1 A-Star

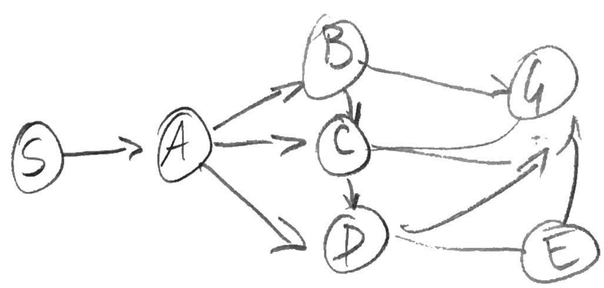
Table 1 gives the edge values for a shortest path problem. Using these and the A* algorithm, find the shortest path from the start node to the goal node. Provide a valid heuristic and show all working. (4 marks)

Directed!

Table 1: Edges to

from

	Start	A	B	C	D	E	Goal
Start	0	6	0	0	0	0	0
A	0	0	7	5	7	0	0
B	0	0	0	2	0	0	6
C	0	0	0	0	4	0	6
D	0	0	0	0	0	6	3
E	0	0	0	0	0	0	3
Goal	0	0	0	0	0	0	0



2 MCMC and Directed Graphical Models

Tables 2 to 6 provide the conditional probability distributions for a directed graphical model.

A. Use this information to draw the graph of the associated directed graphical model. (1 mark)

B. Table 7 provides observed values for some of the nodes. Given these, the initial values provided in Table 8 and the random numbers provided below, use the Metropolis within Gibbs MCMC sampling algorithm to generate two complete samples of the variables. Assume that the candidate function gives the opposite of the current value. At each step, explain what value you are considering, what the current and candidate values are, and why you updated it or did not update it. (4 marks)

Random numbers: 0.568,0.049,0.148,0.815,0.161,0.07

Table 2: P(A)

-	A=F	A=T
	0.55	0.45

Table 3: P(B|A)

A	B=F	B=T
A=F	0.95	0.05
A=T	0.4	0.6

Table 4: P(C|A)

A	C=F	C=T
A=F	0.4	0.6
A=T	0.95	0.05

Table 5: P(D|B,C)

B	C	D=F	D=T
B=F	C=F	0.45	0.55
B=F	C=T	0.8	0.2
B=T	C=F	0.2	0.8
B=T	C=T	0.9	0.1

Table 6: P(E|C)

C	E=F	E=T
C=F	0.25	0.75
C=T	0.3	0.7

Table 7: Observed Values

Node	Value
1	TRUE
3	TRUE

Table 8: Initial Values

Node	Value
2	TRUE
4	FALSE
5	TRUE

3 Hidden Markov Models: Forward-Backward Algorithm

Tables 9 to 12 provide the transition matrix, emission matrix, initial state and a sequence of observations for a hidden Markov model. Use the forward-backward algorithm to calculate the probability distributions for the state of the system at times 0, 1 and 2 given the observations. Show all working. (4 marks)

Table 9: Transition Matrix

S_{t-1}	S_t=0	S_t=1
0	0.2	0.8
1	0.9	0.1

Table 10: Emission Matrix

S	$E = 0$	$E = 1$
0	0.7	0.3
1	0.1	0.9

Table 11: Initial State

$S = 0$	$S = 1$
0.5	0.5

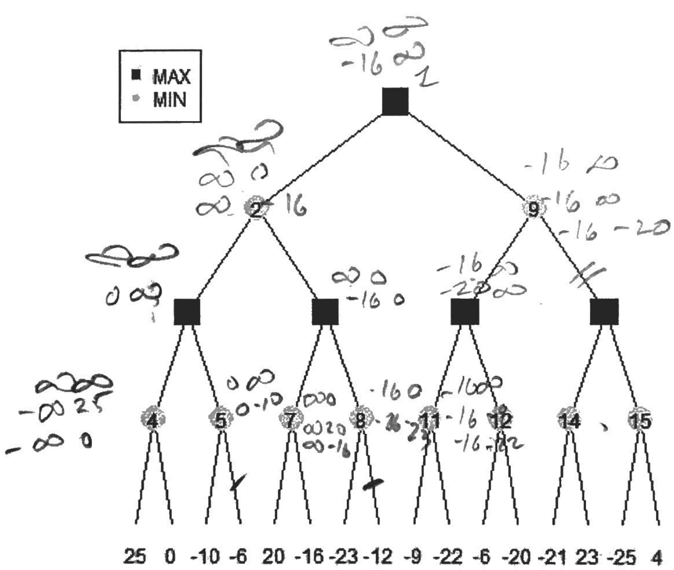
Table 12: Observations

Time=1	Time=2
TRUE	TRUE

4 Alpha-Beta Pruning

Examine the game tree included in this exam. Note that the values in the nodes are node indices, not mini-max values. Perform alpha-beta pruning on this game tree. You should show all working, where this means all alpha-beta values. You can write these values on the diagram, or alternatively on a separate sheet of paper. In both cases, provide a way of identifying the sequence of updates to alpha-beta values associated with nodes (we suggest you just cross out old values and write new values sequentially downwards). If you write on a separate sheet of paper, use the node indices in the diagram as a way of identifying which node particular alpha-beta values are associated with. Show where pruning occurs, by indicating which branches will not be evaluated. Finally, provide the result (value at end state) of the game assuming optimal play. (3 marks)

Game tree for alpha-beta pruning.



5 Scheduling

Provide a complete resource constrained schedule for the actions found in Table 13. (4 marks)

Table 13: Actions

Index	Action	Duration	Uses	Consumes	After
1	Start	0		0 nails	NA
2	Action 1	50		0 nails	1
3	Action 2	45	Saw	-1 nail	1
4	Action 3	35		-1 nail	1
5	Action 4	20	Saw,Hammer	0 nails	3,4,2
~~6~~	~~Action 5~~	~~5~~	~~Hammer~~	1 nail	3,4
7	Action 6	45	Saw,Hammer	0 nails	4,2
8	Action 7	30	Saw,Hammer	1 nail	5,7,4,3
9	Finish	0		0 nails	6,8

6 Depth-First Search

Under what conditions could a depth-first search FAIL to find a solution (in a finite search space with at most a single edge between any two nodes)? (1 mark)