THE UNIVERSITY OF PENNSYLVANIA

SAMPLE EXAM WITH
ANSWERS – Given in Fall,
2015
POINT COUNTS NOT
ACCURATE
Points Possible: 100

CIS 521

INTRODUCTION TO ARTIFICIAL INTELLIGENCE

Midterm I

(Time allowed: 80 minutes)

Section		Points	Max
1	True/False Multiple Choice		15
2	Search		15
3	Heuristics		15
4	Adversarial search		10
5	Constraint Satisfaction		15
6	Naïve Bayes	NOT APPLICABLE	15
	TOTAL		85

Question 1. [15] True-False & Multiple Choice

a) [2] In Python, what does the last command in this sequence print?

```
> x = [[(x, y) for x in 1,2,3] for y in 1,2,3]
> len(x)
```

(a) 1

(b) 3

(c) 6

(d) 9

b) [1] TRUE or **FALSE**: Depth-first search is complete.

c) [1] TRUE or **FALSE**: Depth-first search has, in general, much lower space complexity than iterative deepening.

d) [1] **TRUE** or FALSE: In estimating the travel distance between two locations, Euclidean distance is a consistent and admissible heuristic.

e) [1] TRUE or **FALSE**: A* search uses heuristics to prune the search space so that the use of space is effectively O(bd).

f) [1] **TRUE** or FALSE: Iterative deepening is complete in that it is guaranteed to halt if there is a solution path to the goal.

g) [1] **TRUE** or FALSE: Uniform-cost search is a special case of A* search.

h) [2] Which of the following is true?

a. All admissible heuristics are consistent

b. All consistent heuristics are admissible

c. All of the above

d. None of the above

i) [2] Given the joint probabilities in Table 2 below, what is the probability of having a cavity if there's no toothache?

	toothache	¬ toothache
cavity	0.12	0.4
¬ cavity	0.1	0.38

Table 2: Joint probabilities

(a) 0.12
(b) 0.4
(c) 0.4/(0.12 + 0.4)
(d) none of the above

j) [2] What assumption is made in deriving the Naive Bayes model? (c_i is the class random variable and x_i is the feature random variable; Note that there are N classes and N features)

(a) $\mathbf{P(x_1 \ldots x_N \mid c_i) = P(x_1 \mid c_i)\, P(x_2 \mid c_i) \ldots P(x_N \mid c_i)}$

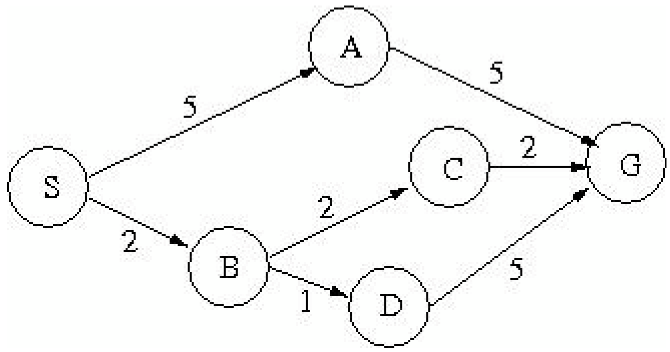
(b) $P(x_1 \ldots x_N, c_1 \ldots c_N) = P(x_1, c_1) \ldots P(x_N, c_N)$

(c) $P(c_1 \ldots c_N \mid x_i) = P(c_1|x_i)\, P(c_2|x_i) \ldots P(c_N|x_i)$

(d) $P(x_1 \ldots x_N, c_1 \ldots c_N) = P(c_1)\, P(x_1|c_1) P(c_2|x_1, c_1) \ldots P(x_N|x_1 \ldots x_{N-1}, c_1 \ldots c_N)$

Question 2. [15] Search

Consider the following search space where we want to find a path from the start state S to the goal state G. The table shows three different heuristic functions *h1*, *h2*, and *h3*.



Node	*h1*	*h2*	*h3*
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

a) [3] What solution path is found by Greedy Best-first search using *h2*? Break ties alphabetically.

S, A, G

b) [12] Give the three solution paths found by algorithm A* using each of the three heuristic functions, respectively. Break ties alphabetically.

h1: (Just Uniform Cost Search) Sequence of nodes expanded: S, B, D, C, A, G Solution path: S, B, C, G

h2: S, B, C, G expanded; solution: S, B, C, G

h3: S, B, D, G expanded; solution: S, B, D, G

Question 3. [15]. Heuristics

(a) Consider the 8-puzzle in which there is a 3 x 3 board with eight tiles numbered 1 through 8. The goal is to move the tiles from a start configuration to a goal configuration, where a move consists of a horizontal or vertical move of a tile into an adjacent position where there is no tile. Each move has cost 1.

(i) [5] Is the heuristic function defined by $h = \sum_{i=1}^{8} \alpha_i d_i$ **admissible**, where d_i is the number of vertical plus the number of horizontal moves of tile *i* from its current position to its goal position assuming there are no other tiles on the board, and $0 \le \mathbf{a}_i \le 1$ is a constant weight associated with tile *i*? Explain briefly why or why not.

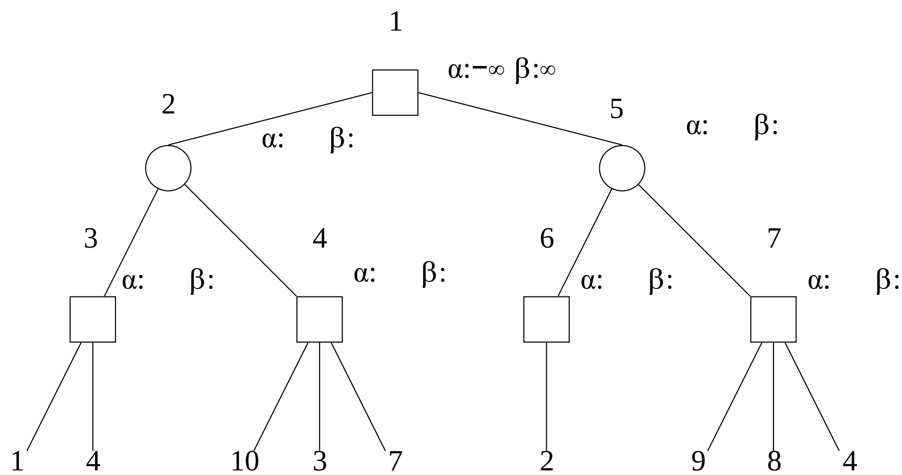
Yes, it is admissible because each d_i is a lower bound on the number of moves to get each tile to its goal position and the weights decrease those values.

(b) [5] Given two arbitrary admissible heuristics, *h1* and *h2*, which composite heuristic is better to use, max(*h1*, *h2*), (*h1* + *h2*)/2, or min(*h1*, *h2*)? Explain briefly why.

Max(h1, h2) is best because it is admissible but greater than or equal to the other two composite heuristics for all nodes.

Question 4. [10] Adversarial Search

The following tree represents all possible outcomes of a hypothetical zero-sum game:



This tree is from the perspective of the MAX player; MAX nodes are represented by squares and MIN nodes by circles. The leaves of the tree represent the value of the game for the MAX player. The number of each node indicates the order in which they are considered by the Minimax and α-β pruning algorithms.

17. [5 point] What are the "back-up" values of each node in tree using the Mini-max strategy? (The lists are ordered by the node numbers, 1 to 7.)

(a) 3, 3, 1, 3, 4, 2, 4

(b) 4, 4, 4, 10, 2, 2, 9 (c) 10,

10, 4, 10, 9, 2, 9 (d) 1, 1, 1, 3,

2, 2, 4

F SOLUTION: b

18. [5 point] Run the α-β pruning algorithm and list each leaf and node that would NOT be considered by the α-β pruning algorithm. (Assume that leaves are considered in left-to-right order.)

(a) Nodes: 7; Leaves: 3, 7, 9, 8 ,4 (b)

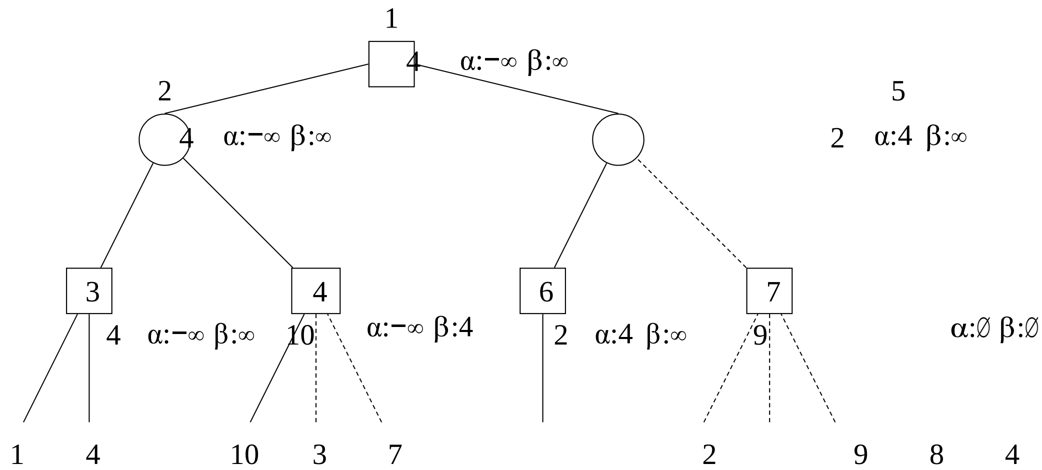
Nodes: 5,6,7; Leaves: 2, 9, 8, 4

(c) Nodes: 4, 7; Leaves: 10, 3, 7, 9, 8, 4 (d)

Nodes: None; Leaves: None

F SOLUTION: a

F SOLUTION: In this picture dashed lines indicate pruned edges.



Question 5. [15] Constraint Satisfaction

Consider the problem of assigning colors to the five squares on board below such that horizontally adjacent and vertically adjacent squares do not have the same color. Assume there are possible two colors, red (*R*) and black (*B*). Formulated as a constraint satisfaction problem, there are five variables (the squares) and two possible values (*R*, *B*) for each variable.

1	2	3
4	5	

a) [5] If initially every variable has both possible values and we then assign variable 1 to have value *R*, what is the result of the *Forward Checking algorithm*?

Forward checking propagates constraints from the current assigned variable to all adjacent unassigned variables, which in this case are variables 2 and 4. This results in the following domains for the variables: 1 = {R}, 2 = {B}, 3 = {R, B}, 4 = {B}, 5 = {R, B}

b) [5] If initially every variable has both possible values and *AC-3* is run, what are the resulting domains for each of the variables?

None of the domains change, with each variable still have both possible values, {R, B}.

c) [5] If initially every variable has both possible values except variable 5 has only value *B*, what is the result of *AC-3*?

1 = {B}, 2 = {R}, 3 = {B}, 4 = {R}, 5 = {B}