CS 540: Introduction to Artificial Intelligence

Midterm Exam: 4:00-5:15 pm, October 25, 2016
B130 Van Vleck

CLOSED BOOK
(one sheet of notes and a calculator allowed)

Write your answers on these pages and show your work. If you feel that a question is not fully specified, state any assumptions you need to make in order to solve the problem. You may use the backs of these sheets for scratch work.

Write your name on this page and initial all other pages of this exam (in case a page comes loose during grading). Make sure your exam contains ***six*** problems on ***eight*** pages.

Name ____

Student ID ____

Problem	**Score**	**Max Score**
1	______	**20**
2	______	**20**
3	______	**10**
4	______	**10**
5	______	**25**
6	______	**15**
TOTAL	______	**100**

Problem 1 – Decision Trees (20 points)

Assume that you are given the set of labeled *training examples* below, where each feature has possible values *a, b*, and *c*. A 'name' field is included for convenience. You choose to learn a decision tree and select '-' as the default output if there are ties.

Name	F1	F2	Output
Ex1	*b*	*b*	+
Ex2	*c*	*b*	+
Ex3	*b*	*c*	+
Ex4	*a*	*b*	-
Ex5	*c*	*a*	-

a) What score would the *information gain* calculation assign to each of the features?
Be sure to show all your work (use the back of this or the previous sheet if needed).

b) Which feature would be chosen as the root of the decision tree being built? ____________
(Break ties in favor of *F1* over *F2*, i.e., in alphabetic order.)

c) Assume *F1* is chosen as the root. Show the recursive calls to ID3 below; be sure to include the *arguments* to ID3. It is fine to simply use the *names* of the examples in the recursive calls (i.e., you do no need to copy the feature values in your answer). You need only show the recursive calls; you do <u>not</u> need to show the <u>results</u> produced by these recursive calls.

Copied for your convenience:

Name	F1	F2	Output
Ex1	*b*	*b*	+
Ex2	*c*	*b*	+
Ex3	*b*	*c*	+
Ex4	*a*	*b*	-
Ex5	*c*	*a*	-

Initials: ______________________________

Problem 2 – Search (20 points)

Consider the search space below, where *S1* and *S2* are the start nodes and *G1* and *G2* satisfy the goal test. Arcs are labeled with the cost of traversing them (so lower is better) and the estimated cost to a goal is reported inside nodes.

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

You can show your work (for cases of partial credit), using the notation presented in class, on the back of the previous page.

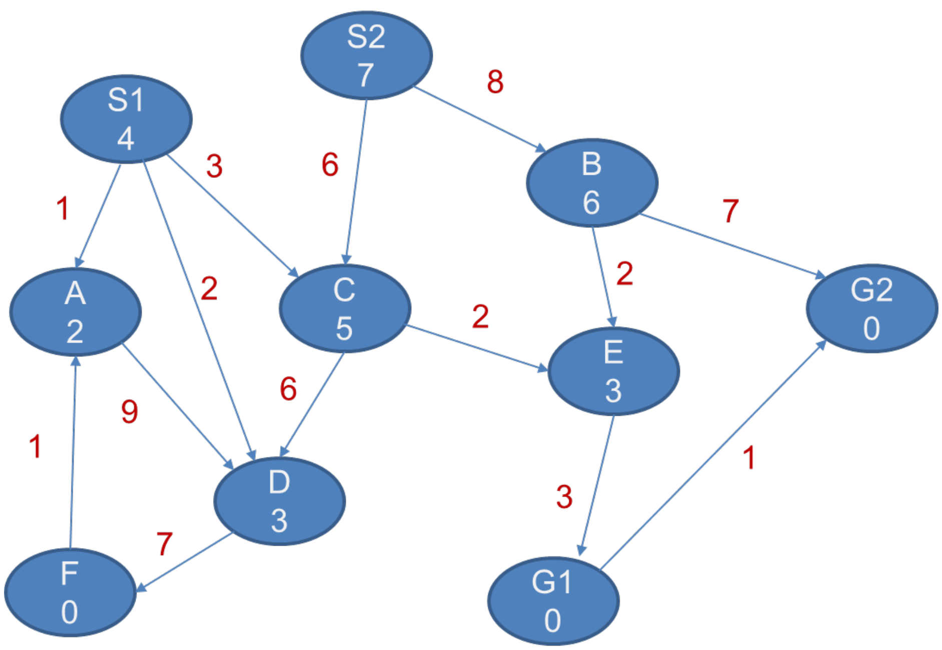
Hint: put both *S1* and *S2* in OPEN (you need to decide the order) at the beginning.

Beam Search (with beam width = 2 and f=h)

Goal state reached: _______ States popped off OPEN: ________________________________

A* (f = g + h)

Goal state reached: _______ States popped off OPEN: ________________________________



Initials: ______________________________

Problem 3 – Probabilistic Reasoning (10 points)

Using the probability table below, for full credit answer the following questions by showing the *complete world states* used to compute the answer. Write your final numeric answer on the line provided.

a) Prob(B = true ∧ C=false | A=false) ? ______________________

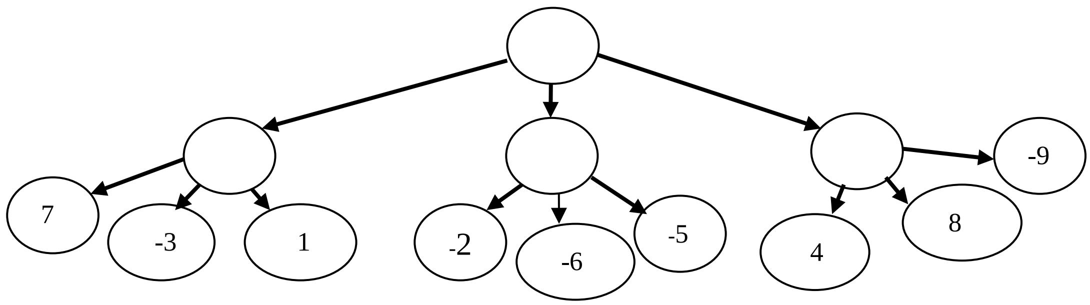
b) Prob((A=true ∧ C = false) ∨ (B = false ∧ C = true)) ? ______________________

A	**B**	**C**	**Prob**
false	false	false	0.10
false	false	true	0.15
false	true	false	0.05
false	true	true	0.11
true	false	false	0.09
true	false	true	0.10
true	true	false	0.28
true	true	true	0.12

Problem 4 – Game Playing (10 points)

a) Apply the *mini-max* algorithm to the partial game tree below, where it is the **minimizer's** turn to play and the game does not involve randomness. The values estimated by the static-board evaluator (SBE) are indicated in the leaf nodes (higher scores are better for the maximizer).

Write the estimated values of the intermediate nodes inside their circles and indicate the proper move of the minimizer by circling one of the root's outgoing arcs. Process this game tree working left-to-right.



b) List the first leaf node (if any) in the above game tree whose SBE score need not be computed: *the node whose score =* _________

Briefly explain why:

Initials: ______________________________

Problem 5 – Multiple-Choice and Short Answer Questions (25 points)

For each multiple-choice question, circle your answer. Unless stated otherwise, choose the one *best* answer. You do not need to explain your answers except for *e*.

a) Genetic algorithms require 'entities' be represented by *fixed-sized* data structures due to the need to *cross over* them. ***TRUE*** ***FALSE***

b) Two search strategies that might not find a solution even when one exists are

____________________________ and ____________________________.

c) If Uniform Cost finds a solution, is Breadth-First guaranteed to find a solution as well?

YES ***NO***

d) Assuming *h1(n)* is admissible and *h2(n)* is admissible, which of the following would be better for A*? **h3(n) = min(h1(n), h2(n))** **h4(n) = max(h1(n), h2(n))**

e) Which of the following do not belong in this group? Briefly explain your choice on the right.

   i) information gain

   ii) heuristic functions

   iii) test-set accuracy

   iv) fitness functions

f) The *singularity* refers to the case where a decision *stump* has better test-set accuracy than a decision *tree.* ***TRUE*** ***FALSE***

g) The reason ID3 typically works better in an ensemble than *k*-NN does (even if they have equivalent testset accuracies when used as 'single model' learners) is because ID3 is

____________________________________ .

h) Draw *arrows* from the phrases on the left to their best match on the right.

Test sets	Address the horizon effect
	Avoid local minima
Tune sets	Estimate future performance
	Judge information gain
	Select good settings for parameters

i) Briefly state one strength of iterative deepening over depth-first search:

____ .

Initials: ______________________________

Problem 6 – Miscellaneous Questions (15 points)

a) List <u>two</u> methods covered in class for *reducing overfitting* (short answers are fine).

____

____

b) Draw a picture that illustrates how *k-NN*, with *k*=1, partitions ('chops up') *feature space.*

c) Imagine you have a genetic algorithm population with these three bit-string entities, where the number next to each is the output of the fitness function. These are the entities remaining <u>after</u> the worst-scoring entities have been killed off.

The genetic algorithm next needs to choose two parents to *cross over*. On the blank lines below show the probability each surviving entity is chosen as the first parent.

0101101110 fitness = 3 prob chosen as first parent = ________________

1101001011 fitness = 7 prob chosen as first parent = ________________

1111101101 fitness = 1 prob chosen as first parent = ________________