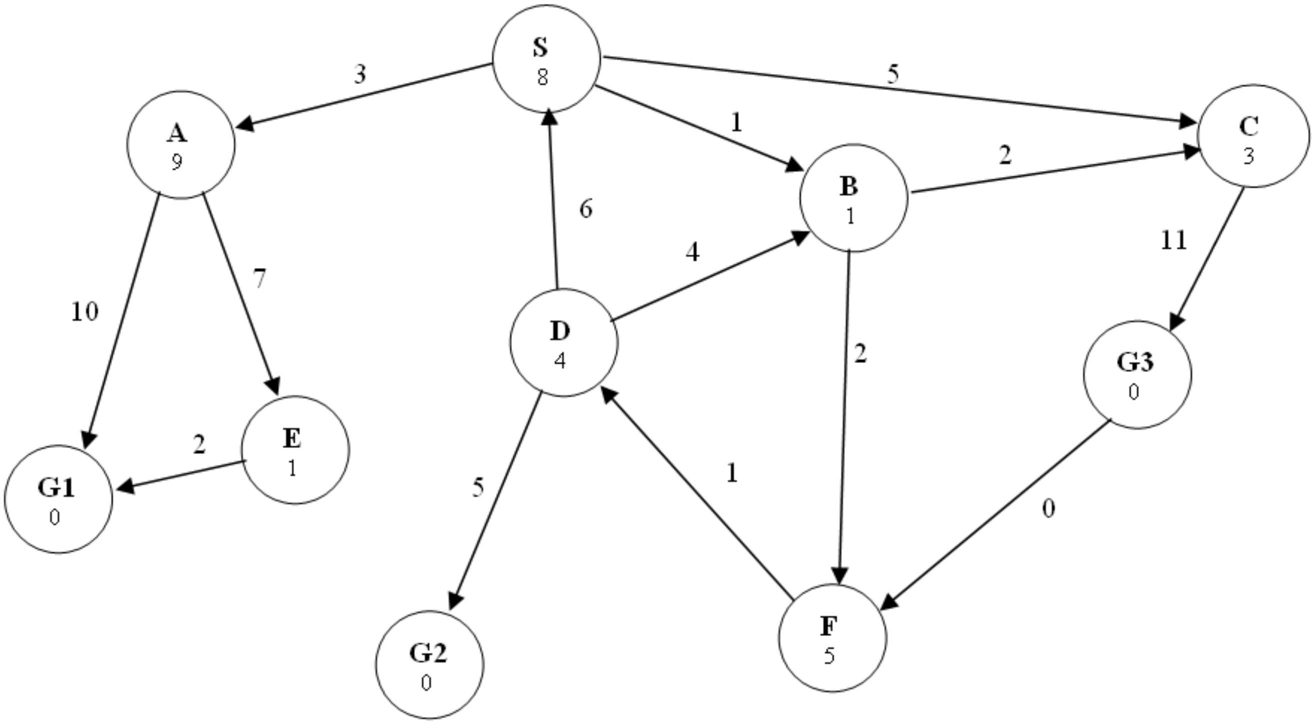
You should not expand nodes with states that have already been visited. Note how the arcs in the figure are oriented, which means that you can only go from one state to another if the arrow points from the first to the second. For example, you can go from S to A (i.e., A is a successor of S) but not from A to S (i.e., S is not a successor of A).

*Reached = expanded
**heuristic may not be admissible

(a) [10%] Depth-first search
Goal state reached: __G1___ States popped off OPEN: _____S_A_E_G1___________

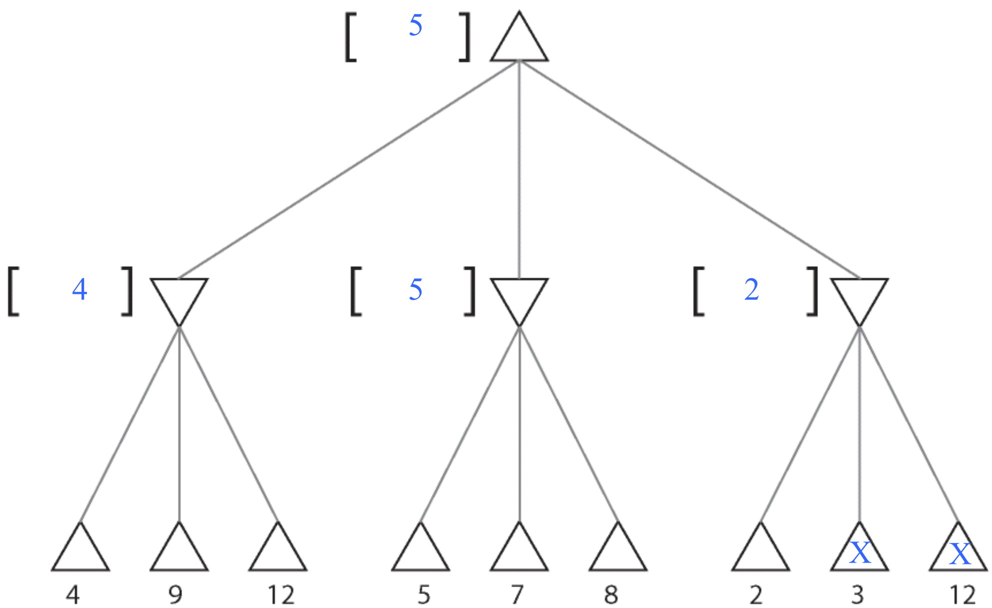
(b) [10%] Uniform cost Search
Goal state reached: ___ G2_ States popped off OPEN: _S_B_ A_C_ F_D_G2________

(c) [10%] A* Search
Goal state reached: ___ G2_ States popped off OPEN: __S_B_ C_ F_D_G2________


S
8
A
9
B
1
C
3
D
4
E
1
F
5
G1
0
G2
0
G3
0
3
5
1
2
6
4
11
10
7
2
2
5
1
0

4. [10%] Game Playing.

Consider the following game tree in which the evaluation function values are shown below each leaf node. Assume that the root node corresponds to the maximizing player. Assume that the search always visits children left-to-right.



(a) [4%] Compute the backed-up values computed by the minimax algorithm. Show your answer by writing values at the appropriate nodes in the above tree.

(b) [6%] Which nodes will not be examined by the alpha-beta pruning algorithm? Mark them on the tree above.

5. [10%] Constraint Satisfaction

Given the constraint graph for the 4-queens problem for the board below.

(a) [1%] List the variable names

Q1, Q2, Q3, Q4 corresponding to one column (row)

(b) [1%] What do the variables represent?

queens on the board

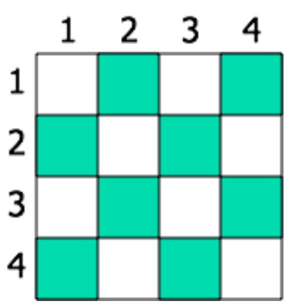
(c) [1%] What are the domain values?

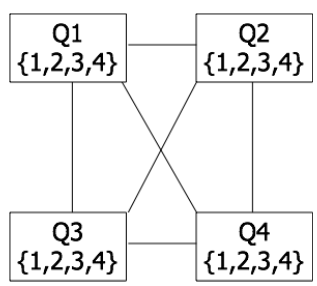
1,2,3,4 -- the row(column) of the queen

(d) [1%] How do the constraints (edges) affect the variables for this problem?

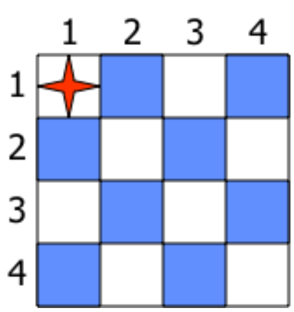
The constraints restrict the domain, so that one queen cannot attack another queen

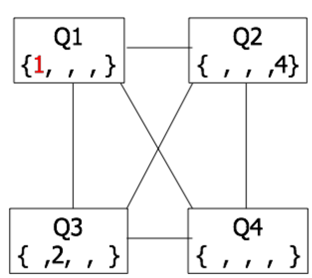
(e) [1%] How many binary constraints are there? 6





(f) [5%] Draw the constraint graph again after performing Arc Consistency on this 4 queens board, given the first queen's position is square 1.





6. [20%] AI Applications.

Circle the **best** choice for each question:

(a) [5%] Which AI application has a discrete, static environment?

a. Robocup soccer robots
b. Google self-driving car
c. IBM's Deep Blue
d. All of the above
e. None of the above

(b) [5%] Virtual agents Ada and Grace have this ability in common with IBM Watson?

a. Text-to-speech synthesis
b. Natural language processing
c. Knowledge representation
d. All of the above
e. None of the above

(c) [5%] In building a poker-playing agent, what is *most important*?

a. It should pass the Turing test
b. It should be able to hold the cards
c. It should be able to reason in a dynamic, continuous environment
d. All of the above
e. None of the above

(d) [5%] An admissible heuristic for route planning in Google Maps could be?

a. Straight-line distance
b. Manhattan distance
c. h(n)=1
d. All of the above
e. None of the above