National University of Computer and Emerging Sciences, Lahore Campus

Course:	Artificial Intelligence	Course Code:	CS-401
Program:	BS(Computer Science)	Semester:	Fall 2016
Duration:	180 Minutes	Total Marks:	50
Paper Date:	26-11-17	Weight	50
Section:		Page(s):	
Exam:	Final		

Problem 1. [BFS + A* 7] **[7 points]**

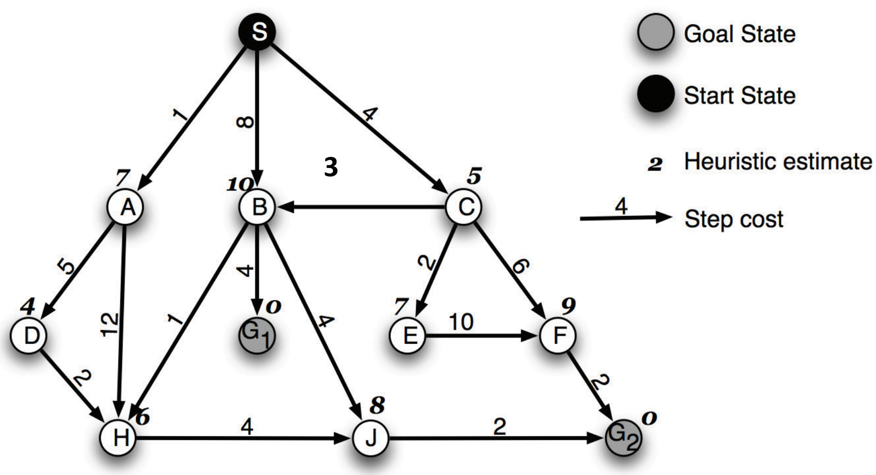
Consider the search space below, where **S** is the start node and $\mathbf{G_1}$ and $\mathbf{G_2}$ are goal nodes.

Arcs are labeled with the value of a cost function; the number gives the cost of traversing the arc.

Along each node is the value of a heuristic function; the number gives the estimate of the distance to the goal.

Assume that uninformed search algorithms always choose the left branch first when there is a choice.

Also assume that the algorithms do not keep track of and recognize repeated states.



For each of the following search strategies:

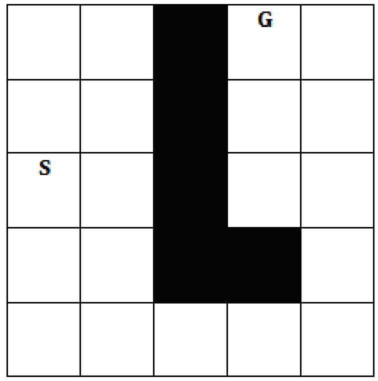
- What path would be found by the algorithm
- List in order, all the states that are popped off the OPEN list. (Sequence of path)
- Nodes in the Fringe when the goal was found

(a) BFS [1 + 1 + 1 Points]

(b)A* [2 + 1 + 1 Points]

Problem 2. [DFS] **[8 Points]**

Given the following maze with S marking the start state cell and G marking the goal cell. An intelligent agent can move in this maze either horizontally or vertically while moving along a diagonal is not allowed. Black Cells are blocked and hence an agent cannot move into these cells.

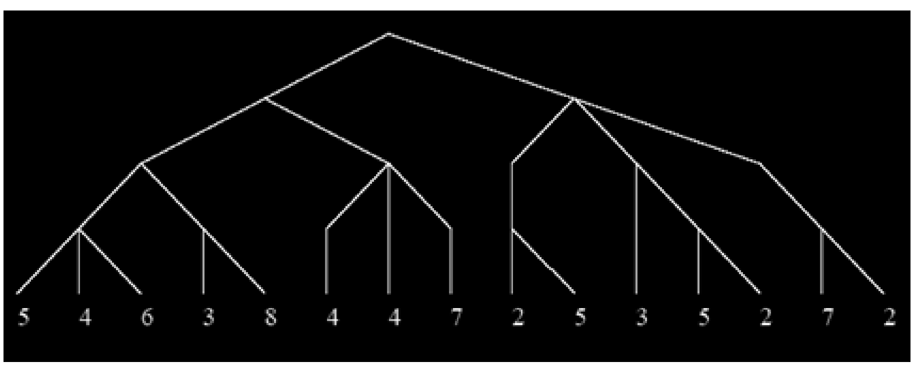


Show the order of cells expanded by DFS algorithm. Mark the start cell as 1 and the next cells as 2, 3, ...

Assume that **DFS** is implemented such that it inserts the cells in the **stack** in **UP-LEFT-DOWN-RIGHT** order and cells already visited are not re-visited (Graph Search Implementation).

Problem 3. [MINIMAX, Pruning] **[3 + 5 + 2 Points]**

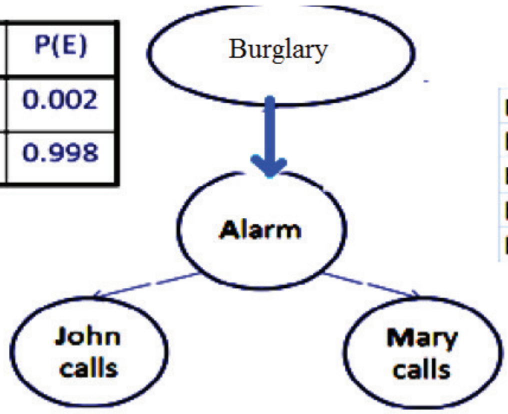
In an adversary game, like the game of CHESS, **MAX** is playing against **MIN.** It is MAXs turn to make a move and he has constructed the following game tree down to 3-levels. **MAX** uses the **MINIMAX** algorithm to decide his move.



a) Simulate the MINIMAX algorithm to figure out the move of MAX by using his game tree.
b) Which parts of the tree will be pruned if MAX uses the MINIMAX algorithm with **α-β** pruning procedure while the nodes are always explored/processed from left to right? To get full marks you must clearly indicate the values of **α** and **β** used by MAX at the of pruning.
c) We know that MINIMAX is optimal against an OPTIMAL opponent but non-optimal against a non-optimal opponent. Justify this statement by arguing that the move chosen by MAX in part a is not optimal against a non-optimal **MIN.**

Problem 4 [Belief Networks] **[12 points]**

E	P(E)
+e	0.002
-e	0.998



B	A	P(A|B)
b+	a+	0.95
b-	a+	0.1
b+	a-	0.05
b-	a-	0.9

A	J	P(J|A)
+a	+j	0.9
+a	-j	0.1
-a	+j	0.05
-a	-j	0.95

A	M	P(M|A)
+a	+m	0.7
+a	-m	0.3
-a	+m	0.01
-a	-m	0.99

Part a) [3 points] Use the belief network given above to compute probabilities of the joint probability distribution of the four variables assuming that each variable can take only two different values.

Part b) [2 + 2 + 2 points] Compute the following

i. P(+b l +j, +m)

ii. P(-b l -m, +j).

iii. Marginal distribution of variables E, A, and M

Part c) [3 Points] By computing some appropriate probabilities show which of the following pair of variables are independent of each other.

i. J and M

ii. M and A

iii. A and M

Problem 5. **[2 + 3 Points]**

We need to learn a concept called **PlayTennis**, that a on a given day the students of FAST-NU-LHR would like to play tennis. The following table represents a snap of the training data gathered by taking measurements of 14 students on 14 different days.

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

It has been decided to train the Naive Bayes classifier from this training data.

i. What is the minimum size of the set of numbers needed to completely give all probabilities needed by the naive Bayes classifier? Justify your answer.

ii. Give the conditional probability tables associated with the nodes corresponding to the attribute ***Wind and outlook.*** Do not use the smoothing while estimating probabilities from the data.

(Write Your answer at the back of this page)

Problem 6 Perceptron Learning **[4 + 4 points]**

You have decided to become a teacher. The only issue is that you don't want to spend lots of time grading essays, so instead you decide to grade them all with a linear classifier. Your classifier considers the number of 7-letter (f7) and 8-letter words (f8) in an essay and then assigns a grade, either A or F, based on those two numbers. You have four graded essays to learn from:

$BIAS$	f_7	f_8	grade
1	2	1	A (+)
1	0	2	F (-)
1	1	2	A (+)
1	1	0	F (-)

You decide to run perceptron learning rule to learn the weights of a perceptron and being optimistic about the students essay writing capabilities, you decide to initialize your weight vector as (1; 0; 0). If the score from your classifier is greater than 0, it gives an A, if it is 0 or lower, it gives an F. Fill in the resulting weight vector after having seen the first training example and after having seen the second training example.

	BIAS	f_7	f_8
Initial	1	0	0
After first training example			
After second training example			

Part b) For each of the following decision rules, indicate whether there is a weight vector (a perceptron) that can represent the decision rule. If your answer is **"Yes"** then show such a weight vector.

1. A paper gets an A if and only if it satisfies (f7 + f8 > 7).

2. A paper gets an A if and only if it satisfies (f7 > 5 AND f8 > 4).

3. A paper gets an A if and only if it satisfies (f7 > 5 OR f8 > 4).

4. A paper gets an A if and only if it has between 4 and 6, inclusive, 7-letter words and between 3 and 5 8-letter words.

Question [BONUS PART]

We need to create an intelligent shopping Robot which will operate in large single-story shopping mall to facilitate the customers.

The shopping mall is divided into a number of regions and each region contains a range of products. Some of the regions are adjacent to each other and the Robot can directly go to any of the neighboring/adjacent region of a given region.

The whole map of the shopping mall can be represented as a graph with the nodes as regions and there is an undirected edge between neighboring regions.

As programmer of the shopping robot we can use a simple command MOVE_To(R_ID) move to region R_ID that is adjacent to the present region and the Command PICK_UP(ITEM_ID) to pick the item using the ITEM_ID.

The user of our Robot will provide it the ITEM_ID of a single item to be purchased and the main job of our shopping Robot will be to identify the target region that contains the item by using a database and then go to the appropriate part of the store and bring the requested item by using minimum number of steps. You can assume that we can query the database by using the interface function R_ID = QUERY(ITEM_ID); It has been decided to formulate the problem of finding the shortest path from the Robots initial position to the destination region and hence your first job is to formulate this problem as a search problem.

Part a) **[3 + 2 Points]**

- Completely specifying a minimal set of items needed to keep the state of the problem.
- Which algorithm(s) could be used to find a shortest path from the starting position to goal region if each **MOVE-TO** command has a constant cost.

Part b) **[3 + 2 Points]**

Now assume that each floor of the mall is divided into n x n square regions. The robot is in the right-bottom corner and the target region is in the upper-right corner where the Robot can only move either horizontally or vertically. Such a floor is shown in the figure below.

			T
			R

- How many regions would be expanded, in the best and worst case, by BFS algorithms. (Assume Graph Search Versions)

How many regions would be expanded by the A* search algorithms if the City-block-Distance is used as a heuristic function. (Assume Graph Search Versions). The City-Block-Distance is defined as the sum of horizontal and vertical distances.