Final Exam Spring 2020 Solutions

Question 1

Q.1.A.1.

a. What is the score of cell B2?

2 (Neighbors are A3 and B3)

b. What is the score of cell B4?

3 (Neighbors are A3, B3 and C4)

c. What is the score of cell C3?

2 (Neighbors are B3 and C4)

d. According to this metric, which is the best cell to place an O?

B4

Q.1.A.2.

a. What is the score of cell B2?

4 (Neighbors are A2, A3, B3 and C2)

b. What is the score of cell B4?

3 (Neighbors are A3, B3 and C4)

c. What is the score of cell C3?

3 (Neighbors are B3, C2 and C4)

d. According to this metric, which is the best cell to place an O?

B2

Q.1.A.3.

a. What is the score of cell B2?

9 (Blocks opponent victory)

b. What is the score of cell B4?

3 (Neighbors are A3, B3 and C4)

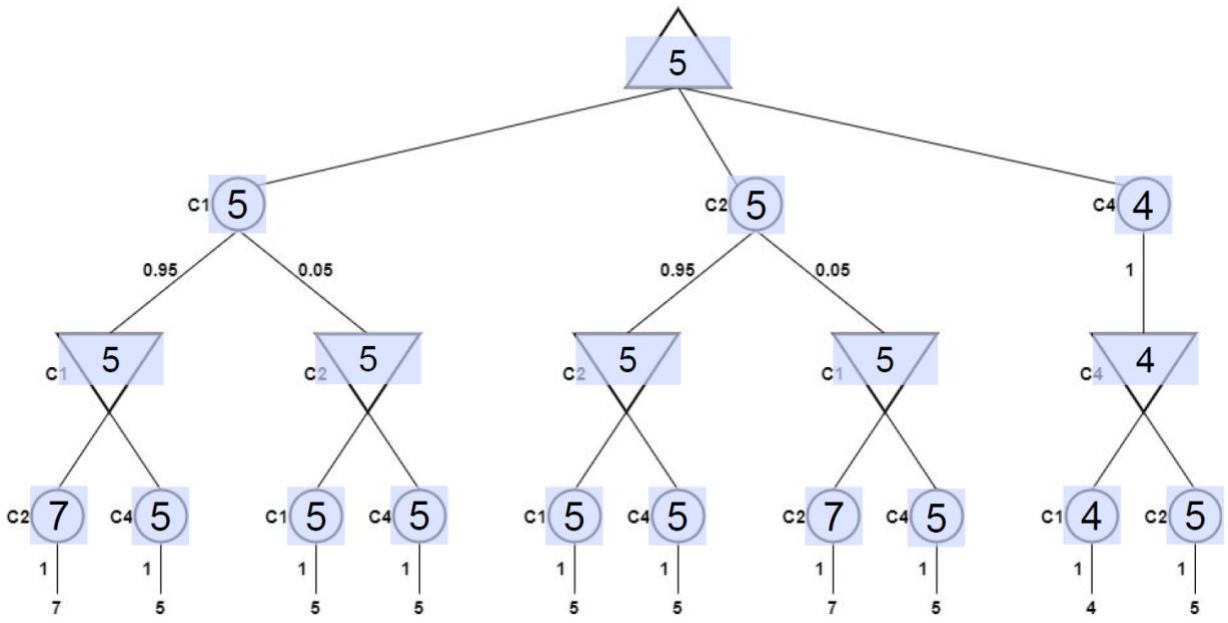
c. What is the score of cell C3?

10 (Causes player victory)

d. According to this metric, which is the best cell to place an O?

C3

Q.1.B.1.



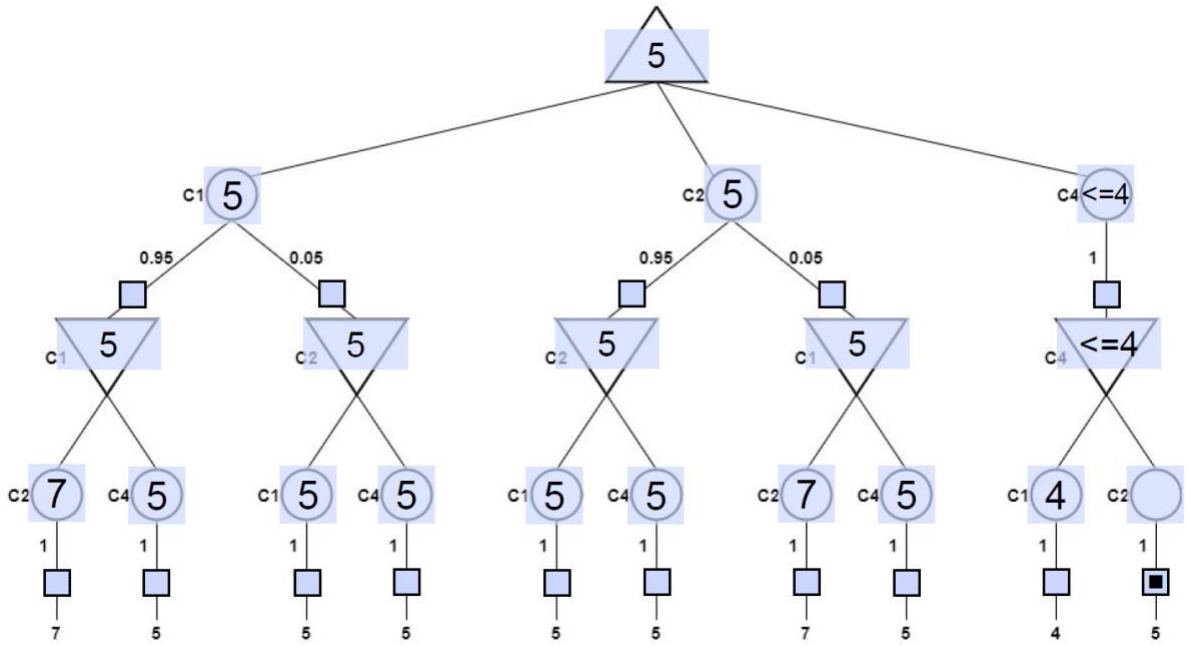
a. What is the value of the root node?

5

b. Which move does O select based on this tree

C1

Q.1.B.2



a. What is the value of the root node?

5

b. Which move does O select based on this tree

C1

c. How many leaf nodes are pruned?

1

Q.1.C.1. Which of these change automatically when you go from regular minimax to minimax with Alphabeta Pruning?

Soln: Option 3 and option 5

Option 1: Incorrect: the selected move doesn't change, the same move is chosen with better efficiency

Option 2: Incorrect: the value of the root node doesn’t change

Option 3: Correct: pruning means some nodes won’t need to be evaluated, so number of computations reduces

Option 4: Incorrect: the evaluation function is something that’s set by the programmer, and so that won’t change as a result of Alphabeta pruning

Option 5: Correct: the number of nodes explored during a search is usually lesser after Alphabeta pruning

Q.1.C.2. Which of the following are true about Iterative Deepening, in the context of game playing?

Soln: Option 1, Option 3 and option 5

Option 1: Correct: during each iteration of Iterative deepening, a DFS is performed

Option 2: Incorrect: the value of the root node changes based on the depth that is currently being evaluated

Option 3: Correct: each iteration is performed by increasing the depth, until time runs out, to get the best possible move within the given time

Option 4: Incorrect: you return the best move from the previous level, that has been fully searched

Option 5: Correct: better moves are predicted if you increase the search depth, and iterative deepening is based on this principle

Question 2

Q2.1 a: **B - Breadth first search**

Q2.1 b: **C - Uniform cost search**

Q2.1 c: **G, A, E, R**

Q2.2 a: **8 nodes**

Q2.2 b: **B - Estimated cost of cheapest path from current node to goal**

Question 3

For the following questions in table format, partial credit was given if some rows were filled in correctly. Unfortunately, no partial credit could be given for 2.D

3.A:

Step Number	Current X Position	Current Y Position	Objective Function
1	4	10	0.48
2	6	8	0.28
3	8	6	0.28
4	8	8	0.15
5	8	10	0.04

3.B:

If you did the steps correctly, you'll see an infinite loop as a result of you hopping from one side of the local minima to the other. This is a result of not having a granular enough step size, and this was fixed in 3.C.

If you left some rows blank because the algorithm terminated due to the infinite loop, you will still receive full points.

Step Number	Current X Position	Current Y Position	Objective Function
1	2	9	0.34
2	1	7	0.15
3	3	5	0.19
4	3	3	0.17
5	3	5	0.19
6	3	3	0.17

3.C

A a different step size was used in this part, and you should see yourself able to reach the local minima that was impossible to hit in 3.B. The algorithm is slower moving across the domain, but it helps explore niches that a larger step size would ignore.

Step Number	Current X Position	Current Y Position	Objective Function
1	2	9	0.34
2	1	8	0.23

3	1	7	0.15
4	2	6	0.09
5	2	5	0.07
6	2	4	0.02

3.D:

Iteration 5 hit the minimum of all the random restarts, returning a shear value of 0.01.

3.E:

There was some confusion about exactly what the context this question was asked in, and what decisions could be taken for granted or not. Additionally, the use of 'potentially' led to some extreme edge-case thought that wasn't originally intended. For these reasons, everyone got full points on this question. The intended answers and their descriptions are below:

- A: False. Gradient Descent is a deterministic process, so starting from the same point with the same step size will always yield the same result. Running multiple iterations won't give you any benefit with these conditions.

- B: True. Laura wants to find the minimum shear value across *every possible value* of X and Y, but searching across an infinite domain would be infeasible. She originally constrains her search to [1, 10] for X and Y to allow the search to actually be completed, and because she estimates that this minimum shear value is in this domain. This estimation is wrong however, and expanding this domain simply allows her to explore more of the shear function, leading to a more informed search.

- C: True. For the same reasons demonstrated in 3.B, we could be skipping over niche containing the global minimum with our step size of 1.

- D: False. Increasing the step size allows you to traverse the domain more quickly which definitely helps when executing a problem feasibly in real-world applications, but can let you skip over areas that contain the local minima. However, some edge scenarios can be imagined where increasing the step size leads to finding the global minimum, so due to our use of the word 'potentially' in the problem this answer could be True or False.

Question 4

Q4.a:

A. When no players are have been assigned to any variable, the domain of M is {M1, M2, M3, M4, M5}.

B. This is basically the constraint that there can only be two players from a single club.

C. This is false since all the players need to be distinct.

D. This follows from the constraint that there can only be two players from a single club.

Hence the answers are B, D.

Partial Points: Half a point for each option being marked (not marked) correctly.

Backtracking Search:

Step 1:

Select G4 for G; budget used $28, remaining $92

Select D2 for D; budget used $57, remaining $63

Select M3 for M; budget used $85, remaining $35

Select F2 for F; budget used $113, remaining $7

Apply inference and rule out F1 and F2 from the domain of F.

No legal value for S. So backtrack.

Step 2:

Already assigned G4, D2, and M3; budget used $85; remaining $35.

Select F5 for F; budget used $105, remaining $15.

Apply inference and rule out F3, F4, and F5 from the domain for F.

No legal value for S. So backtrack.

No legal value for F. So backtrack.

Step 3:

Already assigned G4, D2; budget used $57, remaining $63.

Select M4 for M; budget used $81, remaining $39.

Select F2 for F; budget used $109, remaining $11.

Apply inference and rule out F1 and F2 from the domain of F.

No legal value for F. So backtrack.

Step 4:

Already assigned G4, D2, and M4; budget used $81, remaining $39.

Select F5 for F; budget used $101, remaining $19.

Here you select the value M2 for S.

Q4.b: Answer is 2.

Note: We are giving full points for the answer 15 as well since there was a little confusion between the order in which you apply the inference and add an assignment to the solution.

No partial points

Q4.c: Answer is 2.

No partial points

Q4.d:

Position	Player Selected
G	G4
D	D2
M	M4
F	F5
S	M2

Partial Points: Half point for each position being correct.

Q4.e:

A. The backtracking search finds the first solution and terminates.

B. Forward checking prunes values from the domain and doesn't change the order in which you're assigning values; so the solution remains the same.

C. This statement follows from the definition of minimum values remaining heuristic.

Hence, the answers are B and C.

Partial Points: Half a point for each option being checked (or unchecked) correctly.

Q4.f:

A. This follows from how CSPs are modeled.

B. A fitness function assigns higher fitness values to fitter individuals. This function does the opposite.

C. This function assigns a higher fitness value to fitter individuals.

Hence, the answers are A and C.

Partial Points: Half a point for each option being checked (or unchecked) correctly.

Question 5

Question 5.1.a

Since the question asks you about the potential next word, the language model only calculates the probability of the next word given the bi-gram context - P(word|'are').

P(passing|are) = 3/13

P(terminating|are) = 2/13

P(failing|are) = 3/13

P(not|are) = 2/13

P(running|are) = 1/13

Correct answers = passing or failing

Question 5.1.b

Maximal probability refers to the maximum of the probability values that you calculated in Q.5.1.a

Correct answer = 3/13 or 0.230769

Question 5.1.c

P(please | BOS BOS) x P(terminate | BOS please) x P(the|please terminate) x P(unit|terminate the) x P(tests | the unit) x P(EOS | unit tests) x P(EOS | tests EOS)

since P(the|please terminate) = 0

Correct answer = 0

Question 5.2

Size of the vocabulary |V| = 35 +2 = 37

P(terminating | BOS) = $$\frac{1}{24+37}$$

P(my | terminating) = $$\frac{1}{4+37}$$

P(gradescope|my) = $$\frac{4}{4+37}$$

P(submission|gradescope) = $$\frac{6}{9+37}$$

P(is|submission) = $$\frac{3}{5+37}$$

P(failing|is)= $$\frac{3}{4+37}$$

P(EOS|failing) = $$\frac{6}{5+37}$$

n-gram probability is the product of the above terms.

Correct answer = 3.798982e-9

Question 5.3.a

Correct answer = Perplexity on "unexpected failure in unit tests " is greater

Question 5.3.b.i

n-gram probability of the words = 1.856509e-7 (Calculation is similar to Q.5.2)

N = 7

Correct answer = 9.154080

Question 5.3.b.ii

n-gram probability of the words = 4.672747e-7(Calculation is similar to Q.5.2)

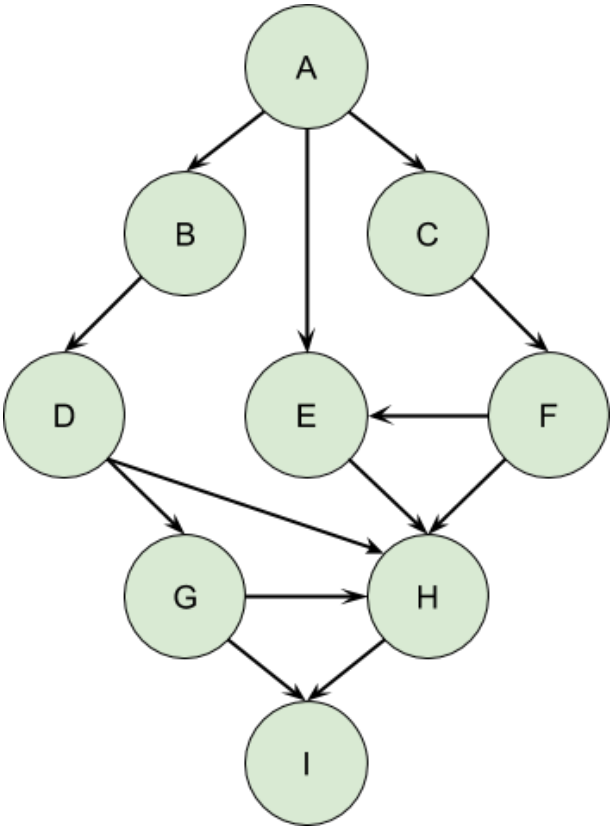
N = 7

Correct answer = 8.023185

Question 6

Part 1

Here is the original graph:



Question 1: Consider the claim: “Nodes A and I are conditionally independent given a set of nodes {X,Y, …, Z}”. For which of the following sets of nodes is the claim true? Choose all that apply.

- G,H
- D,F
- G,E,C
- D,H
- B,E,C

Answer: Only 'G,H' should be ticked

Question 2: Consider the claim: “Nodes A and F are conditionally independent given a set of nodes {X,Y, …, Z}”. For which of the following sets of nodes is the claim true? Choose all that apply.

- C,H
- C,G
- C,E
- C,I
- C

Answer: Only 'C,G' and 'C' should be ticked

Question 3: Consider the claim: “Nodes A and H are conditionally independent given a set of nodes {X,Y, …, Z}”. For which of the following sets of nodes is the claim true? Choose all that apply.

- E
- E,F
- E,F,D
- G,F
- E,F,B

Answer: Only 'E,F,D' and 'E,F,B' should be ticked

Question 4: Given the nodes I and D as evidence, which of the following pairs of nodes are conditionally dependent on one another? Choose all that apply.

- E,F
- G,H
- B,G
- B,E
- C,G

Answer: Every answer choice should be ticked.

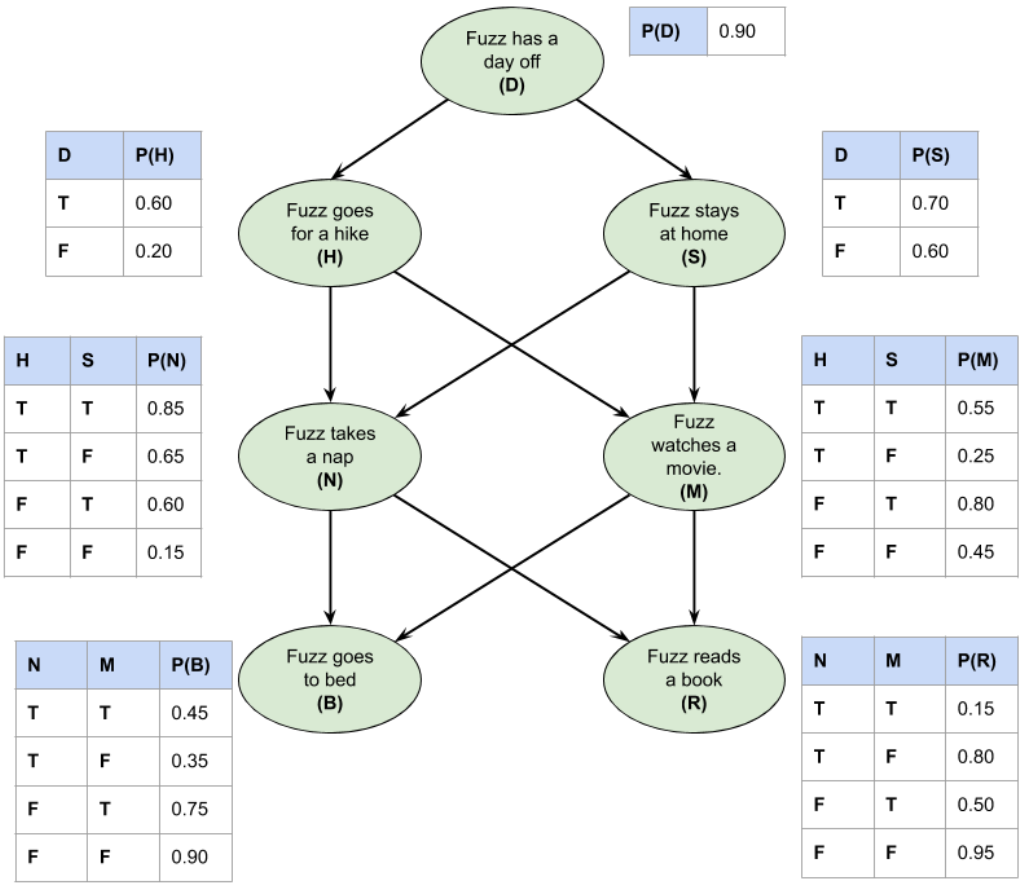
Question 5: Given the nodes B and C as evidence, which of the following pairs of nodes are conditionally independent of each other? Choose all that apply.

- A,D
- A,I
- D,F
- G,F
- D,E

Answer: Every answer choice except 'A,I' should be ticked.

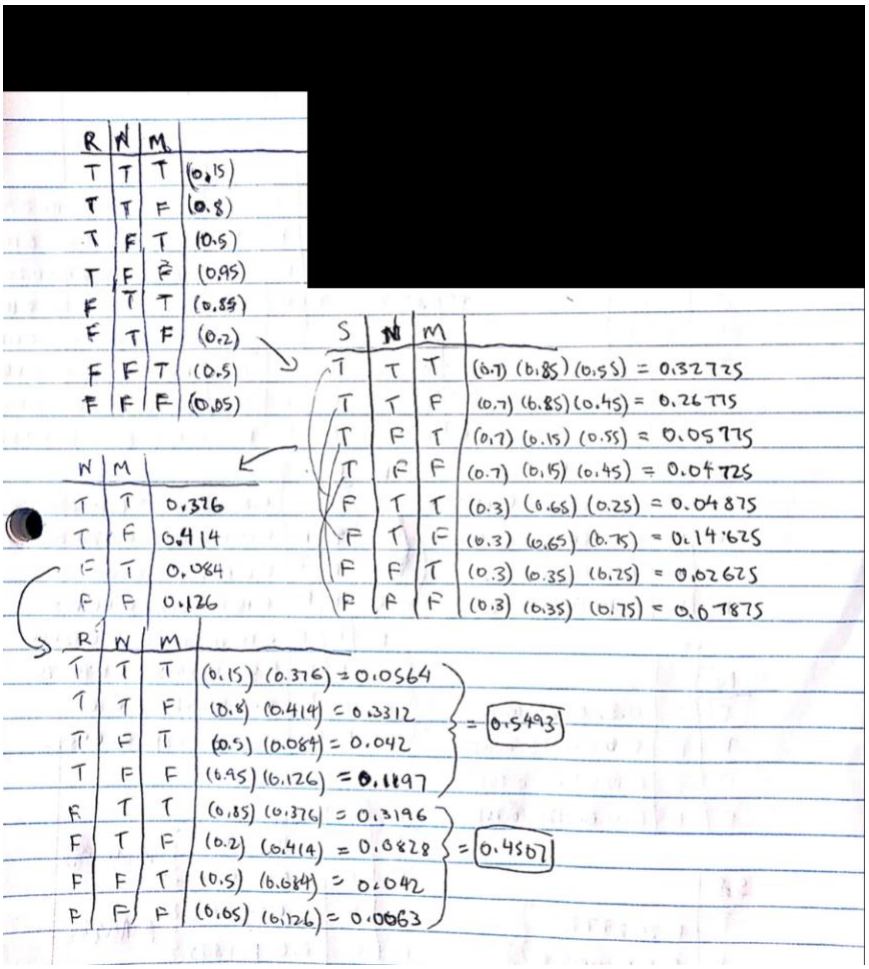
Part 2

Here is the original Bayes Net:



Question 1: What is the probability that Fuzz reads a book (R) given that she goes for a hike (H) and has a day off (D)?

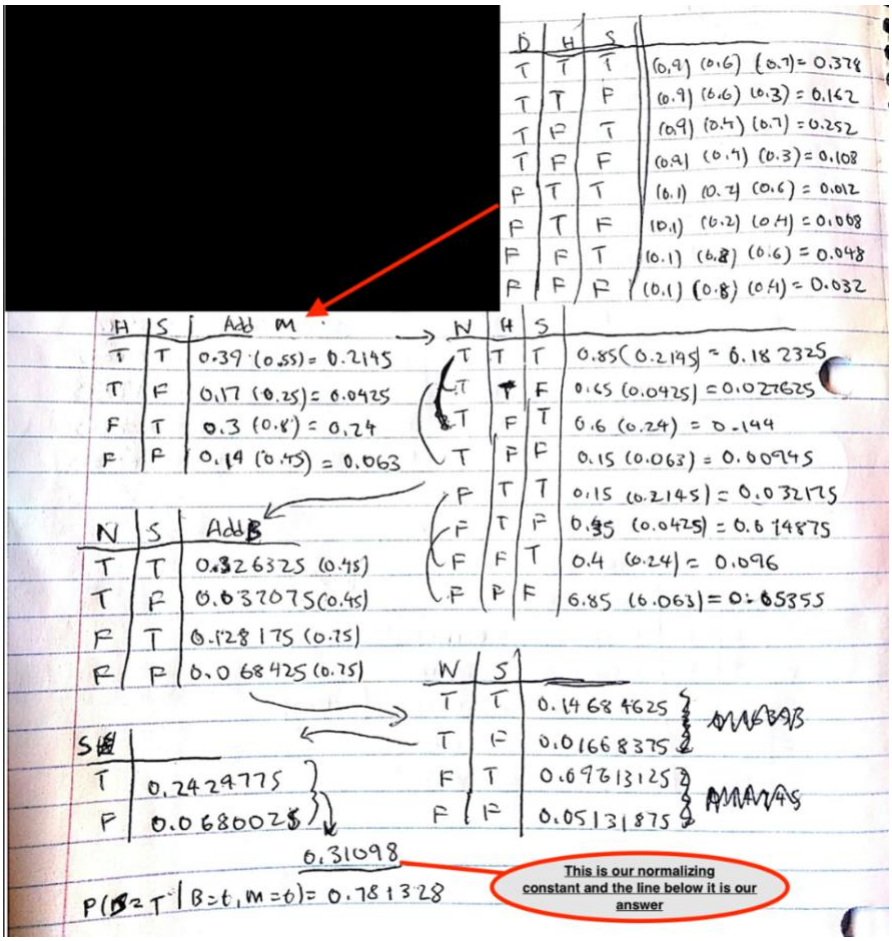
Answer: 0.5493 (work followed by explanation given below).



Here's a step by step overview of what's happening above.

1. We first bring R,N, and M together (table in the top left).
2. We then bring S,N, and M together (table to the bottom right of the table in the previous step).
3. We sum out S (table to the bottom left of the table in the previous step) to get N and M alone in a CPT.
4. Finally, we use the table from the first step to bring R, N and M together.
5. Summing out N and M gets us the desired probabilities.
6. **Note** that we don't normalize by H and D since they are not conditioned on any of the non-evidence variables (so we can pull them out immediately such that they cancel).
7. Also **note** that we don't include B in the summation, since it is neither a predecessor of our variable of interest (R) nor of our evidence (H,D).

Question 2: What is the probability that Fuzz stays at home (S) given that she goes to bed (B) and watches a movie (M)?



Answer: 0.781328 (work followed by explanation given below).

Here's a step by step overview of what's happening above.

1. We first combine D,H, and S into one CPT (table at the top right).
2. Looking to the bottom left of the table of the previous step, we sum out D to get H and S in a table.
    1. We add M in this same table (since it's just a single True value probability).
3. To the right of the table in the previous step, we combine H and S with N to get a CPT.
4. To the bottom left of the table in the previous step, we now sum out H to get N and S in a table.
    1. As in step 2, we add B to this table (since it's just another evidence with constant value).
    2. The table to the bottom right is just the new CPT after adding B.
5. Finally, we sum out N in the CPT below the table in the previous step.
    1. 0.31098 is our normalizing constant and we get 0.781328 as our final probability.

If anything above is unclear, please reach out in the comments. Since it's a scan, please also reach out if anything is difficult to read.

Question 7

Part 1

Q7.1.a. Before we start doing any computations, let's get warmed up a bit first. One of the common activation functions that is used in neural networks is the sigmoid function σ(z). What is the derivative of σ(z)?

Answer: σ'(z) = σ(z) * (1-σ(z))

Q7.1.b. Before doing anything backpropagation-related, let's first see what the forward pass would look like through a neural network. Compute the values of $\hat{\mathbf{Y}}$:

$\hat{y}_1$ = 1.525617
$\hat{y}_2$ = 1.801978

Q7.1.c. With this done, what we need to do is learn with respect to the error. On that note, calculate the following derivatives:

$\frac{\partial E_{out}}{\partial \hat{y}_1}$ = -6.474383
$\frac{\partial E_{out}}{\partial \hat{y}_2}$ = -1.198022

$\frac{\partial E_{out}}{\partial \hat{y}_1}$ = -6.474383
$\frac{\partial E_{out}}{\partial \hat{y}_2}$ = -1.198022

Q7.1.d. The key to backprop is that we learn based on the error that we get. So what we want to do is learn how the parameters of the network that we tune contribute to the error of a network. Suppose we wanted to know how an arbitrary weight, w_x, has an impact on our overall error method. Given the equation, compute the values of the above derivative for six of the network weights:

$\frac{\partial E_{out}}{\partial w_{11}^{(1)}}$ = -4.668776	$\frac{\partial E_{out}}{\partial w_{12}^{(1)}}$ = -0.863912
$\frac{\partial E_{out}}{\partial w_{21}^{(1)}}$ = -6.465675	$\frac{\partial E_{out}}{\partial w_{22}^{(1)}}$ = -1.196411
$\frac{\partial E_{out}}{\partial w_{31}^{(1)}}$ = -6.465761	$\frac{\partial E_{out}}{\partial w_{32}^{(1)}}$ = -1.196427

Q7.1.e. Now we have what we need to re-compute the new weights for backpropagation. Let the learning rate be 0.3. Think about how we want to update these weights. If we were to graph the error as a function of the weights, how would we want to move along that error surface to come up with better weights? Think about this to come up with the new weight matrix after one update?

$w_{11}^{(1)}$ = 2.200633	$w_{12}^{(1)}$ = 0.639174
$w_{21}^{(1)}$ = 2.659703	$w_{22}^{(1)}$ = 1.018923
$w_{31}^{(1)}$ = 2.169728	$w_{32}^{(1)}$ = 1.228928

Q7.1.f. At this point, you should have a good idea how backprop works, so why don't you go ahead and try applying it on to the next set of weights (i.e. between the hidden layer and input layer)? Fill in all your answers below:

$\frac{\partial E_{out}}{\partial w_{11}^{(0)}}$ = -7.932363	$\frac{\partial E_{out}}{\partial w_{12}^{(0)}}$ = -0.051265	$\frac{\partial E_{out}}{\partial w_{13}^{(0)}}$ = -0.023565
$\frac{\partial E_{out}}{\partial w_{21}^{(0)}}$ = -2.266389	$\frac{\partial E_{out}}{\partial w_{22}^{(0)}}$ = -0.014647	$\frac{\partial E_{out}}{\partial w_{23}^{(0)}}$ = -0.006733

$w_{11}^{(0)}$ = 2.489709	$w_{12}^{(0)}$ = 0.745379	$w_{13}^{(0)}$ = 0.72707
$w_{21}^{(0)}$ = 0.769917	$w_{22}^{(0)}$ = 0.754394	$w_{23}^{(0)}$ = 0.79202

Grading Notes:

- If your answers were off on the last decimal point (e.g. 0.3278 instead of 0.3277 or 0.8299 instead of 0.83, you still got full credit).
- If your answers were in exponential form (ex. -2.3565e-02) then you still got full credit.
- Incorrect signs resulted in half credit.

Part 2

In the following questions, the first two questions both got labeled as "Q7.2.a", but we adjusted the numbering here for clarity. All of the correct answers are bolded:

Q7.2.a. With the above information, which one of the options below best addresses why the K-medoids algorithm may be preferred to the K-means algorithm? Consider only the given k-medoids algorithm:

- K-medoids finds the optimal solution
- **K-medoids is more robust to outliers**
- K-medoids initialization method will get it to converge faster than K-means
- None. That is, K-medoids simply uses a different distance metric for computing similarity, but that does not help in addressing any weakness of K-means.

Q7.2.b.i. In what way can we project the data so that an SVM can classify the data perfectly? That is, suppose we project the data to the kth dimension to get the SVM to classify the data linearly. What could the data look like in this k_{th} dimension (ignore the number of points in the answers, we just want the shape of the data)?

- **O's and +'s on opposite sides? (example below)**
- Alternating concentric circles, where we have all O's in the innermost circle, then +'s in the circle outside that, then O's outside that, etc.
- No projection of the data is needed. The data can be separated as is.
- O's and X's alternate but line up with each other? (example below)

Q7.2.b.ii. Which of the following properties are true regarding SVMs (select all that apply):

- **These models create a linear-separating hyperplane to classify data**
- SVMs are parametric
- These models minimize the distance between support vectors and the separator
- **SVMs attempt to minimize expected generalization loss as opposed to empirical loss**

Question 8

Logic

Q8.1.a) $F: (A \Rightarrow D) \vee (B \Rightarrow D) \vee (C \Rightarrow D) \wedge E$

$G: (A \wedge B \wedge C) \Rightarrow D$

$F: \underbrace{(A \Rightarrow D) \vee (B \Rightarrow D) \vee (C \Rightarrow D)}_{a \Rightarrow b \,=\, \neg a \vee b} \wedge E$

$(\neg A \vee D) \vee \underbrace{(\neg B \vee D) \vee (\neg C \vee D)}_{(a \vee (b \wedge c)) \,=\, ((a \vee b) \wedge (a \vee c))} \wedge E$

$(\neg A \vee D) \vee ((\neg B \vee D) \vee (\neg C \vee D)) \wedge ((\neg B \vee D) \vee E))$

$\underbrace{(\neg A \vee D) \vee ((\neg B \vee \neg C \vee D) \wedge (\neg B \vee D \vee E))}_{(a \vee (b \wedge c)) \,=\, ((a \vee b) \wedge (a \vee c))}$

$((\neg A \vee D) \vee (\neg B \vee \neg C \vee D)) \wedge ((\neg A \vee D) \vee (\neg B \vee D \vee E))$

$(\neg A \vee \neg B \vee \neg C \vee D) \wedge (\neg A \vee \neg B \vee D \vee E)$ ← CNF

Q8.1.b) $G: (A \wedge B \wedge C) \Rightarrow D$

$\neg(A \wedge (B \wedge C)) \vee D$

$\neg A \vee \neg (B \wedge C) \vee D$

$\neg A \vee \neg B \vee \neg C \vee D$ ← CNF

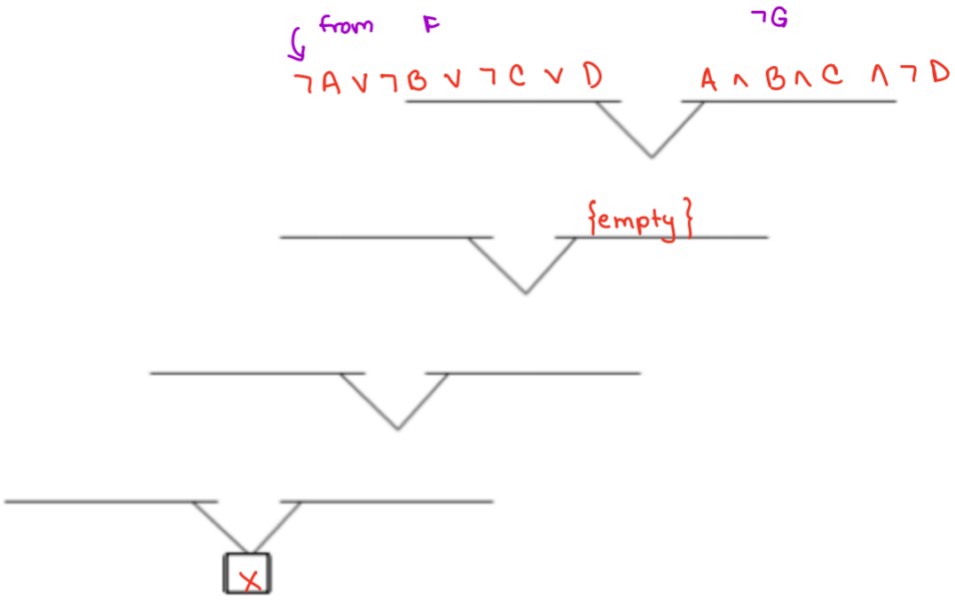
Q8.1.c) You can check this by creating a truth table consisting of each literal, and plugging all possible 'True' or 'False' values for each literal into the implication F ⇒ G, you will find that all possible literal value combinations result in F ⇒ G = 'True'.

For any sentences F and G, F |= G (entails) if and only if the sentence (F ⇒ G) is valid.

To show that F |= G, we show that (F ∧ ~G) is unsatisfiable.
We do this by proving a contradiction. An empty clause is equivalent to False.

$\neg G: \neg(\neg A \vee \neg B \vee \neg C \vee D)$

$A \wedge B \wedge C \wedge \neg D$ ← CNF



Valid

'Until two clauses resolve to yield the empty clause' convergence condition (from textbook) of resolution algorithm is reached.

Q8.1.d)

Q8.1.d. Which of the following sets of clauses are satisfiable (here ¬ is "Not" operator) **(2 points)**:

- [x] {(a, b),(c,¬e),(¬b, c),(¬d, ¬a),(¬a, ¬b),(¬c, ¬d),(d, e),(¬c, b)}
- [ ] {(x), (¬x, y), (¬x, ¬z), (z, ¬x, ¬y)}
- [ ] {(a, b), (¬a, c), (¬a, ¬c), (a, ¬b)}
- [x] {(y), (¬x, y), (y, ¬z), (x, z)}
- [x] {(c, b, ¬d), (¬c, e, f), (d, ¬e), (d, ¬f), (a, ¬b, c), (¬a, f)}

Planning

Q8.2.a)

Patients: P1 P2 P3

Gloves: G1 G2

sides: G1a G1b G2a G2b

Operations:

Invert (Gxx)

Wear (Gxx)

Treat (Px, Gxx)

Remove (Gxx)

* Note there is more than 1 valid answer.

Sample answer:

Q8.2.a. What is an action sequence that moves us from the initial state to the goal state? **(3 points)**:

Wear(G1a); Wear(G2a); Treat(P1, G2a); Remove(G2a); Treat(P2, G1a); Invert(G2a); Wear(G2b); Treat(P3, G2b);

Question 9

Part A

TASK 1. Compute the total distance and the shortest warping path between the unknown recording and the word "George".

Q9.A.1.a. What is the total distance between both time series? **(1 points)**

15

Q9.A.1.b. What is the shortest warping path between both time series? **(2 points)**

Answer Format: 0, 3, 4, 4, 8, 8, 9, 11, 13, 14, 15, 15

0, 2, 2, 2, 2, 2, 5, 8,11, 13, 15

1	48	44	48	51	54	23	23	15
1	42	40	34	37	40	17	17	13
10	36	38	20	23	26	11	11	15
10	33	35	15	18	18	8	8	15
10	30	32	10	10	10	5	8	15
18	27	29	5	2	2	13	24	39
15	16	18	2	5	8	16	24	36
15	8	10	2	5	8	16	24	36
7	0	2	10	21	32	32	32	36
	7	5	15	18	18	7	7	3

TASK 2. Compute the total distance and the shortest warping path between the unknown recording and the word "Georgia".

Q9.A.2.a. What is the total distance between both time series? **(1 points)**

9

Q9.A.2.b. What is the shortest warping path between both time series? **(2 points)**

Answer Format: 0, 3, 4, 4, 8, 8, 9, 11, 13, 14, 15, 15

0, 2, 2, 2, 2, 2, 5, 8, 9, 9, 9

1	48	44	48	51	54	23	23	25	9
1	42	40	34	37	40	17	17	17	9
10	36	38	20	23	26	11	11	9	18
10	33	35	15	18	18	8	8	9	18
10	30	32	10	10	10	5	8	9	18
18	27	29	5	2	2	13	24	33	47
15	16	18	2	5	8	16	24	30	44
15	8	10	2	5	8	16	24	30	44
7	0	2	10	21	32	32	32	34	40
	7	5	15	18	18	7	7	9	1

Notice that the time series for "George" and "Georgia" have both different lengths and yet with DTW we can perform pattern matching with time series of different lengths.

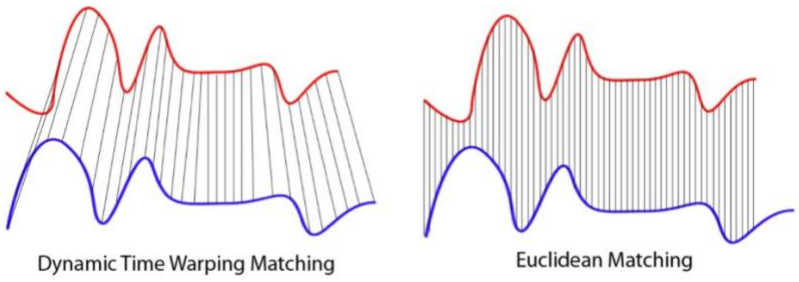
Question B

Q9.B.1. What is the total distance using Sakoe-Chiba bands of degree $N = 0$? **(1.5 points)**

32

1									32
1								32	
10							24		
10						21			
10					18				
18				10					
15			10						
15		10							
7	0								
	7	5	15	18	18	7	7	9	1

Note that Dynamic Time Warping with Sakoe Chiba bands of width $N=0$ is equivalent to performing Euclidean Matching between the two time series.



Q9.B.2. What is the total distance using Sakoe-Chiba bands of degree $N = 1$? **(1.5 points)**

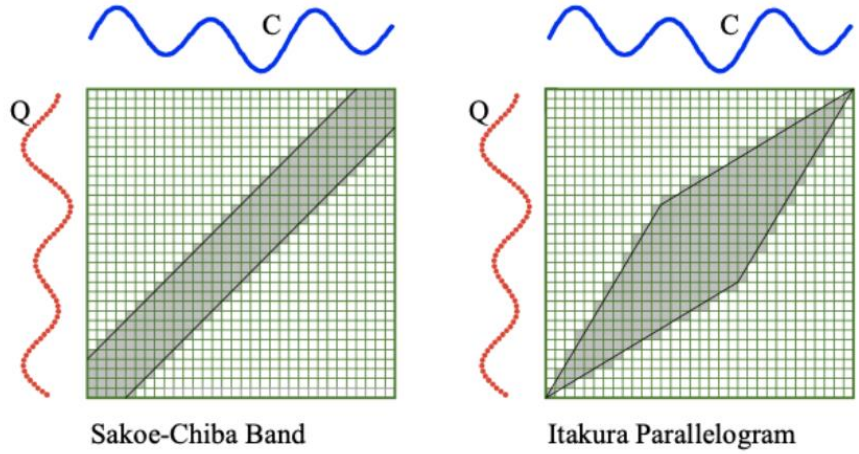
9

1								25	9
1							17	17	9
10						11	11	9	
10					18	8	8		
10				10	10	5			
18			5	2	2				
15		18	2	5					
15	8	10	2						
7	0	2							
	7	5	15	18	18	7	7	9	1

Notice that the total distance 9 and warping path $0,2,2,2,2,2,5,8,9,9,9$ is exactly the same one obtained without any bands in **Q9.A.2**. This is helpful to show that even with a reduced band size we can get a good estimate of the distance between two time series. Hence with some warping allowed, we can reduce the runtime complexity of the DTW and yet obtain a reasonable calculation of the distance between two sequences.

Q9.C.1. Which of the two bands is more apt for the speech recognition task? **(1.5 points)**

- ○ Sakoe-Chiba Band
- ◉ Itakura Parallelogram



Sakoe-Chiba Band Itakura Parallelogram

We learned in **Question B** that limiting the size of the bands limits local warping and can help reduce "unreasonable" matches. **Question C** introduces different kinds of bands and allows the student to think about what kind of bands might be appropriate to use for a particular task. The shape of the *Itakura Parallelogram* allows more warping in the middle and less at the edges, whereas the shape of the *Sakoe-Chiba band* is fixed from beginning to end. Therefore, *Itakura Parallelogram* would perform better in the classification accuracy of the speech recognition task (in general) because speech tends to have the most variability in its middle and very little at the beginning or the end. For example, the word *Florida* can be pronounced in three ways: *Flow-ri-da*, *Flah-ri-da*, and *Flaw-ri-da*.

Question 10

Question 1: (1.5 points)

$$U_{i+1}(s) \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a) U_i(s')$$

S1: V(S1) = max{

0 + 0.8 * 10 + 0.2 * 15

0 + 0.6 * 10 + 0.4 * 15

0 + 0.1 * 10 + 0.9 * 15

} = max{11, 12, 14.5} = 14.5

Question 2: (4.5 points = 1 + 1 + 1 + 0.75 + 0.75)

$$U_{i+1}(s) \leftarrow R(s) + \gamma \sum_{s'} P(s' \mid s, \pi_i(s)) U_i(s')$$

State	Initial Policy	Utility- Iteration 0	Utility- Iteration 1	Utility- Iteration 2
S1	B2	0	0 + 0.6 S2 + 0.4 S3 = 12	17.52
S2	B3	10	10 + 0.6 S4 + 0 = 17.2	20.8
S3	B1	15	15 + 0 + 0.3 S6 = 18	20.52
S4	B3	12	12 + 0.4 S5 = 18	18
S5	B2	10	10 + 0.7 S7 = 18.4	18.4

Question 3 (2.5 points = 1 + 0.5 + 0.5 + 0.25 + 0.25)

$$\pi[s] \leftarrow \underset{a \in A(s)}{\operatorname{argmax}} \sum_{s'} P(s' \mid s, a)\ U[s']$$

State	Policy
S1	argmax(0 + 0.8 S2 + 0.2 S3, 0.6 S2 + 0.4 S3, 0.1 S2 + 0.9 S3) = argmax(20.744, 20.688, 20.548) = B1
S2	argmax(10 + 0.9 S4 + 0, 10 + 0.6 S4) = argmax(26.2, 20.8) = B2
S3	argmax(15 + 0.3 S6, 15 + 0.1 S6) argmax(20.52, 16.84) = B1
S4	B3
S5	B2

Question 4 (2 points = 1 + 0 + 1 + 0 + 0)

$$U_{i+1}(s) \leftarrow R(s) + \gamma \sum_{s'} P(s' \mid s, \pi_i(s)) U_i(s')$$

State	Initial Policy	Utility- Iteration 0	Utility- Iteration 1	Utility- Iteration 2
S1	B1	0	0 + 0.8 S2 + 0.2 S3 = 11	20.24
S2	B2	10	10 + 0.9 S4 + 0 = 20.8	26.2
S3	B1	15	15 + 0 + 0.3 S6 = 18	20.52
S4	B3	12	12 + 0.4 S5 = 18	18
S5	B2	10	10 + 0.7 S7 = 18.4	18.4

Question 5 (1.5 points = 1 + 0 + 0.5 + 0 + 0)

$$\pi[s] \leftarrow \underset{a \in A(s)}{\operatorname{argmax}} \sum_{s'} P(s' \mid s, a)\ U[s']$$

State	Policy
S1	argmax(0 + 0.8 S2 + 0.2 S3, 0.6 S2 + 0.4 S3, 0.1 S2 + 0.9 S3) = argmax(25.064, 23.928, 21.088) = B1
S2	argmax(10 + 0.9 S4 + 0, 10 + 0.6 S4) = argmax(26.2, 20.8) = B2
S3	argmax(15 + 0.3 S6, 15 + 0.1 S6) argmax(20.52, 16.84) = B1
S4	B3
S5	B2