Midterm Examination

CS 540-2: Introduction to Artificial Intelligence

March 9, 2017

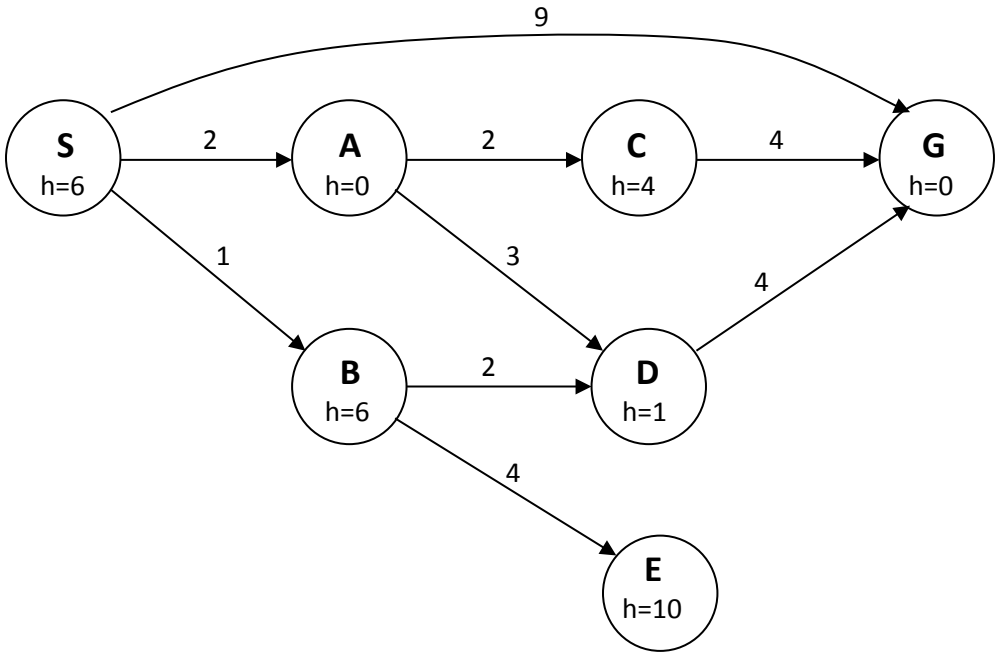
LAST NAME: SOLUTIONS

FIRST NAME: _____

Problem	Score	Max Score
1		15
2		17
3		12
4		6
5		12
6		14
7		15
8		9
Total		100

Question 1. [15] State Space Search

Consider the following state space graph where the directed arcs represent the legal successors of a node. The cost of moving to a successor node is given by the number on the arc. The value of a heuristic function, *h*, if computed at a state, is shown inside each node. The start state is **S** and the goal is **G**.



When a node is expanded, assume its children are put in the Frontier in alphabetical order so that the child closest to the front of the alphabet is removed *before* its other siblings (for all uninformed searches and for ties in informed searches). For each of the search methods below, give (i) the **sequence of nodes removed from the Frontier** (for expansion or before halting at the goal), *and* (ii) the **solution path** found.

(a) [5] Uniform-Cost *graph search* (i.e., use an Explored set)

```
Nodes removed:  S B A D C E G

Solution:  S B D G
```

(b) [5] Greedy Best-First *tree search* (i.e., *no* repeated state checking)

```
Nodes removed:  S A G
Solution:  S G
```

(c) [5] A* *tree search* (i.e., *no* repeated state checking)

```
Nodes removed:  S A D B D G
Solution:  S B D G
```

Question 2. [17] Search

(a) [2] True or False: Depth-First search using iterative deepening can be made to return the same solution as Breadth-First search.

True

(b) [2] True or False: In a finite search space containing *no* goal state and all states are reachable from the start state, A* will always explore *all* states.

True

(c) [2] True or False: The solution path found by Uniform-Cost search *may change* if we add the same positive constant, *c*, to every arc cost.

True. For example, say there are 2 paths S → A → G and S → G where cost(S,A) = 1, cost(A,G) = 1, and cost(S,G) = 3. So the optimal path is through A. If we now add 2 to each cost, the optimal path goes directly from S to G and UCS will now find this new path.

(d) [2] True or False: A* search will always expand *fewer* nodes than Uniform-Cost search.

False

(e) [9] Suppose you are using a **Genetic Algorithm**. Two individuals (i.e., candidate solutions) in the current generation are given by 8-digit sequences: 1 4 6 2 5 7 2 3 and 8 5 3 4 6 7 6 1.

(i) [3] What is the result of performing **1-point crossover** with a cross-point between the third and fourth digits?

1 4 6 4 6 7 6 1 and 8 5 3 2 5 7 2 3

(ii) [3] If there are *n* individuals in the current generation, how many crossover operations are used to produce the next generation?

If there are n individuals in the current generation, then there should be n individuals in every generation. Crossover takes two individuals from the current generation and generates two children in the next generation. Hence **n/2** crossover operations are used.

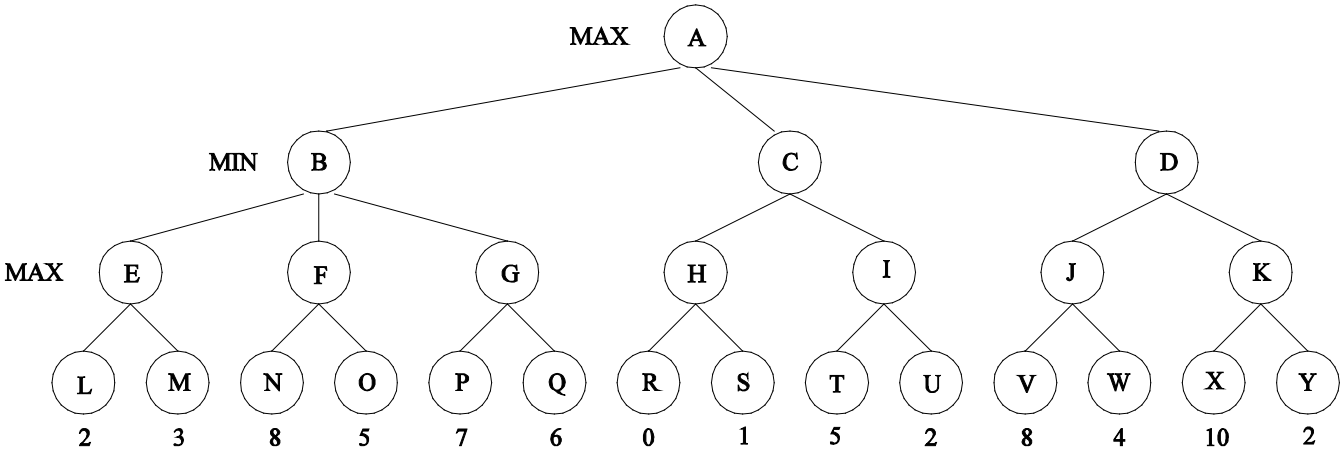
(iii) [3] A Genetic Algorithm (including crossover, mutation and natural selection) where the population size *n* = 1, would perform *most like* which of the following search methods (select *one*):

a. Simulated annealing
b. (Greedy) Hill-climbing
c. A random walk
d. Nothing will change (i.e., the initial individual will stay the same)

Since n = 1, crossover will do nothing since the one parent will be selected twice and the generated child will be the same as its parents. Then, mutation is applied to the child copy of the parent, which will randomly modify the child with a small probability, causing it to have either higher or lower fitness than its parent. So, the algorithm will do a **random walk** in the space of individuals.

Question 3. [12] Game Playing

Consider the following game tree in which the root corresponds to a MAX node, children are visited left to right, and the values of a static evaluation function are given at the leaves.



(a) [4] What are the **minimax values** computed at each node in this game tree? Write your answers to the *LEFT* of each node in the tree above.

E=3, F=8, G=7, H=1, I=5, J=8, K=10, B=3, C=1, D=8, A=8

(b) [4] If the values at all the leaf nodes in the above tree were changed so as to achieve the *maximum possible pruning* by **Alpha-Beta Pruning** algorithm, which nodes would *not* be examined in this best case scenario? Cross out the nodes not examined in the tree above.

O, Q, I, T, U, K, X, Y

(c) [4] In the game tree above assume the three MIN nodes are replaced by CHANCE nodes, and at a CHANCE node each successor move is equally likely. (For example, at node B there is a 1/3 chance of moving to each of its successors.) What are the **expectimax values** computed at each node in this game tree? Write your answers to the *RIGHT* of each node in the tree above.

E=3, F=8, G=7, H=1, I=5, J=8, K=10, **B=6, C=3, D=9, A=9**

Question 4. [6] Hierarchical Agglomerative Clustering

You are given the following table of distances between all pairs of five clusters:

	A	**B**	**C**	**D**	**E**
A	0	1075	2013	2054	996
B	1075	0	3273	2687	2037
C	2013	3273	0	808	1307
D	2054	2687	**808**	0	1059
E	996	2037	1307	1059	0

(a) [3] Which pair of clusters will be merged into one cluster at the next iteration of hierarchical agglomerative clustering using **single linkage**?

`The closest pair is C and D with distance 808`

(b) [3] What will the new values be in the resulting table corresponding to the four new clusters? Include the cluster names in the first row and first column; if clusters *x* and *y* were merged, name that cluster *x+y*.

	A	**B**	**C+D**	**E**
A	0	1075	**2013**	996
B	1075	0	**2687**	2037
C+D	**2013**	**2687**	**0**	**1059**
E	996	2037	**1059**	0

Question 5. [12] *k*-Means Clustering

Given a set of three points, -2, 0, and 10, we want to use *k*-Means Clustering with *k* = 2 to cluster them into two clusters.

(a) [8] If the initial cluster centers are *C*1 = -4.0 and *C*2 = 1.0, show two iterations of *k*-Means Clustering, indicating at each iteration which points belong to each cluster and the coordinates of the two new cluster centers. In other words, fill in the tables below.

Iteration 1:

Point	**Cluster** (*C*1 or *C*2)
-2	***C*1**
0	***C*2**
10	***C*2**

New cluster centers:

*C*1 = **-2**	*C*2 = **5**

Iteration 2:

Point	**Cluster** (*C*1 or *C*2)
-2	***C*1**
0	***C*1**
10	***C*2**

New cluster centers:

*C*1 = **-1**	*C*2 = **10**

(b) [2] True or False: *k*-Means Clustering is guaranteed to find the *same* final clusters for the above three points, no matter what the initial cluster center values are.

False. For example, if initially $c_1 = 0$ and $c_2 = 50$ then all three points will be assigned to cluster 1 and no points will be in cluster 2. Then, updating the cluster centers, they become $c_1 = (2+0+10)/3 = 6$ and $c_2 = 50$ (no change because there are no points). These two cluster centers and their associated points do not change in the next iteration, so the final clustering has all three points in one cluster and none in the other.

(c) [2] True or False: A good way to pick the best number of clusters, *k*, used for *k*-Means clustering is to try multiple values of *k* and select the value of *k* at the "elbow" of the monotonically decreasing curve measuring Distortion as a function of *k*.

True

Question 6. [14] *k*-Nearest Neighbors

(a) [8] You are given a Training set of five real-valued points and their 2-class classifications (+ or −):

(1.5, +), (3.2, +), (5.4, −), (6.2, −), (8.5, −).

(i) [3] What is the predicted **class** for a test example at point 4.0 using 3-NN?

Closest three neighbors are 3.2, 5.4 and 6.2 so majority class is –

(ii) [3] What is the **decision boundary** associated with this Training set using 3-NN? (Hint: The boundary is defined by a single real value.)

The decision boundary will be in between the points 3.2 and 5.4, at the midpoint between 1.5 and 6.2, which is (1.5 + 6.2)/2 = **3.85** Points to the left of this decision boundary will be classified + using 3-NN, and points to the right of 3.85 will be classified –.

(iii) [2] True or False: For *any* 2-class, linearly-separable Training set (e.g., the one given above), a 3-NN classifier will always have 100% accuracy on the Training set.

False. For example, if the training set is (1, +), (2, +), (3, –), (4, –), (7, –), then 3-NN will misclassify point 3 as +.

(b) [6] Say we have a Training set consisting of 30 positive examples and 10 negative examples where each example is a point in a two-dimensional, real-valued feature space.

(i) [3] What will the classification **accuracy** be on the Training set with 1-NN?

Since k-NN just memorizes the points and their classes in the training set, if we use as the testing set the same examples, then each testing set example will have its 1-NN be the point itself in the training set. So, the classification accuracy will be **100%**.

(ii) [3] What will the classification **accuracy** be on the Training set with 40-NN?

In this case all training examples are used, so for any test example, the majority class, +, will be the classification. So, in the training set 30 will be classified correctly and 10 will be classified incorrectly, so the classification accuracy will be **75%**.

Question 7. [15] Decision Trees

(a) [10] Suppose you are using the Decision Tree Learning algorithm to learn a 2-class classification variable, *C*, and you must decide which attribute to assign to a node in the tree. At this node there are 100 examples; 30 are positive and 70 are negative. If attribute *A* is selected, its first child will get 18 positive and 22 negative examples, and its second child will get 12 positive and 48 negative examples. Use log 0.1 = -3.32, log 0.2 = -2.32, log 0.3 = -1.74, log 0.33 = -1.59, log 0.4 = -1.32, log 0.45 = -1.15, log 0.5 = -1.0, log 0.55 = -0.86, log 0.6 = -0.74, log 0.67 = -0.58, log 0.7 = -0.51, and log 0.8 = -0.32, log 0.9 = -0.15, and log 1 = 0, where all logs are to base 2.

(i) [5] What is the **entropy** of *C*, i.e., *H*(*C*), at the node? You may give your answer as either a number or an expression in terms of logs and fractions.

$H(C) = H(30/100, 70/100)$

$= -(30/100 * \log_2 30/100 + 70/100 * \log_2 70/100)$

$= -((.3)(-1.74) + (.7)(-.51))$

$= 0.879$

(ii) [5] What is the **conditional entropy** (aka Remainder) of choosing attribute *A* at the node? That is, compute *H*(*C* | *A*). You may give your answer as either a number or an expression in terms of logs and fractions.

$H(C|A) = 40/100 * H(18/40, 22/40) + 60/100 * H(12/60, 48/60)$

$= .4 * [-18/40 * \log 18/40 - 22/40 * \log 22/40]$

$+ .6 * [-12/60 * \log_2 12/60 - 48/60 * \log_2 48/60]$

$= .4 * [-((.45)(-1.15) + (.55)(-.86))]$

$+ .6 * [-((.2)(-2.32) + (.8)(-.32))]$

$= (.4)(.9905) + (.6)(.72)$

$= 0.8282$

(b) [2] True or False: The **entropy** at the nodes along any path from the root of a Decision Tree to a leaf is always monotonically non-increasing.

False

(c) [3] Which of the following strategies can help reduce **overfitting** in decision trees (select 0 or more correct choices):

(i) Perform some kind of tree pruning
(ii) Make sure each leaf node contains only examples with the same class
(iii) Enforce a minimum number of examples at each leaf node
(iv) Use less training data

(i) and (iii)

Question 8. [9] Random Forests

(a) [2] True or False: Random Forests are better than Decision Trees (built using the basic Decision-Tree-Learning algorithm) in terms of avoiding overfitting the Training set.

True

(b) [4] What *two* (2) techniques are used to construct different decision trees for use in a Random Forest classifier?

1. Use Bagging to construct different training sets for each tree by **randomly selecting a subset of examples** with replacement from the total number of examples in the Training set.

2. Use Randomized Node Optimization, which considers only a **randomly selected subset of the attributes** as candidates at each node of each tree.

(c) [3] How are the results from *k* decision trees in a Random Forest classifier *combined* to determine the output classification for an example in the Testing set?

The **mode** (**majority**) of the classifications, i.e., the most frequently occurring of the *k* classifications.