

**Midterm Examination**  
**CS540-2: Introduction to Artificial Intelligence**

March 15, 2018

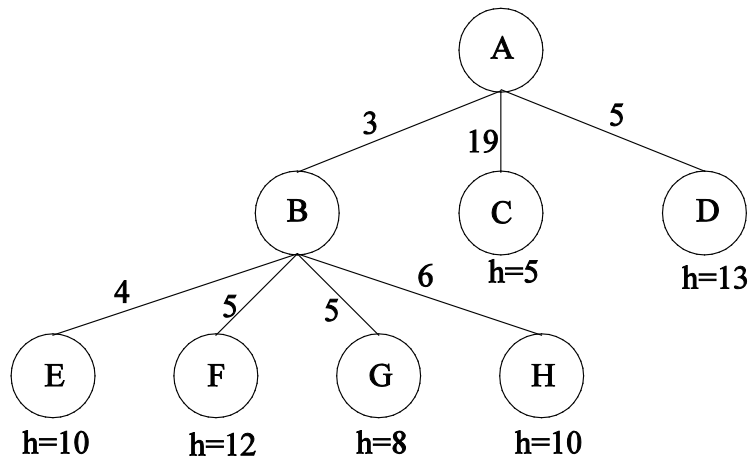
LAST NAME: \_\_\_\_\_

FIRST NAME: \_\_\_\_\_

Problem	Score	Max Score
1	_____	12
2	_____	13
3	_____	9
4	_____	11
5	_____	8
6	_____	13
7	_____	9
8	_____	16
9	_____	9
Total	_____	100

**Question 1. [12] Search**

Consider the following *search tree* produced after expanding nodes A and B, where each arc is labeled with the cost of the corresponding operator, and the leaves are labeled with the value of a heuristic function,  $h$ . For uninformed searches, assume children are expanded left to right. In case of ties, expand in alphabetical order.



Which *one* node will be expanded *next* by each of the following search methods?

(a) [3] Depth-First search

(b) [3] Greedy Best-First search

(c) [3] Uniform-Cost search

(d) [3] A\* search

**Question 2. [13] Search Short Answer**

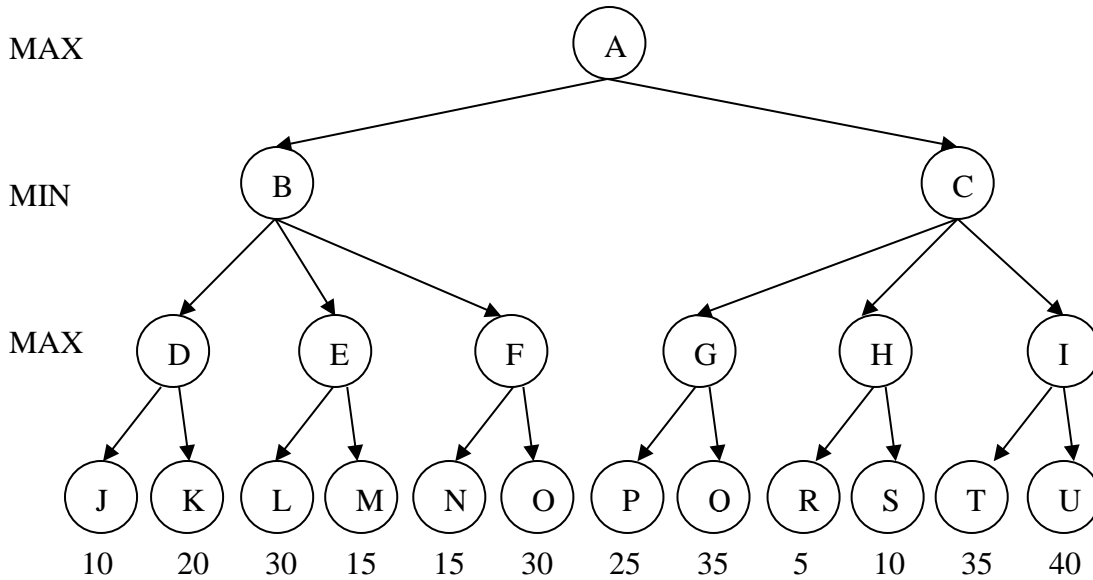
- (a) [2] True or False: *Greedy Best-First search* using an admissible heuristic is guaranteed to find an optimal solution.
- (b) [2] True or False: *Algorithm A search* using the heuristic  $h(n) = c$  for some fixed constant  $c > 0$  is guaranteed to find an *optimal* solution.
- (c) [2] True or False: If a heuristic is consistent, it is also admissible.
- (d) [2] True or False: If  $h_1$  and  $h_2$  are both admissible heuristics, it is always better to use the heuristic  $h_3(n) = \max(h_1(n), h_2(n))$  rather than the heuristic  $h_4(n) = \min(h_1(n), h_2(n))$ .
- (e) [2] True or False: *Beam search* with a beam width  $W = 3$  and an admissible heuristic is *not* guaranteed to find a solution (optimal or not) when one exists.
- (f) [3] Say we have a state space where all arc costs are 1 but we don't have a heuristic function. We want to use a space-efficient search algorithm (in terms of the maximum number of nodes stored at any point during the search in Frontier) but also want to guarantee that we find an optimal solution. Which *one* of the following search methods would be best to use in this situation?
- (i) Breadth-First Search
  - (ii) Depth-First Search
  - (iii) Iterative-Deepening Search
  - (iv) Uniform-Cost Search

**Question 3. [9] Local Search**

- (a) [3] The 8-Queens problem can be represented as an 8-digit vector where the  $i^{\text{th}}$  digit indicates the row number of the queen in the  $i^{\text{th}}$  column. Given a state defined by such a vector, a successor function could be defined by randomly selecting a new row for a single queen. Which *one* of the following actions in a *Genetic Algorithm* is *most similar* to the operation defined by this successor function?
- (i) Selection
  - (ii) Fitness
  - (iii) Crossover
  - (iv) Mutation
- (b) [3] How would *Simulated Annealing* perform if the temperature,  $T$ , is always very close to 0? Select *one* of the following:
- (i) A random walk
  - (ii) Hill-Climbing
  - (iii) It would halt immediately
  - (iv) It would never halt
- (c) [3] In *Simulated Annealing* if the current state's evaluation function value is some positive value,  $c$ , and the selected neighbor state's value is also equal to  $c$ , how does the probability of moving to this neighbor change as the temperature,  $T$ , is gradually decreased? Assume we are maximizing the evaluation function.
- (i) It doesn't change, it's always 0
  - (ii) It decreases
  - (iii) It increases
  - (iv) It doesn't change, it's always 1

**Question 4. [11] Game Playing**

Consider the following game tree. The root is a maximizing node, and children are visited left to right.



- (a) [6] *Ignoring the values given above at the leaf nodes, could there possibly exist some set of values at the leaf nodes in the above tree so that Alpha-Beta pruning would do the following. Answer either Yes or No.*
- (i) [2] Pruning occurs at the arc from D to K.
  - (ii) [2] Pruning occurs at the arc from E to M.
  - (iii) [2] Pruning occurs at the arc from C to G.
- (b) [3] In the tree above assume that the root node is a MAX node, nodes B and C are MIN nodes, and the nodes D, ..., I are **not** MAX nodes but instead are positions where a fair coin is flipped (so going to each child has probability 0.5). What is the *Expectimax* value at node A?
- (c) [2] True or False: When using *Expectimax* to compute the best move at the root, rescaling the values at *all* leaf nodes by multiplying them *all* by 10 may result in a different move at the root.

**Question 5. [8] Hierarchical Agglomerative Clustering (HAC)**

(a) [6] Given three clusters,  $A$ ,  $B$  and  $C$ , containing a total of six points, where each point is defined by an integer value in one dimension,  $A = \{0, 2, 6\}$ ,  $B = \{3, 9\}$  and  $C = \{11\}$ , which two clusters will be merged at the *next iteration* of HAC when using Euclidean distance and

(i) [3] Single linkage?

- i. Merge  $A$  and  $B$
- ii. Merge  $A$  and  $C$
- iii. Merge  $B$  and  $C$

(ii) [3] Complete linkage?

- i. Merge  $A$  and  $B$
- ii. Merge  $A$  and  $C$
- iii. Merge  $B$  and  $C$

(b) [2] True or False: HAC requires the user to specify the number of clusters,  $k$ , when building the dendrogram for a set of data.

**Question 6. [13] K-Means Clustering**

- (a) [2] True or False: *K-Means Clustering* is guaranteed to converge (i.e., terminate).
- (b) [2] True or False: A good way to pick the number of clusters,  $k$ , used for  $k$ -Means clustering is to try multiple values of  $k$  and choose the value that minimizes the distortion measure.
- (c) [9] Consider performing *K-Means Clustering* on a one-dimensional dataset containing four data points: {5, 7, 10, 12} using  $k = 2$ , Euclidean distance, and the initial cluster centers are  $c_1 = 3.0$  and  $c_2 = 13.0$ .
- (i) [3] What are the initial cluster assignments? (That is, which examples are in cluster  $c_1$  and which examples are in cluster  $c_2$ ?)
- (ii) [3] What is the value of *distortion* for the clusters computed in (i)?
- (iii) [3] What are the new cluster centers after making the assignments in (i)?

**Question 7. [9] K-Nearest Neighbor ( $k$ -NN) Classifier**

- (a) [3] Consider a set of five training examples given as  $((x_i, y_i), c_i)$  values, where  $x_i$  and  $y_i$  are the two attribute values (positive integers) and  $c_i$  is the binary class label:  $\{((1, 1), -1), ((1, 7), +1), ((3, 3), +1), ((5, 4), -1), ((2, 5), -1)\}$ . Classify a test example at coordinates  $(3, 6)$  using a  $k$ -NN classifier with  $k = 3$  and Manhattan distance defined by  $d((u, v), (p, q)) = |u - p| + |v - q|$ . Your answer should be either +1 or -1.
- (b) [2] True or False: The decision boundary associated with a 1-NN classifier defined by the above five examples consists of a sequence of line segments that are each parallel to exactly one of the two axes associated with the two attributes,  $x$  and  $y$ .
- (c) [2] True or False: The results of a general  $k$ -NN classifier that uses Euclidean distance *may* change if we multiply each example's attribute value by 0.5.
- (d) [2] True or False: As the value of  $k$  used in a  $k$ -NN classifier is incrementally increased from 1 to  $n$ , the total number of training examples, the classification accuracy on the *training set* will always increase.



**Question 8. [16] Decision Trees**

- (a) [4] What is the *conditional entropy*,  $H(C | A)$ , for the following set of 8 training examples. Use  $\log 0.1 = -3.3$ ,  $\log 0.2 = -2.3$ ,  $\log 0.25 = -2.0$ ,  $\log 0.3 = -1.74$ ,  $\log 0.33 = -1.58$ ,  $\log 0.4 = -1.3$ ,  $\log 0.45 = -1.15$ ,  $\log 0.5 = -1.0$ ,  $\log 0.55 = -0.85$ ,  $\log 0.6 = -0.7$ ,  $\log 0.67 = -0.58$ ,  $\log 0.7 = -0.5$ ,  $\log 0.75 = -0.4$ ,  $\log 0.8 = -0.3$ ,  $\log 0.9 = -0.15$ ,  $\log 1 = 0$ , where all logs are base 2.

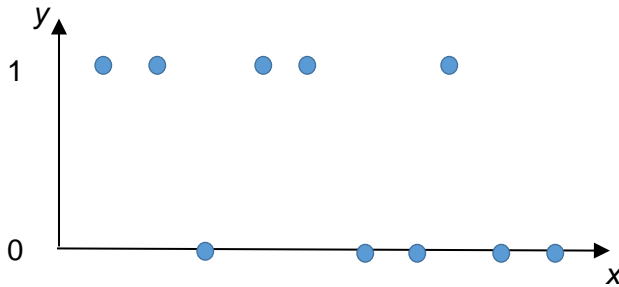
<i>A</i>	<i>B</i>	<i>C</i>
T	F	T
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F

- (b) [3] In a problem where each example has  $n$  real-valued attributes, where each attribute can be split at 2 possible thresholds, to select the best attribute for a decision tree node at depth  $k$ , where the root is at depth 0, how many *conditional entropies* must be calculated?

- (c) [2] True or False: The Information Gain at the *root* node of any Decision Tree must always be at least as large as the Information Gain at any other node in that tree.
- (d) [2] True or False: Two different decision trees (constructed using different methods) that both correctly classify *all* the examples in a given training set will also classify any other testing example in the same way (i.e., both trees will predict the same class for any other example).
- (e) [3] If a decision tree has  $m$  non-leaf nodes and  $n$  leaf nodes, how many different candidate prunings are computed and compared to the current tree when performing one iteration of the decision tree pruning algorithm? Do not count making the root a leaf node.
- (i)  $m + n - 1$
  - (ii)  $m$
  - (iii)  $m - 1$
  - (iv)  $n$
  - (v)  $n - 1$
- (f) [2] True or False: A decision tree containing only a root node, two leaves, and an associated threshold can be built from a training set that contains the four examples:  $((3, 3) +1)$ ,  $((3, 1) +1)$ ,  $((1, 3), +1)$  and  $((1, 1), -1)$ , where  $((x, y), c)$  means the example has two real-valued attributes,  $A=x$  and  $B=y$ , and  $c$  is the binary class label,  $+1$  or  $-1$ , and the training set accuracy is 100%.

**Question 9. [9] Cross-Validation**

Suppose you are using a Majority Classifier on the following training set containing 10 examples where each example has one real-valued feature,  $x$ , and a binary class label,  $y$ , with value 0 or 1. Define this Majority Classifier to predict the class label that is in the *majority in the training set*, regardless of the input value. In case of ties, predict class 1.



(a) [3] What is the *training set accuracy*?

- (i) 0%
- (ii) 10%
- (iii) 50%
- (iv) 90%
- (v) 100%

(b) [3] What is the *Leave-1-Out Cross-Validation accuracy*?

- (i) 0%
- (ii) 10%
- (iii) 50%
- (iv) 90%
- (v) 100%

(c) [3] What is the *2-fold Cross-Validation accuracy*? Assume the leftmost 5 points (i.e., the 5 points with smallest  $x$  values) are in one fold and the rightmost 5 points are in the second fold.

- (i) 0%
- (ii) 20%
- (iii) 50%
- (iv) 80%
- (v) 100%