CS6601 Final Exam – Spring 2017

Instructions – please read thoroughly!

Do not put your name on this exam. The exams will be graded anonymously.

Fill out this PDF form and upload it to **Gradescope** **only** when you are done, with unlimited resubmissions until the deadline. You can: **(a)** type directly into the form – we suggest using Adobe Reader DC (or Evince on Linux). Other programs may not save your answers so **please keep a backup**; or **(b)** print, hand-write, & scan. You can combine the methods as well. Bullet point answers are appreciated over full sentences**. Show your work wherever indicated and clearly indicate your final answer.**

Submit only a single PDF – no phone pictures, please! (You may use an app like CamScanner if you do not have scanner access.) Do not add pages unless absolutely necessary; if you do, please add them at the end of the exam only, and clearly label both the extra page and the original question page. Submit **ALL** pages of the exam, not only the completed ones.

Do not forget to fill the checklist at the end before turning in the exam. The exam may not be graded if it is left blank.

The exam is open-book, open-note, open video lectures, with no time limit aside from the open period. No internet use is allowed with exceptions for e-text versions of the textbook & this semester's CS6601 course materials. No resources outside this semester's CS6601 class should be used. Do not discuss the exam on Piazza, Slack, or any other form of communication. If there is a question for the teaching staff, please make it private on Piazza and tag it as Final Exam with the question number in the subject line.

Point breakdown:

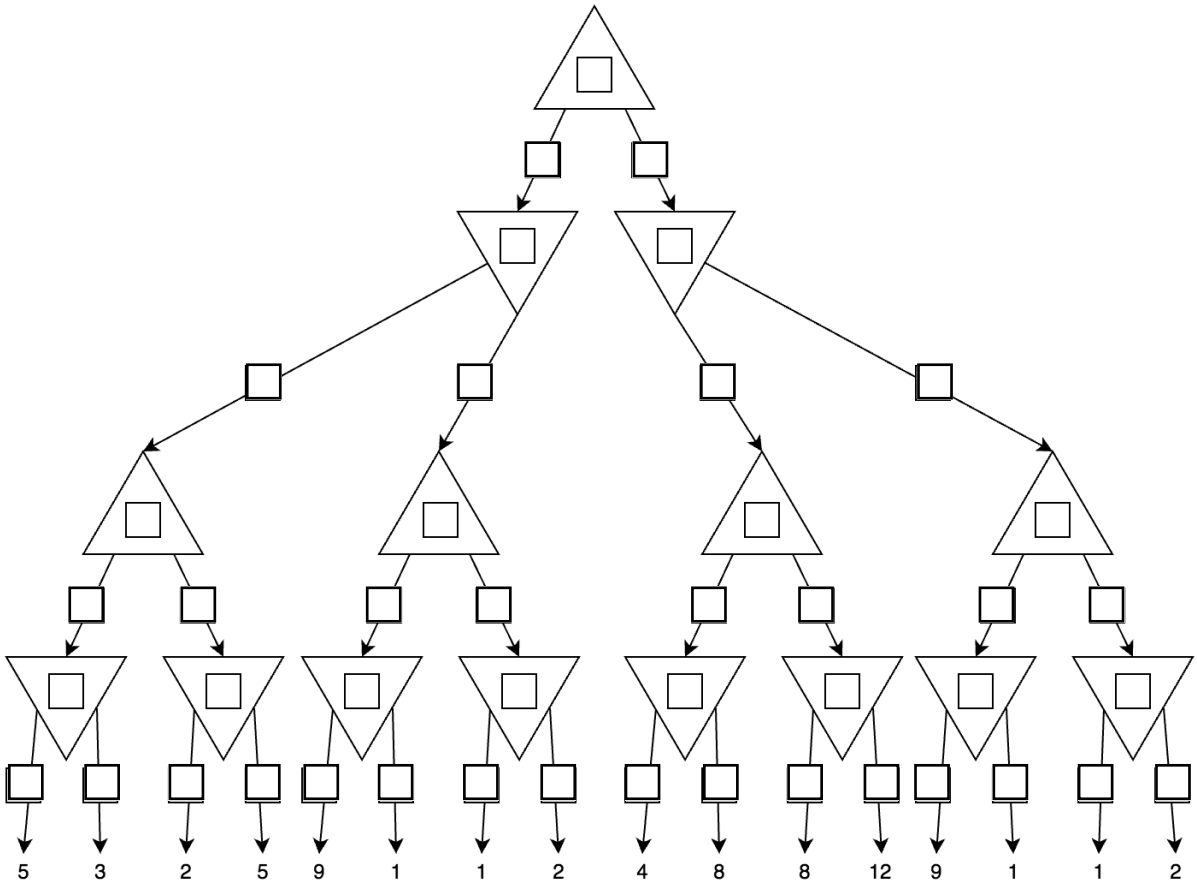
Each question has sub-parts with varying points. Bonus questions (optional) are 1 point each and applied to the overall grade (not included in the exam points).

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Total
Pts	**7**	**4**	**12**	**4**	**8**	**10**	**11**	**3**	**8.5**	**15**	**4**	**13.5**	**100**

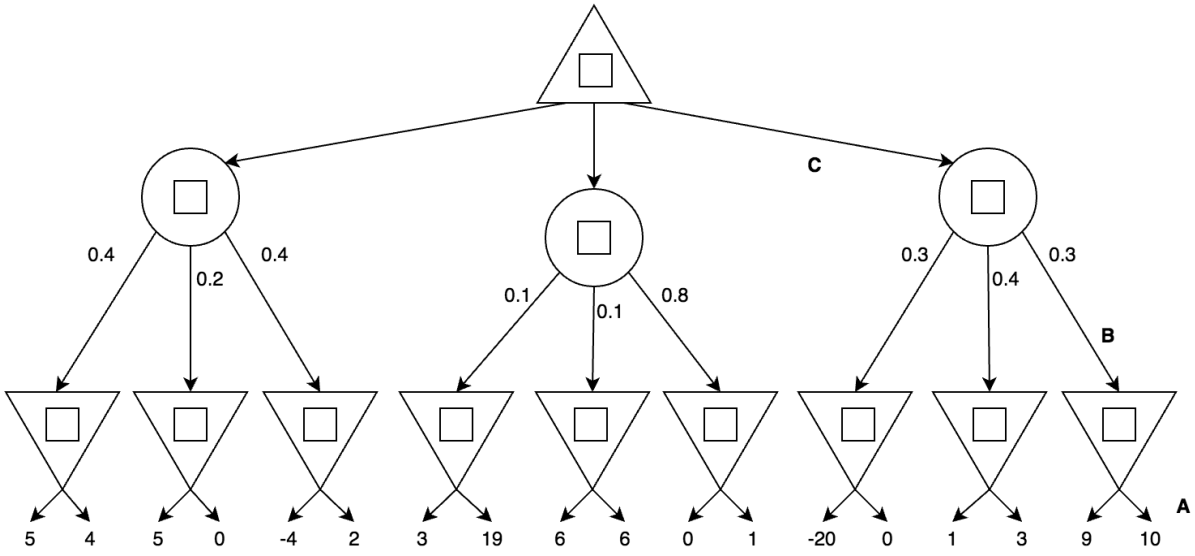
1. Game Playing

(7 points)

Question 1 (2 points):
Solve the game tree below using alpha beta pruning. Evaluate from left to right. Check the box over branches that should be pruned, then fill in the numerical values for remaining branches.



Question 2 (2 points):
Solve the probabilistic game tree below. Note the branches labeled A, B and C - this is what we in the CS6601 business know as *foreshadowing.*



Question 3 (3 points):
For each of the labeled branches (A, B, C) in question 2, identify eval bounds that are consistent with all observed values (5, 4, 5, 0, …) and will lead to the branch being pruned, when evaluated from left to right. Provide the tightest possible integer bound, choose “any” if the branch will be pruned regardless of eval bounds, or choose "not possible" if no valid bound will prune this branch.

Branch A
Lower bound (choose 1):
- ❏ Any
- ❏ Not possible
- ❏ _____ (lowest integer)

Upper bound (choose 1):
- ❏ Any
- ❏ Not possible
- ❏ _____ (highest integer)

Branch B
Lower bound (choose 1):
- ❏ Any
- ❏ Not possible
- ❏ _____ (lowest integer)

Upper bound (choose 1):
- ❏ Any
- ❏ Not possible
- ❏ _____ (highest integer)

Branch C
Lower bound (choose 1):
- ❏ Any
- ❏ Not possible
- ❏ _____ (lowest integer)

Upper bound (choose 1):
- ❏ Any
- ❏ Not possible
- ❏ _____ (highest integer)

2. Search

(4 points)

Sudoku is a logic-based game that consists of 9 3x3 grids that create one large 9x9 grid. The rules of the game are simple: each row has all the numbers from 1-9, each column has all the numbers from 1-9, each individual 3x3 box has all the numbers from 1-9. You can read more about it here: https://en.wikipedia.org/wiki/Sudoku

This is the board you will use to answer the questions below:

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

The start state is the partially filled out board while the end state is all the tiles in the board filled according to the rules provided above. The objective is to find the unknown solution and there exists only one correct solution.

There are three algorithms we will focus on for this part of the question, these algorithms are listed in the textbook:

I . BFS
II. DFS
V. IDA* with depth increasing by 1 each time

Question 1 (1 point):
Rank the algorithms in terms of fewest nodes expected to be expanded to most number of nodes expected to be expanded.

_____

_____

_____

Question 2 (1 point):
Does IDA* use less memory than DFS? Yes/No _____

Question 3 (1 point):
Check the heuristics that are admissible:

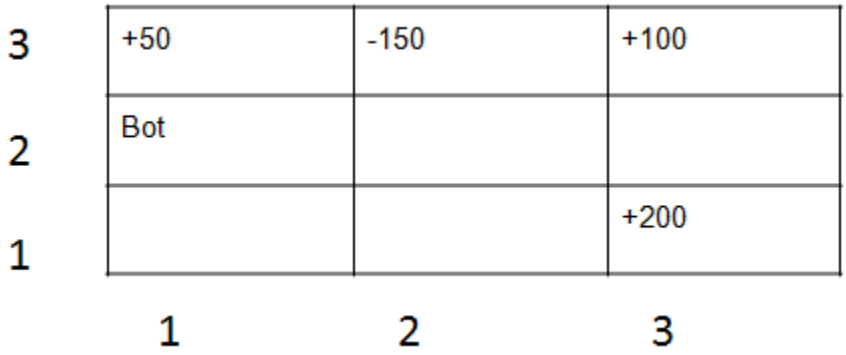
- ❑ h(n) = 0
- ❑ h(n) = 1
- ❑ h(n) = c(n) where c(n) is the average of the current 3x3 grid
- ❑ h(n) = c(n) where c(n) is the average of the current row
- ❑ h(n) = c(n) where c(n) is the average of the current column
- ❑ h(n) = c(n) where c(n) is the number of empty cells left
- ❑ h(n) = c(n) where c(n) is the number of misplaced cells based on the final solution (ie, if a cell should be a 5 but is labeled as a 3)

Question 4 (1 point):
Assume that we have an empty 9x9 grid, which algorithm from below would you use to solve the grid in the least number of expected to expand nodes possible? Pick one algorithm and give a one sentence explanation. If you choose an algorithm with a heuristic, please explain the heuristic you are using as well.
Algorithms: BFS, DFS, UCS, A*, IDA*

3. MDP

(12 points)

Question 1 - Value Iteration (3 points):
Consider the grid-world given below and a bot who is trying to learn the optimal policy. If an action results in landing in one of the reward states, the corresponding reward is awarded during that transition. All states with rewards are terminal states. The other states have the North, East, South, West actions available. Assume this is a deterministic environment. The discount factor γ = 0.5. Here the bot is at (2,1)



What is the optimal utility at each of the following states?

1. (2,2) _____
2. (2,3) _____
3. (1,1) _____

Question 2 - Policy Iteration:
Consider the MDP shown. For simplicity, all transitions are deterministic.

S1 R=1	S2	S3	S4	S5	S6 R=10

- There are two actions: move left and move right.
- States are labelled with the immediate reward that results from landing in that state. (unlabeled states have zero reward).
- S1 and S6 are the termination states. Discount factor γ= 0.8.

We wish to run Policy Iteration to find the optimal policy for this MDP. Initially, our policy is to always take left.

Q1. Compute the policy evaluation step for the first iteration using the initial policy. Show the calculations. **(4 points)**

	S2	S3	S4	S5
U0	0	0	0	0
U1	_____	_____	_____	_____

Q2. Next step is policy improvement. **(2 points)**

a. What is policy for s2 after the first iteration of improvement.

_____

b. Show calculations to justify the above answer.

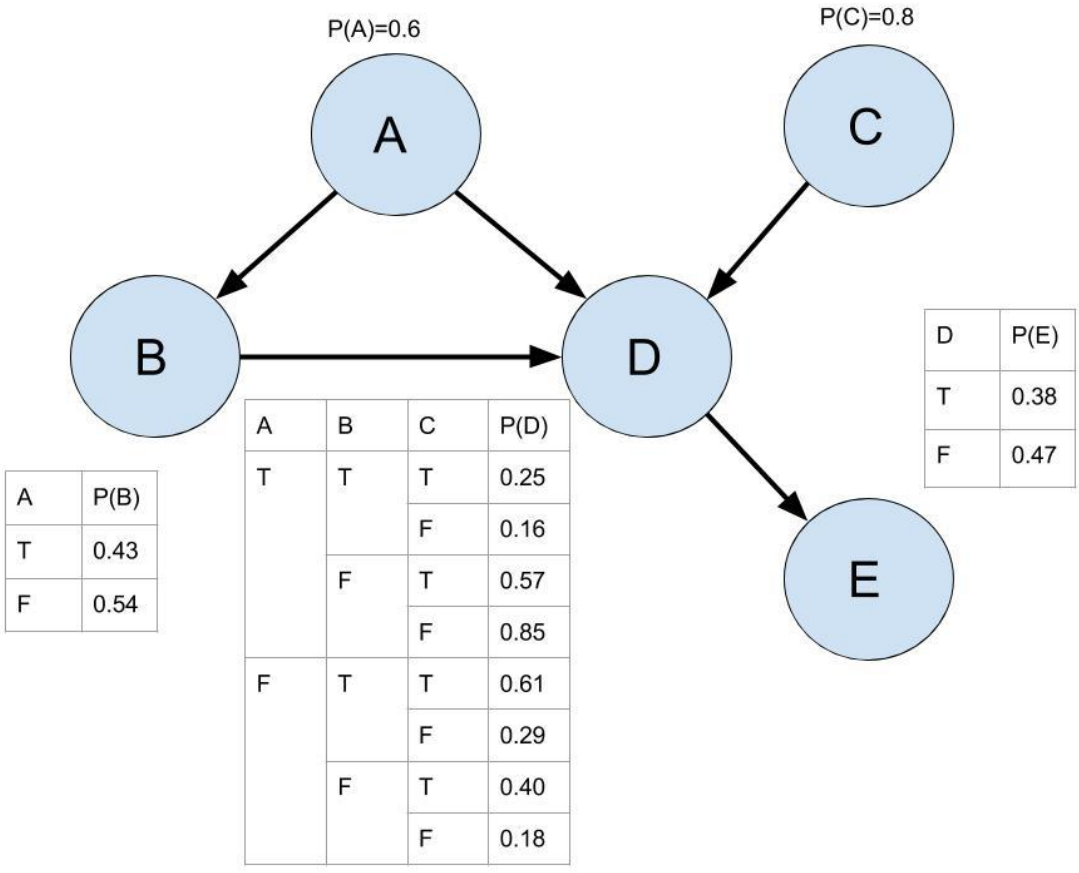
Question 3 - POMDP:
In a combat game, we have 2 fighters who we need to fight one at a time. Here you have a choice of playing against any one first. Fighter one (F1) is very weak and you can always defeat him. If you attempt to fight F1, you will get a reward of 5. Fighter 2 (F2) is strong. You can defeat F2 50% of the time. If you attempt to fight F2 you will get either reward of 10 or 0. If F2 is defeated, it pays 10 with probability 2/3. If it is not defeated, it pays 10 with probability 1/3. Assume gamma γ =1 (discount factor)

Q1. What is the maximum expected utility after playing one fight? _____
(1.5 points)

Q2. Whom should you fight first? (One word answer) _____
(1.5 points)

4. Bayes Nets

(4 points – 1 point each)



A	P(B)
T	0.43
F	0.54

A	B	C	P(D)
T	T	T	0.25
		F	0.16
	F	T	0.57
		F	0.85
F	T	T	0.61
		F	0.29
	F	T	0.40
		F	0.18

D	P(E)
T	0.38
F	0.47

Please find the following probabilities and give **the final answer only** in the space provided. **This question does NOT have partial credit**, so please do all the calculations carefully and there is NO need to show your calculations as they will NOT be considered while grading. The answer should be correct up to the 6^{th} decimal place. (example : if your final answer is 0.12 write it as 0.120000, if it is 0.1234567 write it as 0.123457 and so on). Strictly follow the above instructions, else you might not be awarded credit for this question. **Do not use the simulator. Do not round in intermediate steps as that may cause rounding different than ours.**

1. P(E=True|A=False,C=True)= _____
2. P(C=True|A=True, B=False)= _____
3. P(C=True|A=False, E=False)= _____
4. P(D=True|E=True)= _____

5. Machine Learning

(8 points)

Question 1 (2 points):
A configuration of k-Means algorithm corresponds to the k-clusters the algorithm finds at the end of an iteration. Is it possible for k-Means to revisit an already seen configuration? (Revisit means a change from configuration C1 to one or more configurations (!= C1) and back to C1.)

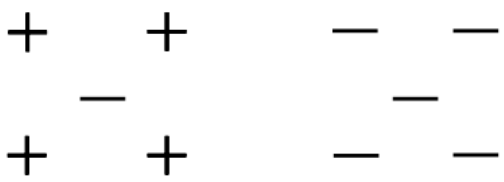
- ❏ No, if we have a consistent tie-breaking criteria
- ❏ Yes, if we have a consistent tie-breaking criteria
- ❏ No, if we do not have a consistent tie-breaking criteria
- ❏ Yes, if we do not have a consistent tie-breaking criteria
- ❏ No, because the cost function would only get optimized if the clustering changes
- ❏ Yes, because the cost function would only get optimized if the clustering changes but you would realise later that an earlier configuration was better

Question 2 (2 point):
Let's assume we have a decision tree to classify binary strings of length 100 (each input has length 100) into K classes. Can we formulate a 1-Nearest Neighbor (1-NN) that could give us the same classification? Specify yes or no and justify in one sentence.

Question 3 (2 points):
You have studied the perceptron rule for training neural nets. Let us apply it for natural language processing task. Let's say we are trying to classify documents. We have K classes. We use a bag of words (BOW) model - we define a Vocabulary of unique words and for each document we count the frequency of all the Vocabulary words in that document. Each word in the vocabulary is a feature and the frequency of each word in the document is treated as the value of that feature. We use a perceptron update for learning the weights from training data. Say, a particular document x has F 'active' features (active features are all the words for which the document has a non-zero value). Given that the size of the vocabulary is V and that your current prediction for x is incorrect, what are the total number of weights that will be updated at this point in the perceptron update? Remember in each update, you would have to increase some weights, decrease some weights.

a) K x V
b) 2 x K x F
c) 2 x V
d) 2 x F
e) 2 x V x F
f) 0

Question 4 (2 points):



You are given the above data points. They belong to two classes. All the "+" are in one class and all the "-" are in the other class. Which out of the below algorithms gives lower leave one out cross-validation (LOOCV) error on this dataset? Also report the cross-validation errors for each.

1. 1-NN LOOCV Error: _____

2. 3-NN LOOCV Error: _____

Which one does better?

---

6. HMMs

(10 points)

You are working for the local government health department. There have been various epidemic scares in the area recently. The government wants to stay ahead of the curve by building a prediction model for the various diseases based on historical data. The indicator variable to be used is the number of people visiting the government health care centers weekly. The government thus intends to run the model every week based on the newly available data to see if the current numbers indicate the presence of an epidemic.

You are tasked with building the model. Being a student of AI, you decide to use a hidden markov model as your prediction model. The historical training data that you have is in the following format,

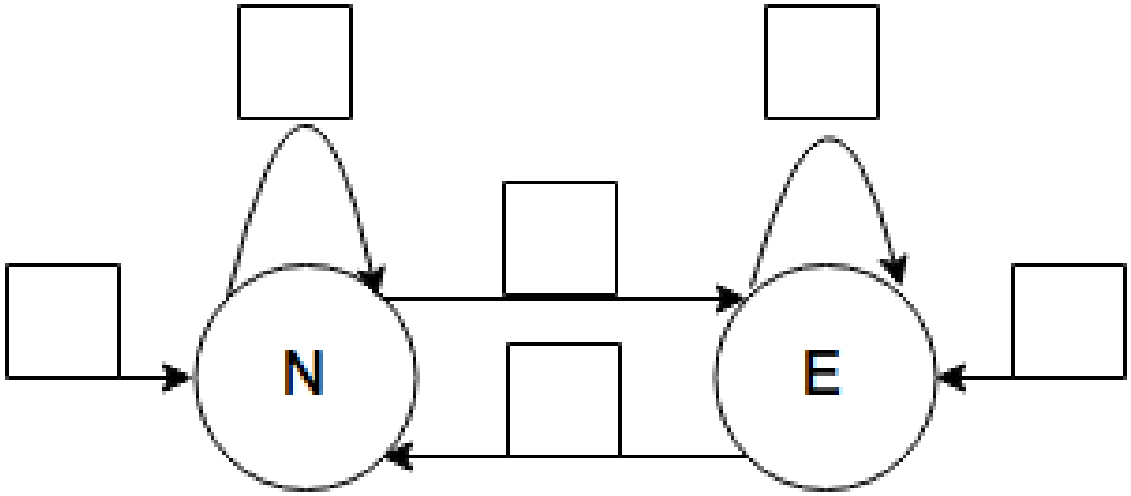
N	E	E	E	E	N	N	E	E	N	E	E
10	98	87	98	78	6	9	78	67	45	67	87

where E indicates the presence of an epidemic and N indicates that it was successfully controlled / not prevalent. The numbers are the number of patients in hospitals.

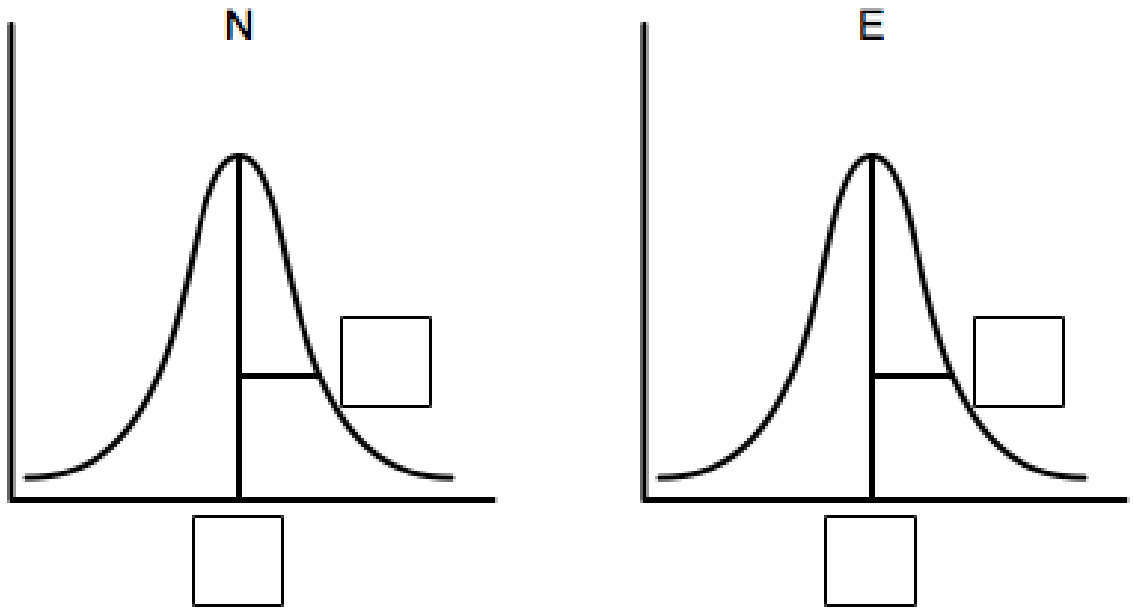
Given the training data, build a Hidden Markov Model using relative frequency estimation for the transition probabilities and model a Gaussian distribution for the continuous emission probabilities. Label the test sequences with E or N using Viterbi decoding.

Find all the data (in separate tabs) here.

Question 1 (2 points):
Fill in the transition model below.



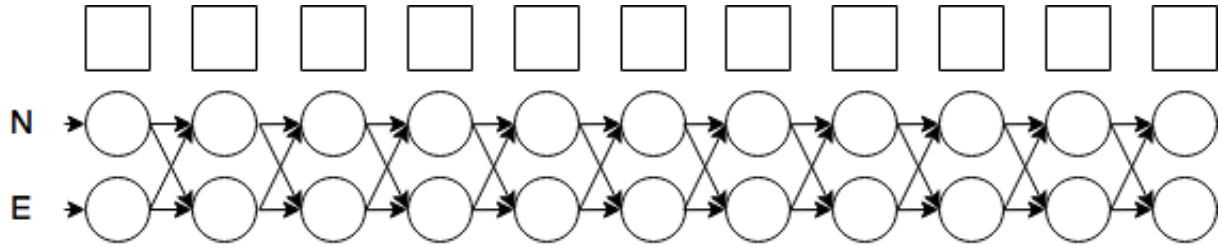
Question 2 (2 points):
Fill in the emission model (mean and std) below.



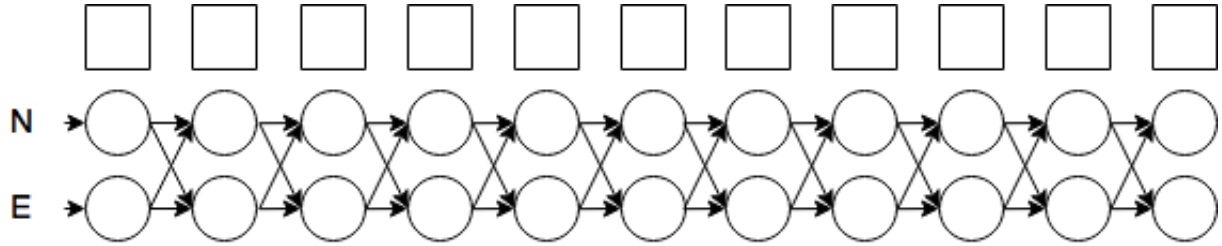
All answers above should be correct up to 2 decimal places. However, it's recommended that you use the entire precision while decoding the sequence.

You can use the Viterbi trellis sketched below for decoding.

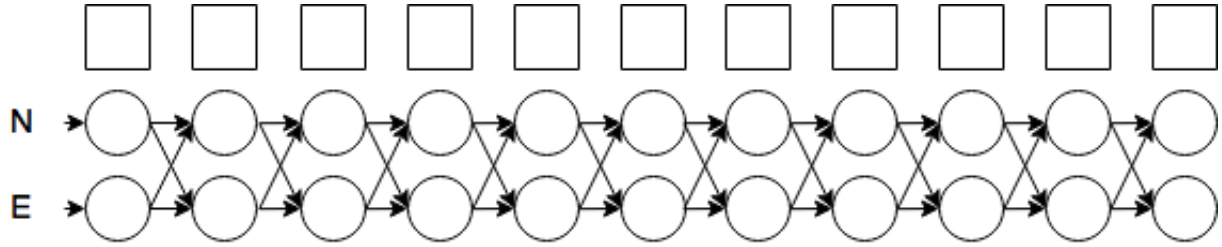
Test Sample 1



Test Sample 2



Test Sample 3



Question 3 (6 points - 2 for each):
Fill in your answers below in the following format -
[N, E, E, E, E, E, E, E, E, E, N]

Test Sample 1
[__, __, __, __, __, __, __, __, __, __, __]

Test Sample 2
[__, __, __, __, __, __, __, __, __, __, __]

Test Sample 3
[__, __, __, __, __, __, __, __, __, __, __]

For those who want a further reading, the inspiration for this question comes from this paper. It will not help you in solving the question though.

7. Logic

(11 points)

In the near future, the Democratic Republic of Starneria is at war with the Autocratic Dictatorship of Isbelland. As an agent working for Starneria's intelligence agency AI-6, your mission is simple: go behind enemy lines and take down the dictatorship by finding evidence linking Isbelland's ML-4 to human rights violations. However, if there is a person who needs to be rescued, they are more important than your mission.

In order to help you, AI-6 has collected the following intelligence about the people in Isbelland: If a person is a spy, their life is in danger. Otherwise, he/she is a commoner and their life is safe. If a person is a commoner or their life is in danger, they need to be rescued. Moreover, spies in Isbelland get some great perks: every spy drives a fast car, and only spies are allowed to use smartphones and smartwatches.

For the first three questions, one or more options are correct. Points will only be awarded if all correct options and no incorrect options are marked.

Question 1 (1 point):
You step out of the airport in Isbelland City and find that the hotel is only ten minutes away on foot. You decide to walk. A car suddenly speeds past you, and you realize it's very *very* fast. What can you say about the driver?

- ❏ The driver is a spy
- ❏ The driver uses a smartphone
- ❏ The driver needs to be rescued

Question 2 (1 point):
On your way to your hotel, you meet a woman on the street. What can you say about her for sure?

- ❏ She is a spy
- ❏ She is more important than the mission
- ❏ She needs to be rescued

Question 3 (1 point):
You walk into the hotel and are greeted by the receptionist. He offers to take your bags to your room, and you notice that he is wearing an old analog watch. Which of the following **can** be true about the receptionist?

- ❏ He is a spy
- ❏ His life is safe
- ❏ The mission is more important than the receptionist

Question 4 (1 point):
After reaching your room, you check in with your handler. She tells you that one sentence in the intelligence you received was actually added by an Isbellan spy just to make you help the enemy. Which sentence is it?

- ❏ If a person is a spy, their life is in danger.
- ❏ If a person is not a spy, he/she is a commoner and their life is safe.
- ❏ If a person is a commoner or their life is in danger, they need to be rescued.
- ❏ Every spy drives a fast car.
- ❏ Only spies are allowed to use smartphones.
- ❏ Only spies are allowed to use smartwatches.

Question 5 (3 points):
After removing the above sentence, using the following symbols:

C: Person is a commoner
D: Person's life is in danger
F: Person drives a fast car
I: Person is more important than the mission
P: Person uses a smartphone
R: Person needs to be rescued
S: Person is a spy
W: Person uses a smartwatch

Prove using resolution that if a person doesn't drive a fast car, they are a commoner.

Question 6 (4 points):
You are given the following predicates:

- **i.** Commoner(x)
- **ii.** Danger(x)
- **iii.** FastCar(x)
- **iv.** Smartwatch(x)
- **v.** Smartphone(x)
- **vi.** User(x, y)
- **vii.** Spy(x)
- **viii.** Rescue(x)

For each of the following logical expressions, state whether it

1. Exactly expresses one of the sentences given in the intelligence
2. Does not exactly express a sentence, but can be inferred from the intelligence
3. Is syntactically valid, but cannot be inferred from the intelligence
4. Is syntactically invalid, and therefore meaningless

You may assume that the arguments are always of the correct type

a. $\forall x, y\ [\ FastCar(x) \wedge User(x, y)\ ] \Rightarrow Spy(y)$ _____

b. $\forall x\ Spy(x) \Rightarrow [\ \exists y\ Smartwatch(y) \wedge User(x, y)\ ]$ _____

c. $\exists x\ \neg Commoner(x) \vee Spy(x) \vee Rescue(x)$ _____

d. $\exists x\ [\ Commoner \vee Danger(x)\ ] \Rightarrow Rescue(x)$ _____

e. $\forall x, y\ \neg Smartphone(x) \vee \neg User(x, y) \vee Spy(y)$ _____

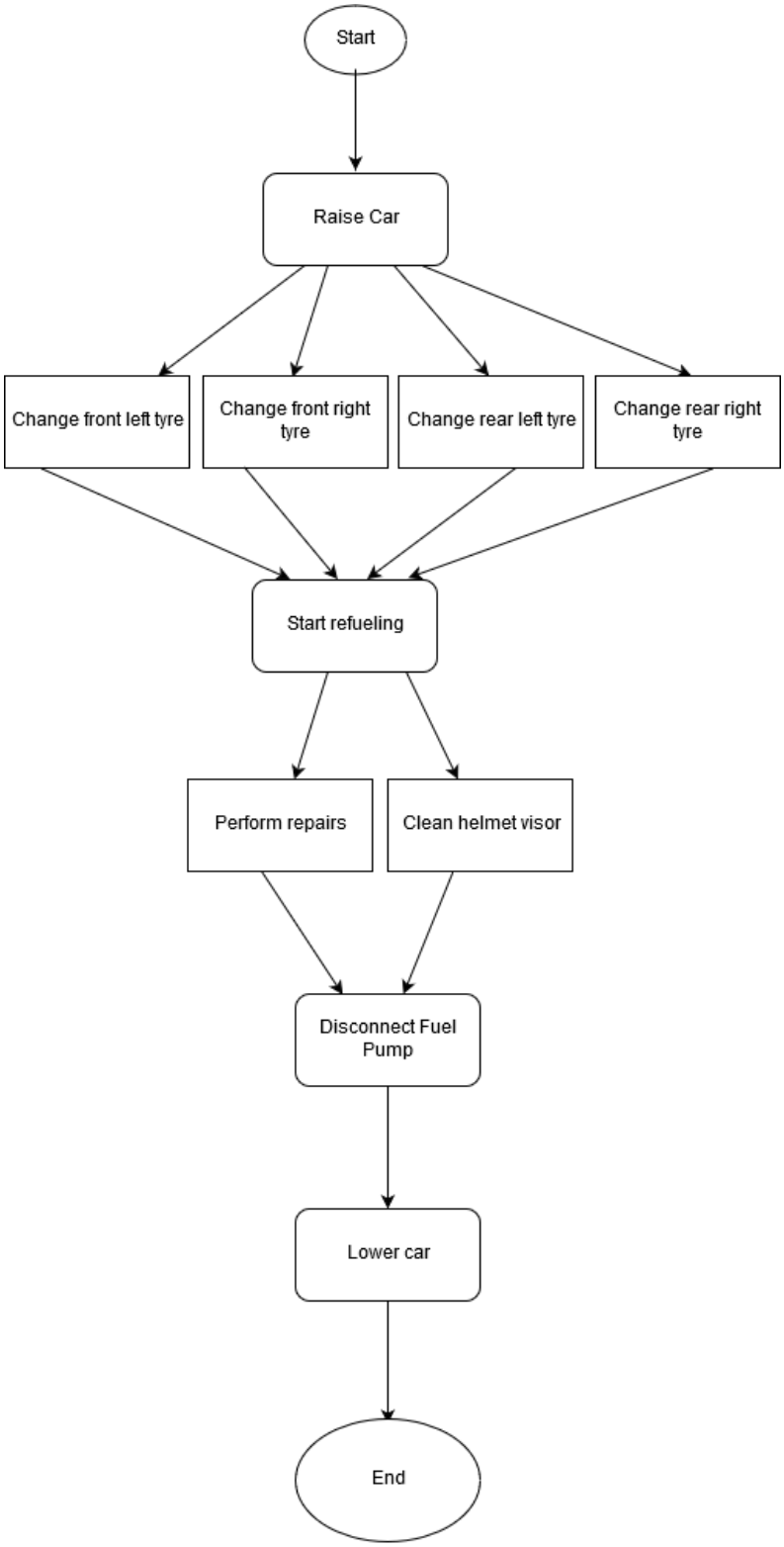
f. $\forall x\ \neg Spy(x) \Rightarrow [\ Commoner(x) \wedge \neg Danger(x)\ ]$ _____

g. $\forall x, y\ [\ FastCar(x) \wedge User(x, y)\ ] \vee Commoner(y)$ _____

h. $\forall x \wedge Spy(x) \wedge FastCar(x) \Rightarrow \neg Commoner(x)$ _____

8. Planning

(3 points)



Pitstops are an extremely vital part of motor racing, where maintenance is performed on the racecar in a matter of seconds by a team of highly-trained professionals. Consider the partially ordered plan for a pitstop given above and answer the following questions.

Question 1 (1.5 points):
Which of the following preconditions would be appropriate for the steps given? One or more options may be correct.

- ❏ Step: Start
  Precondition: In (car, pit lane)
- ❏ Step: Raise Car
  Precondition: Moving (car)
- ❏ Step: Change front left tire
  Precondition: TyreReady (FrontLeft)
- ❏ Step: Change rear right tire
  Precondition: TyreReady (FrontRight)
- ❏ Step: Start refueling
  Precondition: IsReady(Repairman)

Question 2 (1.5 points):
How many totally ordered plans can you create from this partially ordered plan?

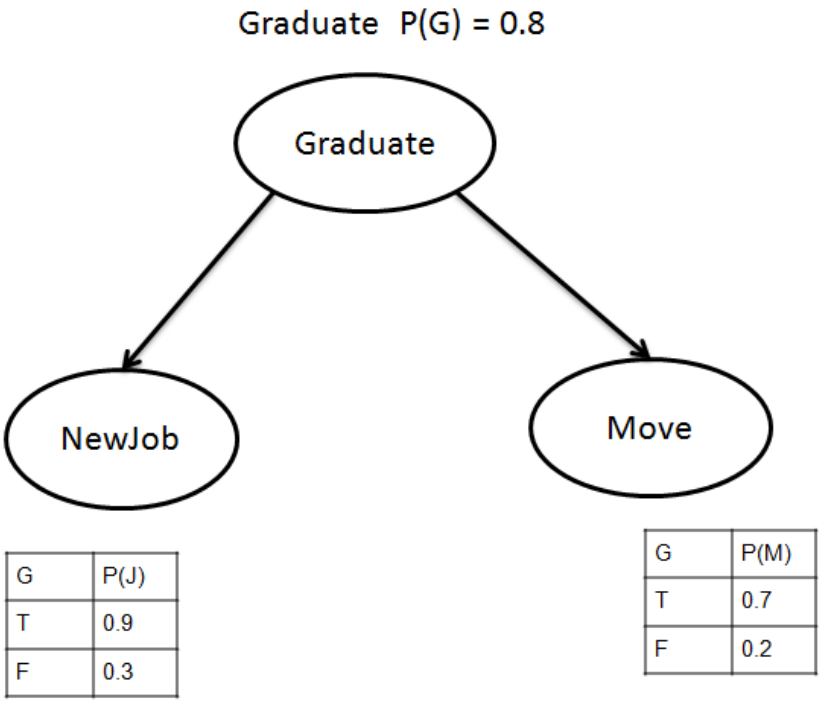
_____

9. D-Separation

(8.5 points)

In the lectures, we saw how d-separation could be used to simplify Bayes Nets calculations. In particular, Sebastian discusses “active triplets” versus “inactive triplets.” With inactive triplets, the middle node separated the two other nodes and made them conditionally independent.

To really understand d-separation, let's take a look at the case B←A→C. Here is an example Bayes Net where we examine the probability that a 4th year undergraduate student graduates (Graduate) which often has an effect on whether they get a new job (NewJob) and move to a new city (Move).



G	P(J)
T	0.9
F	0.3

G	P(M)
T	0.7
F	0.2

Question 1 (4 points):
Create the full joint distribution table for this model

	Graduate		~Graduate	
	Move	~Move	Move	~Move
newJob				
~newJob				

Question 2 (4.5 points):

What is

P(J | G, M) = _____

P(J | G, ~M) = _____

P(J | G) = _____

P(J | ~G, M) = _____

P(J | ~G, ~M) = _____

P(J | ~G) = _____

P(J | M) = _____

P(J | ~M) = _____

P(J) = _____

10. Beyond Classical Search

(15 points)

Question 1:
Consider the following Fitness function for n-bit binary strings: +1 for every adjacent pair of bits that is different. (So, for a bit string 000 or 111 the fitness function would be 0, for 010 and 101 the fitness function is 2 and so on.)
Also, for any n-bit binary string, we are defining its neighbours as all the n-bit binary strings which differ from it at exactly 1 bit position.

A: For n = 5 and starting state 01001, using simple Hill Climbing, which of the following can be the possible next states. **(1.5 points)**

- ☐ 11001
- ☐ 01000
- ☐ 01101
- ☐ 01011
- ☐ 00001
- ☐ All of the above
- ☐ None of the above

B. For n = 5 and starting state 01001, using Simulated Annealing, which of the following can be the possible next states. **(1.5 points)**

- ☐ 11001
- ☐ 01000
- ☐ 01101
- ☐ 00001
- ☐ 11011
- ☐ All of the above
- ☐ None of the above

C: If we try using GA for the above problem, for n = 5, with the starting population of [00000, 11110, 00001, 00010] what is the maximum fitness value that can be achieved given that we always perform a 1-point crossover between 2nd and 3rd bit and we do not perform mutation? **(1 point)**

- 2
- 3
- 4
- 5

Answer: _____

D: If we add a single mutation at every step, what is the maximum fitness value that can be achieved with the same starting population? **(1 point)**

- 2
- 3
- 4
- 5

Answer: _____

Question 2:

"The trade-off between the need to obtain new knowledge and the need to use that knowledge to improve performance is one of the most basic trade-offs in nature, and optimal performance usually requires some balance between exploratory and exploitative behaviors." - Wise men.

In simpler terms:
The exploration-exploitation trade-off is a fundamental dilemma whenever you learn about the world by trying things out. The dilemma is between choosing what you know and getting something close to what you expect ('exploitation') and choosing something you aren't sure about and possibly learning more ('exploration').

The trade-off between Exploration and Exploitation can be tuned through different parameters in different algorithms. Which one do we increase when the following parameters are modified?

Introducing Random restarts in Hill Climbing **(1 point)**

- Exploration
- Exploitation
- Both
- None

Answer: _____

Reducing the number of Mutations in Genetic Algorithms **(1 point)**

- Exploration
- Exploitation
- Both
- None

Answer: _____

Increasing Crossovers in Genetic Algorithms **(1 point)**

- Exploration
- Exploitation
- Both
- None

Answer: _____

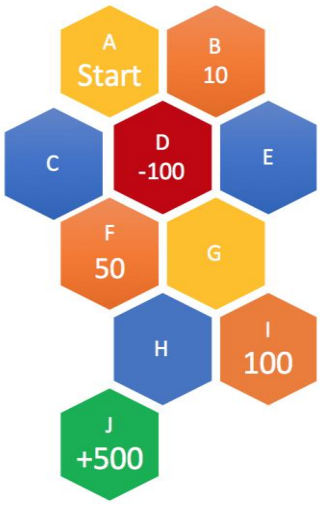
Increasing the acceptance probability of worse samples in MH Sampling to 0.95 **(1 point)**

- Exploration
- Exploitation
- Both
- None

Answer: _____

Question 3:

Consider the following unknown environment for an online search agent. The agent starts in block A. Block J is the Exit block from which the agent can exit the environment. One-time rewards that can be earned at each block are shown. Blocks C, E, G and H offer no reward. At every block, the agent can take a direction from 1 to 6, where 1 represents left, 2 represents upper left and so on in clockwise order.



A Policy is defined as a function which takes a state as an input and outputs an action.
For example, if I am in state A, the Policy will tell me if I need to go in direction 4,5 or 6.
We want to find an optimal policy (i.e., one that maximizes reward) for a search agent in this given environment. This can be found using Value Iteration and Policy Iteration. We can also achieve this by searching through the policy space. We want to use Genetic Algorithms to search through this policy space for an optimal policy.

A: Which of the following could be **one individual** of this population? **(2 points)**

- ❏ Any representation of a path going from start state to end state.
- ❏ A path going from H to D.
- ❏ Every state mapped to one of the possible actions.
- ❏ Path distance calculated by taking all the paths between A and J and aggregating by taking an average of all the path distances.
- ❏ String 1125435122 where each number represents the action the agent would take in A, B, C… and so on respectively, given the problem definition.
- ❏ One individual node.

B: Using a valid representation of an individual from question A, show one complete iteration of Genetic Algorithm for the above scenario. Clearly show each step in the iteration. Population Size is defined as 3. You can pick the number of crossovers and mutation on your own (both must be positive). You need to define a valid fitness function. **(4 points)**

11. Probability

(4 points)

You love adopting pets at the animal shelter down the street.
This shelter has *C* Corgis, *P* Poodles, *L* Black Labs, and *K* Cats.
C >5, *P* >5, *L* >5, *K* >5
There is no partial credit for the following questions.

Question 1 (1 point):

You don't have any pets right now and you don't want to adopt on your own, so you tell your Butler to go to the shelter and **pick out one dog at random**.
What are the odds that the Butler picks out a Poodle?

_____

Question 2 (1 point)

1 month later. The shelter hasn't received any more pets and nobody has taken any pets, and your dog really wants some company. You send the Butler again to **select a pet, but you don't mention whether the pet should be a cat or dog**. What are the odds the Butler picks a cat at random?

_____

Question 3 (2 points)

Having pets is so much fun. You currently have 1 dog and 1 cat from that shelter. The shelter hasn't received any more pets and 2 Poodles have been adopted by another person.
Your dog really wants another dog as company, so you tell your Butler to go to the shelter and **pick out one dog at random**.

Family folklore states that if you have 2 dogs of the same breed you will win the lottery in the near future. What is probability that you are set to fulfill this prophecy? Check one.

- ❏ [*P*/(*C*+*P*+*L*) * *(P-2)*/(*C*+*P*+*L -2*)] + [*C*/(*C*+*P*+*L*) * *(C)*/(*C*+*P*+*L -2*)] + [*L*/(*C*+*P*+*L*) * (*L*)/(*C*+*P*+*L -2*)]
- ❏ [*P*/(*C*+*P*+*L*) * (*P-2*)/(*C*+*P*+*L*)] + [*C*/(*C*+*P*+*L*) * *(C)*/(*C*+*P*+*L*)] + [*L*/(*C*+*P*+*L*) * *(L)*/(*C*+*P*+*L*)]
- ❏ *P*/(*C*+*P*+*L*) + *C*/(*C*+*P*+*L*) + *L*/(*C*+*P*+*L*)
- ❏ [*P*/(*C*+*P*+*L*) * *(P)*/(*C*+*P*+*L -2*)] + [*C*/(*C*+*P*+*L*) * *(C)*/(*C*+*P*+*L -2*)] + [*L*/(*C*+*P*+*L*) * *(L)*/(*C*+*P*+*L -2*)]
- ❏ [*P*/(*C*+*P*+*L*) * *(P-3)*/(*C*+*P*+*L -3*)] + [*C*/(*C*+*P*+*L*) * *(C-1)*/(*C*+*P*+*L -3*)] + [*L*/(*C*+*P*+*L*) * (*L-1*)/(*C*+*P*+*L -3*)]

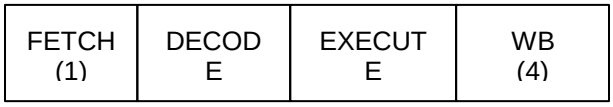
12. Constraint Satisfaction

(13.5 points)

Instruction Scheduling for Pipeline Microprocessor

Background

Modern microprocessors implement a pipeline model of instruction execution. In this model, a single instruction is broken into stages. In the example below, the microprocessor has 4 stages in its pipeline: FETCH - DECODE - EXECUTE - WB (WRITEBACK). As an instruction flows through the pipeline (e.g. FETCH to DECODE stage), the next instruction enters the FETCH stage, and so on. The pipeline acts like an assembly line for executing a series of instructions, leading to significant performance gains at runtime.



FETCH	Instruction is fetched from memory
DECODE	Instruction is decoded. Determines which execution unit will execute it and what operands are needed
EXECUTE	Functional unit executes the instruction
WB	Execution results are written back to storage (memory, registers)

In-order execution: Instructions are executed in the order in which they are fetched from memory (program or lexical order)

Out-of-order execution: Instructions may be executed in a different order than that listed in the program. This is usually done to reduce the total number of cycles needed to execute the entire program. This is achieved by utilizing parallel functional (execution) units in the hardware.

Instruction Scheduling Constraints

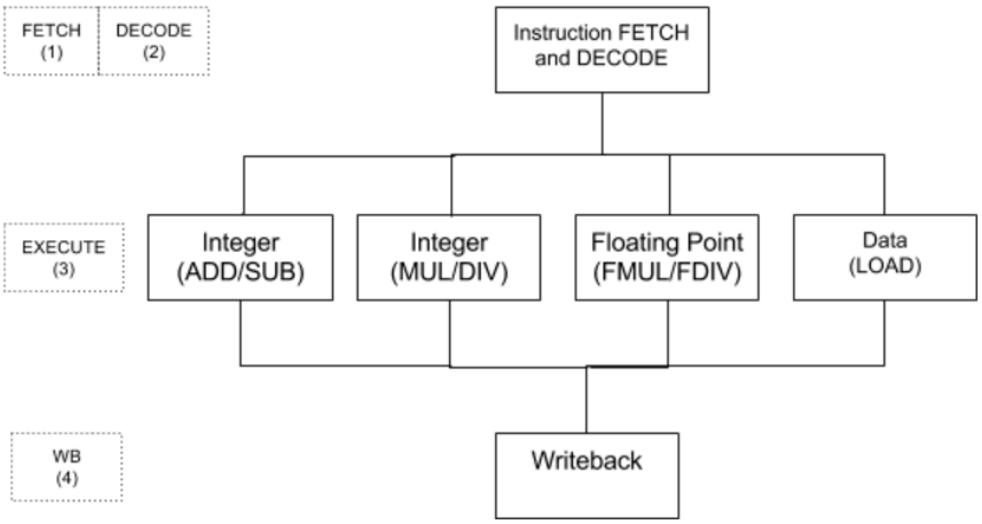
- *Procedural/Control dependencies:* are due to the nature of the instruction e.g. BRANCH (BRA, BEQ, BNE), JUMP, RETURN (RET), etc.
- *Resource conflicts:* instructions that use same shared resource (e.g. a hardware functional unit) cannot be executed simultaneously
- *True data dependencies:* cannot execute an instruction until all required operands are available and updated with correct values. E.g., **r1** := r2 + r3 and r5 := 5 * **r1** so second instruction has true data dependency on the first instruction
- *Storage conflicts:* value in a storage (register, memory) must have the most recent assignment to that location

Problem

You are writing a microprocessor simulator program that needs to simulate the scheduling of instructions in hardware. For this question, the only concern is to schedule instructions so they can be executed correctly by the execution units and the results written back to storage (registers or memory). The instructions must be scheduled so that they do not violate any of the constraints imposed by the hardware. These hardware constraints can vary from processor to processor so the software architect designing the simulator has decided to create a generic constraint satisfaction solver program that can work for any type of microprocessor given a set of hardware-specific constraints.

Your boss, the Chief Software Architect, has asked for someone who can take the lead role in writing the constraint satisfaction-based instruction scheduler. You recently aced the OMSCS AI course, and sensing a career growth opportunity, eagerly volunteer for the role. Now your job is to write the constraints for the specific microprocessor architecture described below and to answer the questions given below.

The *TECH6601S17* Microprocessor: Selected Functional Units



Notes:

The block diagram above shows only selected functional units relevant to this problem. Each block is a functional unit (FETCH and DECODE are two functional units in the same block).
Pipeline stages are: `(1)FETCH (2)DECODE (3)EXECUTE (4)WRITEBACK`
Instruction types and the execution units where they are dispatched are shown in the boxes above.
All functional units (boxes with solid borders) run concurrently in hardware. Assume "Instruction FETCH and DECODE" unit can feed instructions to ALL execution units in the SAME cycle if the execution unit is not busy. Dotted boxes show pipeline stages.

Number of Cycles in EXECUTE Stage by Instruction Type

Instruction Type	**# of cycles (EXECUTE Stage only)**	**Example**
`Integer (ADD/SUB)` `Type = IADD`	1	`ADD r1, r2, r3` `// r1 := r2 + r3`
`Integer (MUL/DIV)` `Type = IMUL_DIV`	5	`MUL r3, r2, 10` `// r3 := r2 * 10`
`Floating Point (FMUL)` `Type = FMUL`	7	`FMUL  f.r1, f.r2, f.r3` `// f.r1 := f.r2 * f.r3`
`Floating Point (FDIV)` `Type = FDIV`	8	`FDIV   f.r6, 16, f.r1` `// f.r6 := 16.0 / f.r1`
`Data (LOAD)` `Type = DATA`	3	`LDW r1, 0xABCD,r2` `// r1 := MEMORY(0xABCD + r2)`

`r0, r1, r2, r3, ...` are integer registers.
`f.r0, f.r1, f.r2, ...` are floating point registers. They are independent of integer registers (no dependency with `r0, r1, r2, r3,` etc.)
Writeback stage for all instructions takes 1 cycle to complete.
Functional units can execute instructions out-of-order as long as they meet all specified constraints.

Scheduling Constraints to consider in this problem (see above for definition):

- Procedural dependencies: out of scope -- not applicable for this problem
- Resource conflicts: YES
- True data dependencies: YES
- Storage conflicts: YES

For a general Constraint Satisfaction Problem (CSP):

X is a set of n **variables**, X1, X2, …Xi,.. Xn each defined by a finite **domain** D1, D2, …., Dn possible values.

C is a set of **constraints** C1, C2, …, Cm such that Cm involves a subset of variables of set X and it specifies the allowable combination of values for that subset of variables.

A **solution** to the problem is an assignment of values to the variables that satisfy **ALL** constraints.

NOTE: This CSP problem requires you to only simulate instruction scheduling for the EXECUTE and WRITEBACK stages of the pipeline. Assume instructions are ready at the DECODE stage to be dispatched (executed) when the required functional unit is available.

CSP Variable & Domain Definition:

Given these sets:

```
X = { X1, X2,..., Xi }
      where Xi represents a single program instruction
Instr_type = { IADD, IMUL_DIV, FMUL, FDIV, DATA }
Cycles = { 1, 2, 3, …, N }
      where N is an Integer and equal to the maximum number of cycles
to execute a given program.
```

Each instruction in the program is assigned a unique variable, Xi which has the following attributes:

```
Xi attributes:
      X.type              ∈ Instr_type
      X.start_exe_cycle   ∈ Cycles     // cycle no. when instruction begins EXEC
      X.end_exe_cycle     ∈ Cycles     // cycle no. when instruction ends EXEC stage
      X.wb_cycle          ∈ Cycles     // cycle number when WRITEBACK stage ends
```

Attribute X.type is assigned when program is loaded and remains unchanged for the duration of the simulation.

Useful Function Definition:

```
Alldiff ( {v1, v2, v3, … } ):    boolean
//  returns true if all elements in the input set are different;
false otherwise

assume i and j are integers
```

Question 1.1 - Constraints Definition (1 point):
∀Xi stands for all Xi. ∈stands for "element of"

1.1	Which of the following constraint(s) are valid? Select all answers that are correct.
1	`Xi.end_exe_cycle - Xi.start_exe_cycle >= 5  for ∀ Xi: Xi.type = IADD`
2	`Xi.end_exe_cycle + Xi.start_exe_cycle >= 5  for ∀ Xi: Xi.type = IADD`
3	`Xi.end_exe_cycle + Xi.start_exe_cycle + Xi.wb_cycle >= 5  for ∀ Xi: Xi.type = IADD`
4	`Xi.end_exe_cycle - Xi.start_exe_cycle >= 5  for ∀ Xi`
5	`Xi.end_exe_cycle - Xi.start_exe_cycle >= 5  for ∀ Xi: Xi.type ∈ { IMUL_DIV, FDIV, FMUL}`

Answer: _____

Question 1.2 (1 point):

1.2	Which of the following constraint(s) are valid? Select all answers that are correct.
1	`alldiff(Xi.end_exe_cycle)  for ∀ Xi: Xi.type = IMUL_DIV`
2	`alldiff(Xi.start_exe_cycle + Xi.end_exe_cycle)  for ∀ Xi: Xi.type = DATA`
3	`alldiff(Xi.start_exe_cycle)  for ∀ Xi: Xi.type = IADD`
4	`alldiff(Xi.start_exe_cycle)  for ∀ Xi`
5	`alldiff(Xi.start_exe_cycle)  for ∀ Xi: Xi.type ∈ { FDIV, FMUL}`

Answer: _____

Question 1.3 (1 point):

1.3	Which of the following constraint(s) are valid? Select all answers that are correct.
1	`Xi.wb_cycle < Xj.wb_cycle  for ∀ Xi, for ∀ Xj:  Xi.type = Xj.type`
2	`alldiff(Xi.wb_cycle)  for ∀ Xi`
3	`alldiff(Xi.wb_cycle + Xi.end_exe_cycle)  for ∀ Xi`
4	`Xi.wb_cycle = Xi.end_exe_cycle + 1  for ∀ Xi`
5	`Xi.wb_cycle >= Xi.end_exe_cycle + 1  for ∀ Xi`
	This row has been left blank intentionally.

Answer: _____

Now we need to deal with **true data dependencies** between instructions. Remember that this means that the processor cannot execute an instruction Xi until all referenced operands (r1, r2, …) of that instruction are available and updated with correct values according to the program flow.
For example:

```
Instr #1:       r1 := r2 + r3          // r1 is updated AFTER WB cycle
                                       // is complete.
Instr #2:       r5 := 5 * r1           // instruction #2 has true data
                                       // dependency on register r1
```

r1 is updated in instruction #1 at the end of WB cycle. So, Instr #1 WB cycle must complete before Inst #2 can start EXEC stage due to true data dependency.

```
Xi additional attribute:
     X.true_data_deps ∈ X        // set of all dependent instructions
                                 // see definition of X above
```

Question 1.4 (1 point):

1.4.	Which of the following constraint(s) for **true data dependencies** are valid? Select all answers that are correct.

Functions (given)	`all_deps (Xi): set`
	`// returns a subset of all X that are true data dependencies of Xi`
	
	`all_deps_recursive (Xi): set`
	`// returns subset of all X that are true data dependencies of Xi or data dependencies of dependencies (recursively up to first program instruction)`

1.	`Xi.start_exe_cycle < Xj.start_exe_cycle  for ∀ Xi, for ∀ Xj: Xj ∈ { all_deps_recursive(Xi) }`
2	`Xi.start_exe_cycle < Xj.start_exe_cycle  for ∀ Xi, for ∀ Xj: Xj ∈ { all_deps(Xi) }`
3	`Xi.start_exe_cycle > Xj.start_exe_cycle  for ∀ Xi, for ∀ Xj: Xj ∈ { all_deps_recursive(Xi) }`
4	`Xi.start_exe_cycle > Xj.start_exe_cycle  for ∀ Xi, for ∀ Xj`

Answer: _____

A single Writeback (WB) stage of the pipeline is used by all instruction types. An enhancement in the hardware allows it to assign priority to which functional unit can use the WB stage in case there is a conflict for the current cycle.

WB priority is assigned as follows:

```
Priority(Floating Pt. instr. FMUL, FDIV)> Priority(IMUL_DIV)
Priority(IMUL_DIV) > Priority(DATA)
Priority(DATA) > Priority(IADD)
```

```
In case of a WB conflict, instruction with an earlier start_exe_cycle
will WB before instruction with later start_exe_cycle.
```

Question 1.5 (1 point):

1.5	Which of the following constraint(s) for **WB cycle priority** are valid? Select all answers that are correct.

1	`Xi.wb_cycle < Xj.wb_cycle  for ∀ Xi, Xj:`
	`Xi.end_exe_cycle == Xj.end_exe_cycle and`
	`Xi.type ∈ {IMUL_DIV} and Xj.type ∉ {FMUL, FDIV, IMUL_DIV}`

2	`Xi.wb_cycle <= Xj.wb_cycle  for ∀ Xi,Xj:`
	`Xi.end_exe_cycle == Xj.end_exe_cycle and`
	`Xi.type ∈ {FMUL, FDIV} and Xj.type ∉ {FMUL, FDIV}`

3	`Xi.wb_cycle < Xj.wb_cycle  for ∀ Xi,Xj:`
	`Xi.end_exe_cycle == Xj.end_exe_cycle and`
	`Xi.type ∈ {FMUL, FDIV} and Xj.type ∉ {FMUL, FDIV}`

4	`Xi.wb_cycle > Xj.wb_cycle  for ∀ Xi,Xj:`
	`Xi.end_exe_cycle == Xj.end_exe_cycle and Xi.type == Xj.type and`
	`Xi.start_exe_cycle > Xj.start_exe_cycle`

Answer: _____

Write any other additional constraints for this problem in the box below. This is to help you work out the next section correctly. This part will **NOT** be graded.

Question 2 - Solving the CSP:

Besides the constraints given in the multiple-choice questions above, list down all other constraints for your reference (based on processor architecture description above) as you work out a solution to this part. You can look at assembly instruction or equivalent comment after "//". Hint: Second instruction (LDW) has a true data dependency on r0 which is set in first instruction

```
ADD         r0, 0,1          // r0 := 0 + 1
LDW         r1, 0x1000,r0    // r1 := MEM(0x1000 + r0)
LDW         r2, 0x4,r1       // r2 := MEM(0x4 + r1)
ADD         r3, r0, 1        // r3 := r0 + 1
MUL         r4, r0, 2        // r4 := r0 * 2
FMUL        f.r0, 2.0,2.0    // f.r0 := 2.0 * 2.0
FDIV        f.r1, f.r0, 5.0  // f.r1 := f.r0 / 5.0
LDW         r4, 0x2000       // r4 := MEM(0x2000)
LDW         r3, 0x2004       // r3 := MEM(0x2004)
ADD         r3, r3, r4       // r3 = r3 + r4
```

Question 2.1 (4.5 points):
Assign valid values to each variable (attributes `start_exe_cycle, end_exe_cycle, wb_cycle)` given the constraints. Cycles start with 1. If instruction is 1 cycle long, Start EXEC and End EXEC will have the same cycle number as shown below.

Variable	`Xi.start_exe_cycle`	`# of EXEC cycles for Xi`	`Xi.end_exe_cycle`	`Xi.wb_cycle`
X1	1	1	1	2
X2	3	3		
X3		3		
X4		2		
X5		5		
X6		7		
X7		8		
X8		3		
X9		3		
X10		1		

Hint: You may use the following table as a worksheet. This table will **NOT be graded**.
For each box, fill in E if X in EXEC stage; W if X in WB stage

Scratch Worksheet (not graded)

Cycle #-->	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**...**	**...**	**...**
X1	E	W									
X2			E	E	E	W					
X3											
X4											
...											

Question 2.2 (1 point):
Answer these questions about your solution:

2.2.1. Total number of cycles for all program instructions (EXEC + WB stages only) = _____ cycles

2.2.2. Average Cycles per Instr (CPS) = _____ cycles per instruction (EXEC + WB only)

Question 3.1 (1.5 points):
Can you provide one **preference constraint** to minimize the total number of cycles (EXEC and WB stages only) for the entire program?

Question 3.2 (1.5 points):
4.3. Explain which of the following variable assignment heuristics would you use for this problem:

- ❑ Minimum Remaining Values (MRV)
- ❑ Minimum Conflicts

Why?

References (if you need more background):

- See Wikipedia article, Instruction Pipelining, https://en.wikipedia.org/wiki/Instruction_pipelining
- Lee, Peter. Introduction to Instruction Scheduling, CMU, Spring 2006 Lecture 11 Notes. www.cs.cmu.edu/afs/cs/academic/class/15745-s06/web/handouts/11.pdf

13. Bonus (1 point)

How many questions were on the CIOS for this course? If you already completed the survey and did not make note of this, please include a screenshot of your CIOS homepage for credit.

Answer: _____

And now mark the checklist below making sure you have taken care of each of the points mentioned:

- ☐ All pages are being uploaded in the correct order that they were presented to you.
- ☐ Any extra pages are only attached at the END of this exam, after page 41 with clear pointers to wherever the actual answer is in the PDF (reference properly)
- ☐ Are you submitting only one pdf and nothing else (no docx, doc, etc)?
- ☐ Did you avoid putting your name on the exam?
- ☐ Is the pdf you are submitting not blank (unless you want it to be)?
- ☐ **Did you go over the uploaded pictures on Gradescope and made sure that all the answers are clearly visible? Dull scans will not be graded.**