CS 6601 Final – Fall 2021

Please read the following instructions thoroughly.

Fill out this PDF form and submit it on **Gradescope**. Remember to also submit on Canvas. **You will be penalized with 5 points on this exam if you don't submit on both the platforms.**

You have unlimited resubmissions until the deadline. You can: **(a)** type directly into the form – we highly recommend using Adobe Reader DC (or Master PDF 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.

Submit only a single PDF – no phone pictures, please! (You may use an app like CamScanner or Office Lens 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, except for e-text versions of the textbook, this semester's CS6601 course materials, Piazza, and any links provided in the PDF itself. No resources outside this semester's 6601 class should be used. Do not discuss the exam on Piazza, Slack, or any other form of communication. More generally, do not post **publicly** about the exam. 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** (for example, a question on Search would be "Final Exam #2"). Please make **different posts for different questions.**

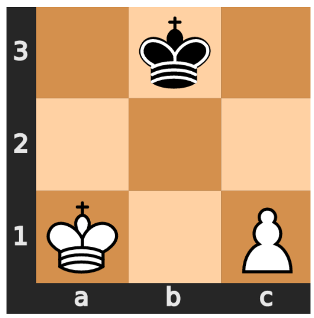
Please round all your final answers to 6 decimal places, don't round intermediate results. You can use `round(your_number, 6)` function in Python for help.
You will not receive full credit if your answers are not given to the specified precision.

Point breakdown (Each question has sub-parts with varying points):

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Total
Pts	8	6	8	6	6	8	12	16	12	18	100

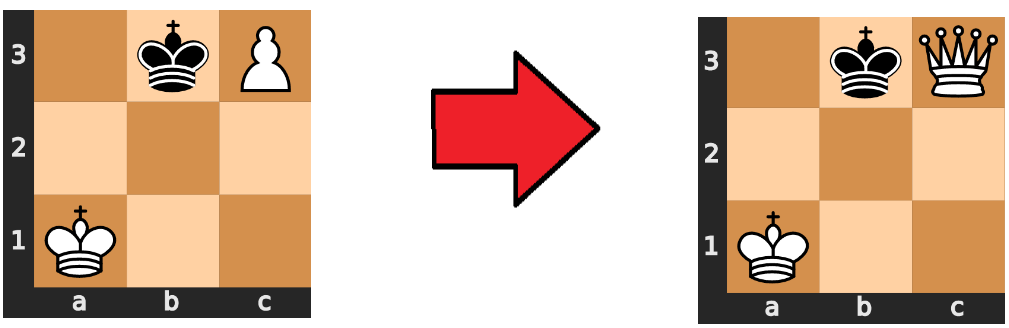
1. Game Playing
(8 points)

Imagine a game of mini-chess played on a 3x3 board as shown below. In this game White has a king and pawn on squares a1 and c1 respectively and Black has a king on square b3. **It is White's turn to play.** White will be said to win this game if it checkmates Black's King. A checkmate occurs when Black's King is threatened by another piece and Black has no legal moves to escape. Black will be said to win the game if it can force a stalemate against White. A stalemate occurs when Black is not in check but has no legal moves. A stalemate will also occur if Black captures White's non-King piece (see below for an example of checkmate and stalemate.)



The starting board state for mini-chess

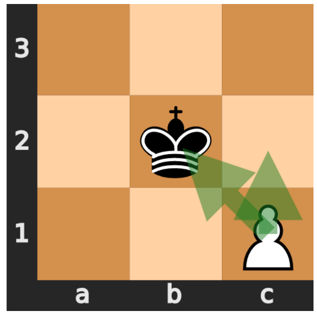
In this version of chess, pawn promotion is still possible. If White's pawn reaches the last row (i.e. the square c3) then the white player can choose to exchange the pawn for one of four pieces (queen, rook, bishop, or knight). **Assume that a pawn reaching the back row must be promoted to a Queen.**



A pawn being promoted to a queen once reaching the back row

There are 3 possible pieces on the board; the pawn, the king, and the queen(after pawn promotion). A board square can only have one piece occupying it at a time. None of these pieces can jump over another piece.

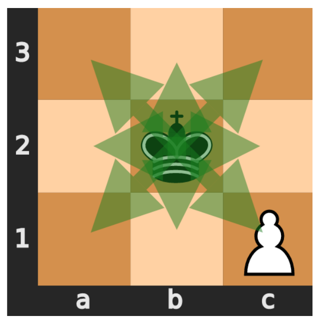
- **The pawn can only move 1 space at a time** towards the top of the board when it is not attacking. Pawns moving forward two spaces in one turn do not exist in mini-chess. The pawn can only attack diagonally one square.



The pawn can move forward 1 square or attack diagonally 1 square

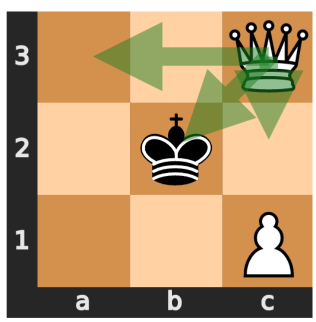
•

- The king can move 1 space to any adjacent square on the board. **Kings are not allowed to move to a space that can be attacked by another piece. This move would be considered illegal.**



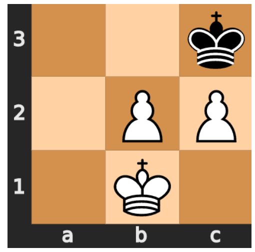
The king can move one square to any space

- The queen can move along a row, column, or diagonal.

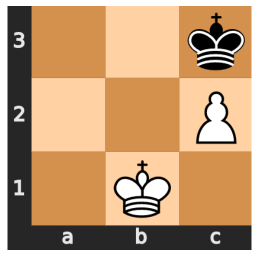


The queen can move to any of these squares from c3

Below are examples of checkmate(White wins) and stalemate(Black wins)



Checkmate: The Black King is under attack by the pawn at b2. Any move it makes will result in the potential for White to capture it.

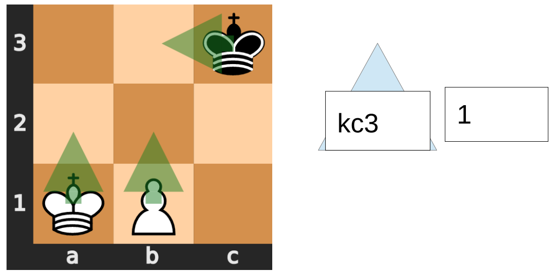


Stalemate: The Black King is not under attack but it cannot move without putting itself in a vulnerable position.

Below is a game tree. While filling out the tree observe following rules:

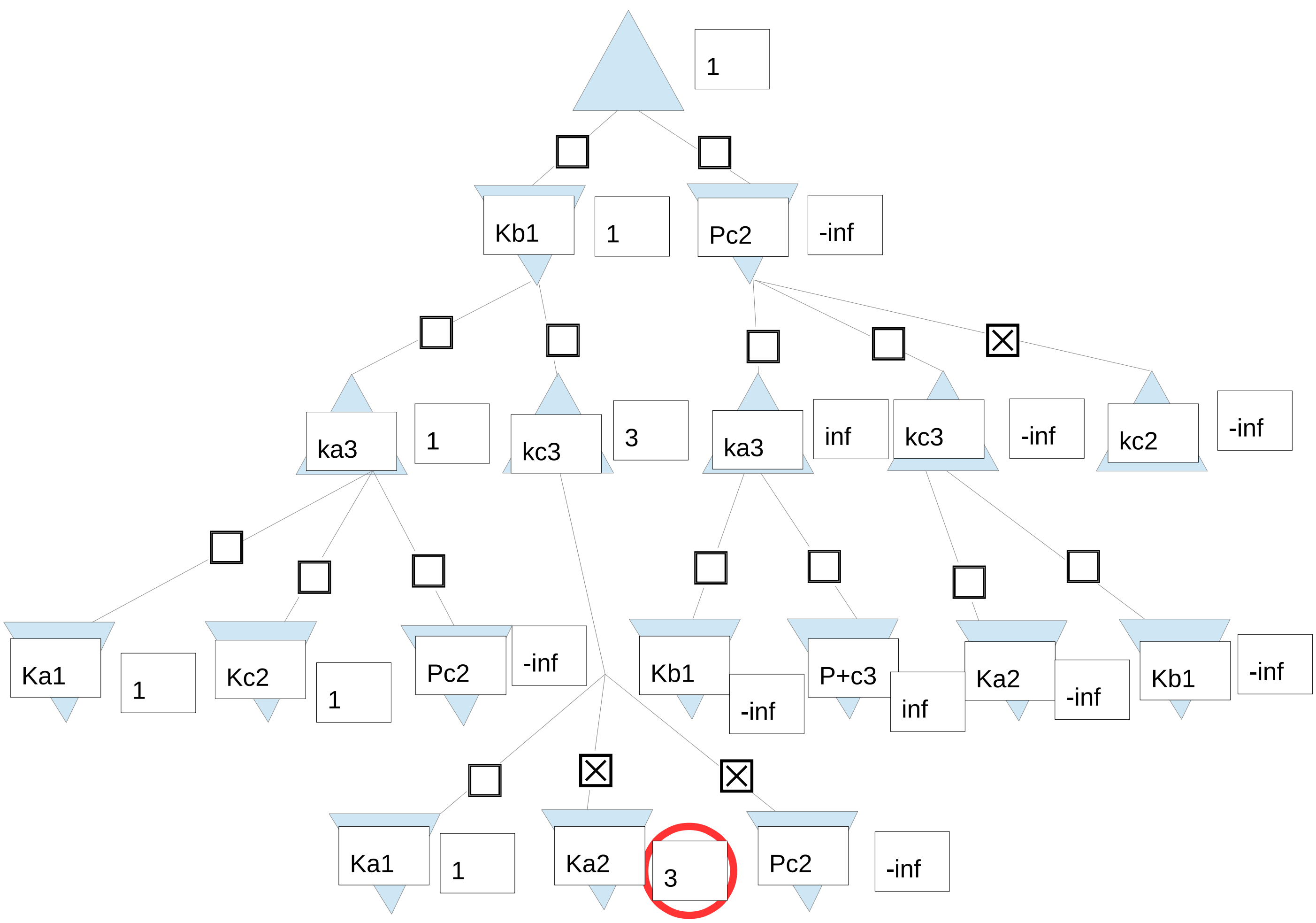
1. The tree is filled out from left- to-right
2. The king's moves are always evaluated first before the pawn's.
3. The king's moves are evaluated starting with the left-most legal move and then proceeding in a clockwise direction.
4. If White's Pawn gets to the back row it gets promoted in the same turn . The symbol for the promoted Pawn should be P+. For example White's Pawn on the back row would be written P+c3.
5. You don't have to include any of the chess notation for situations such as capture, checkmate, check etc. Only write in the piece and its board position.
6. The evaluation function will be Eval(board) = num. of White's legal moves – num. of Black's legal moves.
7. If you reach an endgame state the evaluation will be inf if White has a checkmate and -inf if Black forces a stalemate.

Within each triangle write the legal move. Moves are written in a Simplified Algebraic Notation (SAN). In SAN a move is composed of two parts; the piece and the square it has moved to. For example, if the White King moves to a3 write Ka3. When White moves the piece is written in uppercase and when Black moves it is written in lowercase. Therefore, if it was the Black King which moves to a3 it is written ka3. Write the move within the triangle and the score from the evaluation function on the right.



Example node for board after Black's King moves to c3 with score = 2 white_moves – 1 black_move. Potential moves for each side are shown as green arrows.

Fill out the game tree using the minimax algorithm and following the rules of mini-chess. The values for each node won't be graded; only the checkboxes for Q1.3. The tree is there to assist you when answering the questions.



Q1.1 What is score for the circled node? **(2 points)**

3

Q1.2 What is the average branching factor of the game up to a depth of 3? **(1 point)**

2.428571

Q1.3 Perform Alpha-Beta Pruning. Click the check-box on the branch if it is pruned according to the Alpha-Beta algorithm. **(2 points)**

Q1.4 According to the evaluation function and the minimax algorithm what first move should White make? **(1 point)**

Kb1

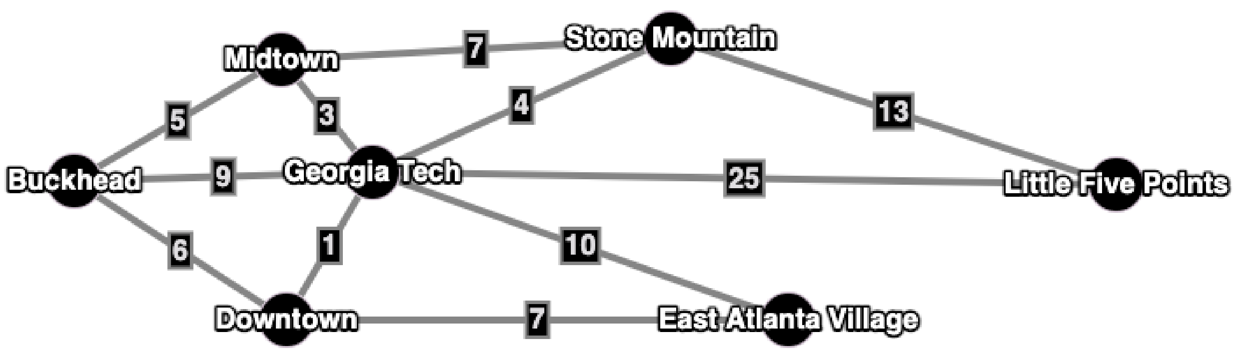
Q1.5 Assume that this game is played using a full sized 8x8 chess board. The Black King is placed at one end of the board and White's King and Pawn at the other (their exact positions don't matter). We want to run our minimax algorithm to endgame to find the optimal plays. Assuming that you don't have to worry about available computing power will the minimax algorithm find the optimal play? (Select only one) **(2 points)**

- ○ No, the branching factor is too high
- ○ No, the evaluation function only works with mini-chess
- ◉ No, the algorithm searches to infinite depth
- ○ Yes, the mini-max algorithm witll find the optimal play

2. Search

(6 points)

The diagram below shows a network of roads and highways. The number on each edge represents the time (in minutes) to drive from one neighborhood of Atlanta to another.



Q2.1 Use the Uniform-cost search algorithm to find the shortest path between Buckhead and Little Five Points. What is the total time of this path? **(0.5 points)**

24

Q2.2 What is the corresponding route found by the Uniform-cost search algorithm? Answer as a comma separated list, e.g. "Downtown, Buckhead, ... " **(0.5 points)**

Buckhead, Downtown, Georgia Tech, Stone Mountain, Little Five Points

Q2.3 Imagine we run Breadth First Search on this graph starting at any vertex. Which of the following is a possible order for visiting the nodes? Do not repeat nodes. You should consider neighbors in alphabetical order, meaning Downtown would be expanded before East Atlanta Village if both nodes were next possible options to be expanded. **(1.5 points)**

- ○ Midtown -> Buckhead -> Downtown -> East Atlanta Village -> Georgia Tech -> Little Five Points -> Stone Mountain
- ○ Stone Mountain -> Midtown -> Georgia Tech -> Buckhead -> East Atlanta Village -> Downtown -> Little Five Points
- ◉ Little Five Points -> Georgia Tech -> Stone Mountain -> Buckhead -> Downtown -> East Atlanta Village -> Midtown
- ○ Downtown -> Buckhead -> East Atlanta Village -> Georgia Tech -> Little Five Points -> Stone Mountain -> Midtown

Q2.4 Now, let's run Depth First Search. We will start at East Atlanta Village and traverse the entire graph. In what order will nodes be expanded? Do not repeat nodes. Again, you should ignore the weights of the graph, and break ties alphabetically. Answer as a comma separated list, e.g. "Downtown, Buckhead, ... " **(1 point)**

East Atlanta Village, Downtown, Buckhead, Georgia Tech, Little Five Points, Stone Mountain, Midtown

Q2.5 A new highway is being built between East Atlanta Village and Little Five Points. The time to drive along this highway is still unknown, but you know it will reduce the amount of time taken on the quickest path between Buckhead and Little Five points. What is the maximum amount of time, in minutes, that this path can be for it to be included in this new shortest path between the two aforementioned points using the Uniform-cost search algorithm? Consider only integer values as the solution. **(2.5 points)**

10

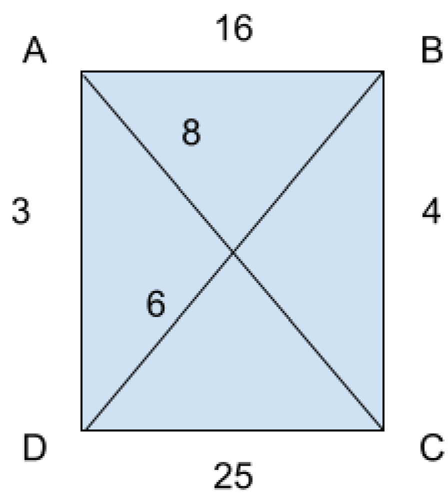
3. Optimization Algorithms

(8 points)

You get your hands on a galactic map that shows the locations of treasures. Assume that your starting position is your home star system and you must ultimately return to that position.

We will use simulated annealing to minimize the total distance traveled.

Figure - Galactic treasure map. Star systems A, B, C, and D with respective distances.



Rules for valid paths using the galactic treasure map:

- You must visit every star system exactly once
- You must return to your home star system
- Our objective function will be defined as

$$E = Total\ Distance(path) = \sum_{i=1}^{i=N-1} distance(path_i, path_{i+1})$$

- Our goal is to minimize our objective function

Part A

(1.5 points)

Q3.A.1 How many valid paths exist on our galactic treasure map? **(1 point)**

24

Q3.A.2 Given the path [A - B - C - D - A], what is E (value of objective function)? **(0.5 points)**

48

Part B

(1 point)

For the following questions, compute the acceptance probabilities.

$$P = e^{-\frac{E_2 - E_1}{T}}$$

- When probability is greater than 1, input 1
- Let our temperature $T = 10$
- Let E_1 be the value of our current and E_2 be our proposed path

Q3.B.1 What is the probability of acceptance from [A - B - C - D - A] to [A - B - D - C - A]? **(0.5 points)**

0.496585

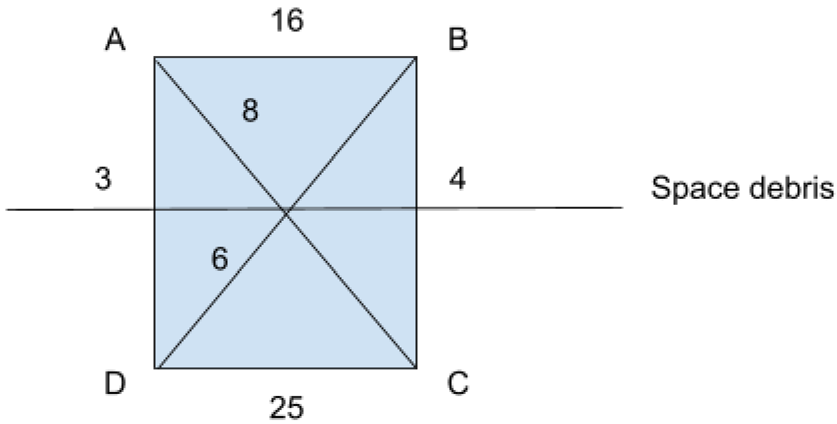
Q3.B.2 What is the probability of acceptance from [A - B - C - D - A] to [A - C - B - D -A]? **(0.5 points)**

1.0

Part C

(3.5 points)

Assume there is space debris as illustrated by the figure below.



Q3.C.1 Which objective function would you choose given you wish to minimize crossing space debris? **(1.75 points)**

- (selected) Let α be a factor that can be tuned. Let $\beta = -1$ for A and B. Let $\beta = 1$ for C and D. $E = \sum_{i=1}^{i=N-1} [distance(path_i, path_{i+1}) + \alpha(\beta_i - \beta_{i+1})^2]$
- Let α be a factor that can be tuned. Let $\beta = -1$ for A and B. Let $\beta = 1$ for C and D. $E = \sum_{i=1}^{i=N-1} [distance(path_i, path_{i+1}) - \alpha(\beta_i - \beta_{i+1})^2]$
- Let α be a factor that can be tuned. Let $\beta = -1$ for A and B. Let $\beta = 1$ for C and D. $E = \sum_{i=1}^{i=N-1} [distance(path_i, path_{i+1})] + \alpha(\beta_i - \beta_{i+1})^2$
- Let α be a factor that can be tuned. Let $\beta = -1$ for A and B. Let $\beta = 1$ for C and D. $E = \sum_{i=1}^{i=N-1} [distance(path_i, path_{i+1})] - \alpha(\beta_i + \beta_{i+1})^2$

You picked up on your scanner that space pirates are on your tail. In this case, going through space debris will help conceal your location!

Q3.C.2 Which objective function would you choose given you prefer to cross the space debris? **(1.75 points)**

- ○ Let α be a factor that can be tuned. Let $\beta = -1$ for A and B. Let $\beta = 1$ for C and D. $E = \sum_{i=1}^{i=N-1} [distance(path_i, path_{i+1}) + \alpha(\beta_i - \beta_{i+1})^2]$
- ◉ Let α be a factor that can be tuned. Let $\beta = -1$ for A and B. Let $\beta = 1$ for C and D. $E = \sum_{i=1}^{i=N-1} [distance(path_i, path_{i+1}) - \alpha(\beta_i - \beta_{i+1})^2]$
- ○ Let α be a factor that can be tuned. Let $\beta = -1$ for A and B. Let $\beta = 1$ for C and D. $E = \sum_{i=1}^{i=N-1} [distance(path_i, path_{i+1})] + \alpha(\beta_i - \beta_{i+1})^2$
- ○ Let α be a factor that can be tuned. Let $\beta = -1$ for A and B. Let $\beta = 1$ for C and D. $E = \sum_{i=1}^{i=N-1} [distance(path_i, path_{i+1})] - \alpha(\beta_i + \beta_{i+1})^2$

Part D

(1 point)

You decide on using the following temperature schedule

$$T = T_0(1 - \tfrac{n}{N})^{\alpha}$$

Where n is current iteration, N is your iteration budget, T_0 is your current temperature, and α is a value that can be specified and tuned.

Q3.D.1 Suppose we increase α from 1 to 4. What can we expect to occur when running simulated annealing? **(1 point)**

- ○ greater number of iterations at higher temperatures
- ◉ greater number of iterations at lower temperatures
- ○ simulated annealing will take longer to find optima
- ○ nothing will change

Part E

(1 point)

Suppose we are performing a minimization using simulated annealing. Answer the following questions:

Q3.E.1 We generally move uphill but sometimes move downhill based on temperature. **(0.5 points)**

- ○ True
- ◉ False

Q3.E.2 When the temperature is 0, our algorithm will move downhill and converge to a local minimum. **(0.5 points)**

- ○ True
- ◉ False

4. Constraint Satisfaction Problems

(6 points)

The DotA [Defence of the Ancients] Internationals 2022 is underway and 6 teams [OG, EG, LGD, TNC, Spirit, Secret] are participating. The Internationals consists of exactly 2 phases. In **Phase I**, teams play 3 matches each and in **Phase II**, teams play 2 matches each. Every match ends with a win or loss and therefore, there are no ties. No two teams play against each other more than once in the entire tournament.

The results of the first two phases are as below:

Phase I:

1. Number of teams that won all the matches: 1
2. Number of teams that lost all the matches: 2
3. OG won against TNC
4. TNC won against LGD and Secret
5. Spirit lost in the game against EG
6. Spirit won against LGD and Secret
7. EG lost at least 1 match
8. The top team did not play against Secret.

Phase II:

1. The top team of **Phase I** lost all the matches
2. One of the teams in the lowest 2 ranks in **Phase I** lost all the matches
3. One of the teams in the lowest 2 ranks in **Phase I** won both the matches.
4. A total of three teams lost all their matches in **Phase II**.

Questions: [**Note:** All questions are multi choice]

Q4.1 Which team(s) won all their matches in **Phase II**? **(1.5 points)**

- ☐ TNC
- ☑ Spirit
- ☑ Secret
- ☐ LGD
- ☑ EG
- ☐ OG

Q4.2 Which team(s) won only 2 games in the entirety of **Phase I** and **Phase II**? **(2 points)**

- [x] TNC
- [ ] Spirit
- [x] Secret
- [ ] LGD
- [ ] EG
- [ ] OG

Q4.3 Which team(s) won the most matches after the end of **Phase II** (Overall competition)? **(2.5 points)**

- [ ] TNC
- [x] Spirit
- [ ] Secret
- [ ] LGD
- [x] EG
- [ ] OG

5. Probability

(6 points)

Answer the following questions

Q5.1 True or False (Mark **the box** if True and **leave blank** if False) **(2 points)**

- [x] P(~X|Y) = 1- P(X|Y)
- [ ] P(X|~Y) = 1-P(X|Y)
- [ ] P(~X|~Y)=1-P(X|Y)
- [ ] P(A,B,C) = P(A)P(B|A)P(C|A,B)
- [x] P(A,B,C) = P(A|B,C)P(B|C,A)P(C|A,B) if A,B,C are independent

Q5.2 A deck of 52 cards is shuffled and then divided into two piles of 13 cards and 39 cards. A card is drawn from the 13 card pile and it turns out to be an ace. The ace is then placed in the 39 card pile. The second pile (now containing 40 cards) is then shuffled and a card is drawn from it. Compute the probability that this drawn card is an ace. Please note that there are a total of 4 aces in a standard deck of 52 cards. **(4 points)**

0.082353

6. Bayes Nets

(8 points)

Health professionals recommend people to live a healthy life. Smoking and enjoying unhealthy food are known to impair people's immune system and increase the chance of getting severe diseases such as Cancer. Research has been conducted on the causes and symptoms of lung cancer, among people over 60-years old worldwide. The result reveals the following discoveries:

- Both smoking and eating unhealthy food contribute to the probability of the occurrence of lung cancer.
- When a lung cancer is developing, the initial symptom can be a sudden severe breathing difficulty, or sudden severe rash, or the coexistence of both symptoms simultaneously.
- 30% of people over 60-years old smoke. 40% of people over 60-years old enjoy unhealthy food.
- More probabilities between smoking, enjoying unhealthy food and lung cancer:
    - A person who both smokes and enjoys unhealthy food has a 15% probability of developing lung cancer.
    - A person who smokes but does not enjoy unhealthy food has a 10% probability of developing lung cancer.
    - A person who does not smoke but enjoys unhealthy food has a 8% probability of developing lung cancer.
    - A person who does not smoke and does not enjoy unhealthy food has a 5% probability of developing lung cancer.
- When lung cancer is developing, 60% of people have sudden severe breathing difficulty, and 10% of people have sudden severe rash. When no lung cancer is developing, 5% of people have sudden severe breathing difficulty, and 3% of people have sudden severe rash.

You may use the five nodes, Smoking (S), Unhealthy food (U), Lung cancer (L), Breathing difficulty (B), and Rash (R), to draw a Bayes net to assist your calculation. But you do not need to submit any Bayes net drawings. You only need to answer the 4 questions below.

Q6.1 One significant advantage of Bayes Net is the compact description of joint probability distribution for all nodes. How many probability values are required to specify this Bayes network? **(1.5 points)**

- ○ 31
- ○ 5
- ◉ 10
- ○ 32
- ○ 16

Q6.2 What are the correct independence relationships for the 2 scenarios given below? ('A ⊥ B' means A is independent of B). **(1.5 points)**

1) Breathing difficulty (B) and Rash (R) (B ⊥ R) ?

2) Breathing difficulty (B) and Rash (R) given Lung cancer (L) (B ⊥ R | L) ?

- ◉ Independent, Dependent
- ○ Independent, Independent
- ○ Dependent, Dependent
- ○ Dependent, Independent

Q6.3 What are the correct independence relationships for the 3 scenarios given below? ('A ⊥ B' means A is independent of B). **(2 points)**

1) Unhealthy food (U) and Rash (R) (U⊥ R) ?

2) Smoking (S) and Unhealthy food (U) given Lung cancer (L) (S ⊥ U | L) ?

3) Smoking (S) and Breathing difficulty (B) given Lung cancer (L) (S ⊥ B | L) ?

- ○ Independent, Independent, Independent
- ○ Independent, Dependent, Independent
- ○ Dependent, Independent, Dependent
- ○ Dependent, Dependent, Dependent
- ◉ Dependent, Dependent, Independent

Q6.4 What is the probability of a person over 60-year old NOT having lung cancer, given that person suddenly has severe breathing difficulty, but does NOT have a severe rash simultaneously? **(3 points)**

0.510129 (Note: Full points rounding to 6 decimals)

7. Machine Learning

(12 points)

For this question, you will help me to decide how I can make some practical improvements to my neural network model. I am working with the CIFAR-10 dataset, which has 60,000 small (32x32) images, each of which is labeled with one of 10 categories. I'd like to train a neural network to classify these images. Here are some example images:

I start off with the simplest type of neural network: a multi-layer perceptron (MLP). I decide to define a two-layer MLP, with an input for each pixel for each color (red, green, and blue), and an output for each of the 10 classes.The network looks like the following, where **bold lines** describe the **network**, and *italicized lines* describe the *features* as they propagate forward through the network:

input image (32 x 32 x 3)

↓

linear layer 1 (120 hidden units, bias term, activation function: sigmoid)

↓

intermediate features (120)

↓

linear layer 2 (10 hidden units, bias term)

↓

output (10 probabilities)

However, when I train this MLP, it seems to learn very slowly -- the training loss doesn't really seem to decrease at all, regardless of my optimization procedure. I wonder if maybe images are too big for fully connected layers.

Q7.1 To answer this, I'd like to know: how many learnable parameters are in this network? *(Hint: Remember to include the weights and bias of both layers!)* **(1 point)**

369970

Ah, exactly! That number is maybe a little too large, for the complexity of this problem. Instead, I'll be using this fancy class of architectures I keep hearing about: "convolutional neural networks." Don't worry -- you won't need to understand what the "convolutional" part means to help me out -- you just need to know that this is a neural network.

Below is the code for my first attempt at defining a neural network architecture for this task, which I've slightly altered from the Pytorch CIFAR-10 tutorial. If you've never seen pytorch before, this code should still be easy to understand -- in the *__init__* function, we define all of the modules in the network, and in the *forward* function, we run the network on some input and return the output.

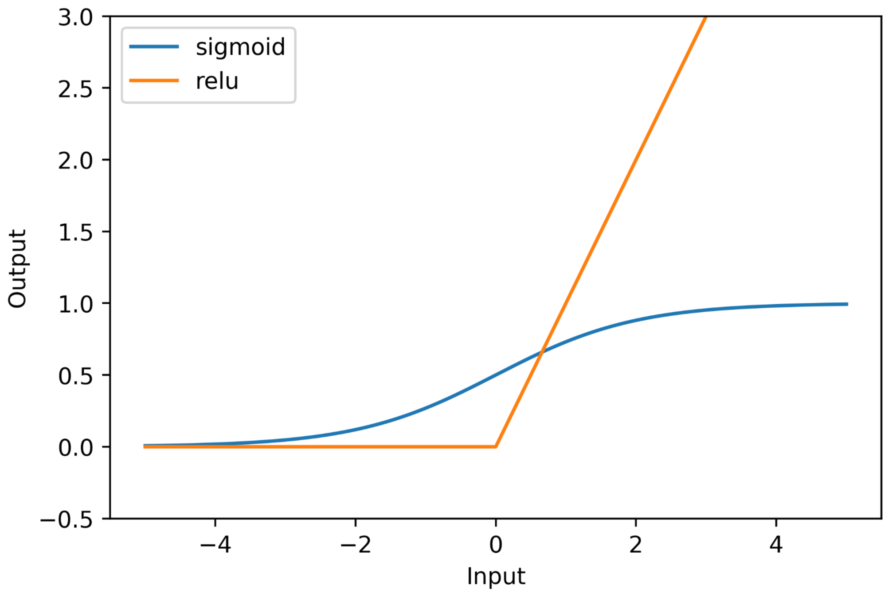
```
class Net(nn.Module):
    def __init__(self):
        super().__init__()
        # activation function
        self.act = nn.Sigmoid()
        # pooling function
        self.pool = nn.MaxPool2d(2, 2)
        # layer 1 (convolutional a.k.a. conv)
        self.conv1 = nn.Conv2d(3, 6, 5)
        # layer 2 (conv)
        self.conv2 = nn.Conv2d(6, 16, 5)
        # layer 3 (fully connected a.k.a. fc)
        self.fc1 = nn.Linear(16 * 5 * 5, 120)
        # layer 4 (fc)
        self.fc2 = nn.Linear(120, 84)
        # layer 5 (fc)
        self.fc3 = nn.Linear(84, 10)

    def forward(self, x):
        # layer 1: conv1 -> activation -> pooling
        x = self.pool(self.act(self.conv1(x)))

        # layer 2: conv2 -> activation -> pooling -> flatten
        x = self.pool(self.act(self.conv2(x)))
        x = torch.flatten(x, 1) # flatten all dimensions except batch

        # layer 3: fc1 -> activation
        x = self.act(self.fc1(x))

        # layer 4: fc2 -> activation
        x = self.act(self.fc2(x))

        # layer 5: fc3
        x = self.fc3(x)
        return x
```

I won't ask you to calculate how many parameters are in this network. Just know that the number is much smaller! This difference would be even more pronounced with images larger than 32x32.

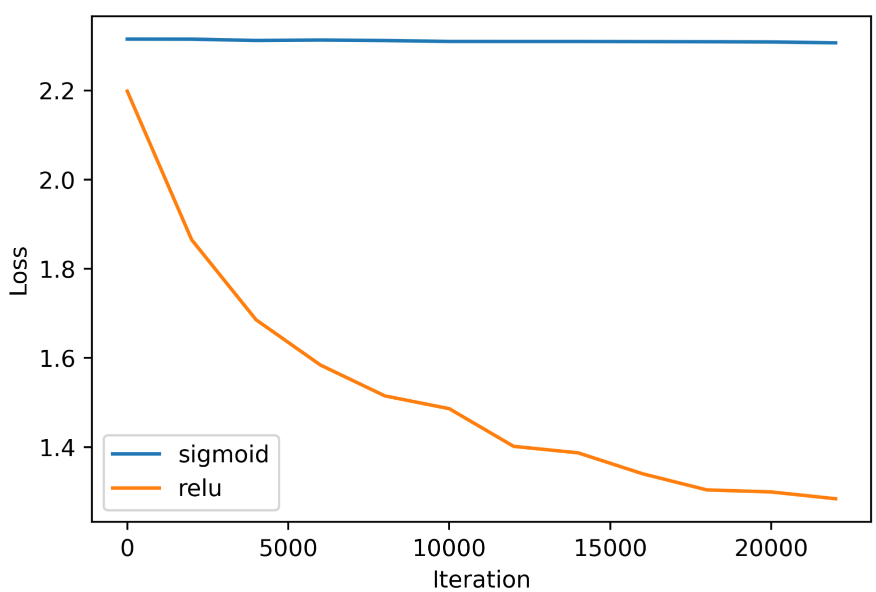
However, I've noticed this still doesn't work at all. Specifically, I notice that neither the training loss nor the validation loss seem to decrease during training. I take note of the fact that, while older, classical methods tended to use Sigmoid as an activation function, it is far more common nowadays to use a Rectified Linear Unit (ReLU). So, I change one line of code:

```
# activation function
self.act = nn.ReLU()
```

For your reference, the Sigmoid and ReLU activation functions are plotted below:



And, this change works like magic! Suddenly my training loss actually decreases! Below I've plotted the (smoothed) training loss per iteration, reported every 2000 iterations.



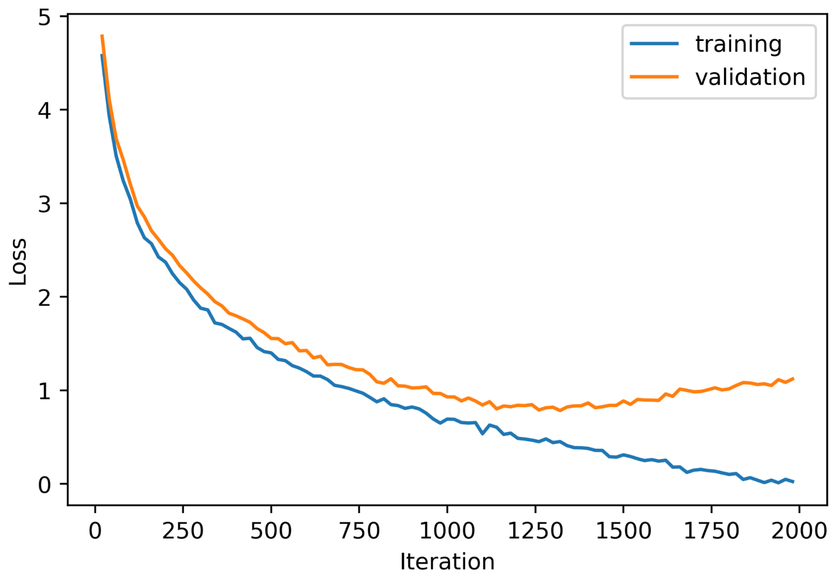
Q7.2 What concept *best* explains why ReLU seems to work so much better for my problem than Sigmoid? *(Hint: think about what happens during backpropagation)* **(1.5 points)**

- ◯ no-free-lunch theorem
- ◯ the curse of dimensionality
- ◯ underfitting
- ◯ overfitting
- ◯ a composition of linear functions is a linear function
- ◉ the vanishing gradient
- ◯ learning rate annealing

Q7.3 Well, I like the improvement, but ReLU still seems sketchy to me -- in particular, I don't like that it flattens all negative inputs. I decide to just remove the activation function from the network entirely. This really hurts my training (and validation) results -- neither loss decreases as much as it did with ReLU. What concept *best* explains why removing the activation function is a bad idea? **(2 points)**

- ◯ no-free-lunch theorem
- ◯ the curse of dimensionality
- ◯ underfitting
- ◯ overfitting
- ◉ a composition of linear functions is a linear function
- ◯ the vanishing gradient
- ◯ learning rate annealing

My roommate kindly let me borrow her bitcoin-mining machine for a day. It has 8 powerful graphics cards, so I decided to use a massive network for this task. The following is a plot of the network's loss on the training and validation sets throughout training:



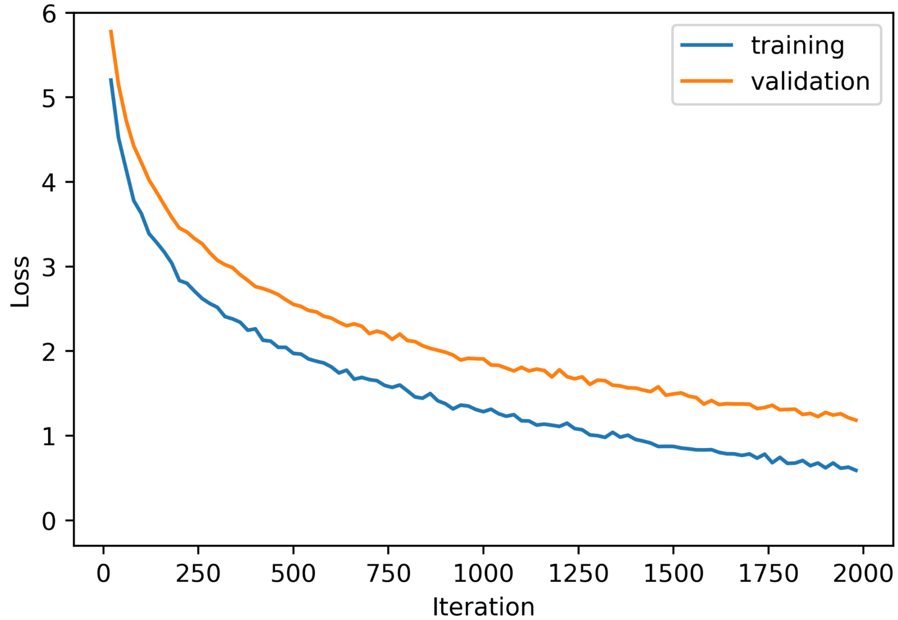
Q7.4 Which of the following *best* explains the problem I'm running into here? **(2 points)**

- ◯ no-free-lunch theorem
- ◯ the curse of dimensionality
- ◯ underfitting
- ◉ overfitting
- ◯ a composition of linear functions is a linear function
- ◯ the vanishing gradient
- ◯ learning rate annealing

Q7.5 In order to fix this problem, which of the following changes are likely to help? Select all that apply. *(Hint: only select the option(s) that directly address(es) the problem)* **(2.5 points)**

- [ ] lower the learning rate
- [ ] raise the learning rate
- [x] apply random transformations (e.g. flips, rotations, crops) to training images
- [x] use a smaller network
- [ ] use a larger network
- [x] stop training when validation loss plateaus
- [x] randomly zero out 20% of the features after every layer

What if I instead saw this loss plot:



Additionally, my evaluation metrics are unsatisfactory on both the training and validation sets.

Q7.6 In order to fix this problem, which of the following changes are likely to help? Select all that apply. **(3 points)**

- ☐ lower the learning rate
- ☑ raise the learning rate
- ☑ apply random transformations (e.g. flips, rotations, crops) to training images
- ☐ use a smaller network
- ☑ use a larger network
- ☑ stop training when validation loss plateaus
- ☑ randomly zero out 20% of the features after every layer

8. Pattern Recognition Through Time

(16 points)

Thomas Radsportler earned his PhD in CS from Georgia Tech in 2010 and went on to great success in developing Geographical Information Systems. His success afforded him time to pursue his passion for Cycling. More than anything else he dreams of someday riding the entire route of the Tour de France. Thomas has challenged himself to successfully complete the climbs along stages of the Tour, and with his expertise in CS has been compiling records of his efforts to chart his progress. Unfortunately, on his last trip his computer malfunctioned leaving some of his data incomplete. He has outputs measured at predetermined points on the route but has not been able to match several sets of data to his previous trips. He fears that due to the points being lost, this could be a big setback, so he decides to visit fellow GT grad, Sally Zeitfolgen for help.

Sally specializes in time-series Markov Chains. Thomas explains the situation to Sally, and asks for help. He is certain that the data is from Stage 7 of the Tour between Chalon-sur-Saône and Bourges, and would like to have her look at the data. Sally decides to start with Dynamic Time Warping as a possible solution[1].

Part A

(6 points)

Dynamic Time Warping (DTW) is an algorithm used to compare the similarity between two vectors of uneven length. It works well on the time series data you are given provided there is a consistent measurement used. Luckily, Thomas has used elapsed time as his measurement, so your task is to complete the DTW between the output provided and a known climb.

Fill in the matrix for the output (X values) and the known climb, Côte de Château-Chinon (Y values) and answer the 2 questions. For the matrix, you need to compute the squared difference between the matched y value and x value, and place that value in the corresponding y-row and x-column for credit.

Q8.A.1 Place the square root of the summed squared differences in the box below. **(0.5 points)**

24.718414

[1] Ratanamahatana, C. A., & Keogh, E. (2004, August). Everything you know about dynamic time warping is wrong. In *Third workshop on mining temporal and sequential data* (Vol. 32). Citeseer. Everything you know about Dynamic Time Warping is Wrong

25	196	729	1681	144	1936	900	2025	25
78	1521	676	144	1681	81	529	64	3364
53	196	1	169	256	256	4	289	2089
64	625	144	4	729	25	81	36	1936
58	361	36	64	441	121	9	144	1444
22	289	900	1936	255	2209	1089	2304	4
40	1	144	676	9	841	225	900	400
71	1024	361	25	1156	4	256	1	2601
62	529	100	16	625	49	49	64	1764
43	16	81	529	36	676	144	729	529
	39	52	66	37	69	55	70	20

Q8.A.2 Identify the constraints of DTW (Choose all that apply) **(2.75 points)**

- ☑ The warp path must begin at the beginning index of each time series
- ☑ The warp path must end at the ending index of each time series
- ☐ Every index must be used at most once on the longer path
- ☑ Both indices must be monotonically increasing in the warp path
- ☑ Every index of each time series must be used

Q8.A.3 Identify the ways to improve time and space complexity of DTW (Choose all that apply) **(2.75 points)**

- ☑ Limit the number of cells used to evaluate cost
- ☑ Abstract the data to perform DTW on a reduced representation of the data
- ☐ Advanced indexing, use bounding functions to reduce the number of times run
- ☑ Use the shortest path to iterate through all the possible paths
- ☐ Use methods b & c to iteratively coarsen (decrease resolution), project (map the coarse path), then refine (use a heuristic, to find the optimal path)

Part B

(10 points)

Fill out 2 Viterbi trellises for the next two problems per the directions given (5 points each)

So far Sally's hard work has paid off and Thomas is very thankful. He asks Sally if she has any ideas on how to build software to be able to predict completion time on his climbs. He can easily find out the weather for rain and winds prior to starting his climbs and has data on his speed. Sally suggests that if they could improve the model, that a Hidden Markov Model would work well.
Thomas tells Sally that for every climb he has recorded steep grades (SG), wind (W), and rain (R). Sally studies the data and comes up with her state models for part of one of the climbs.

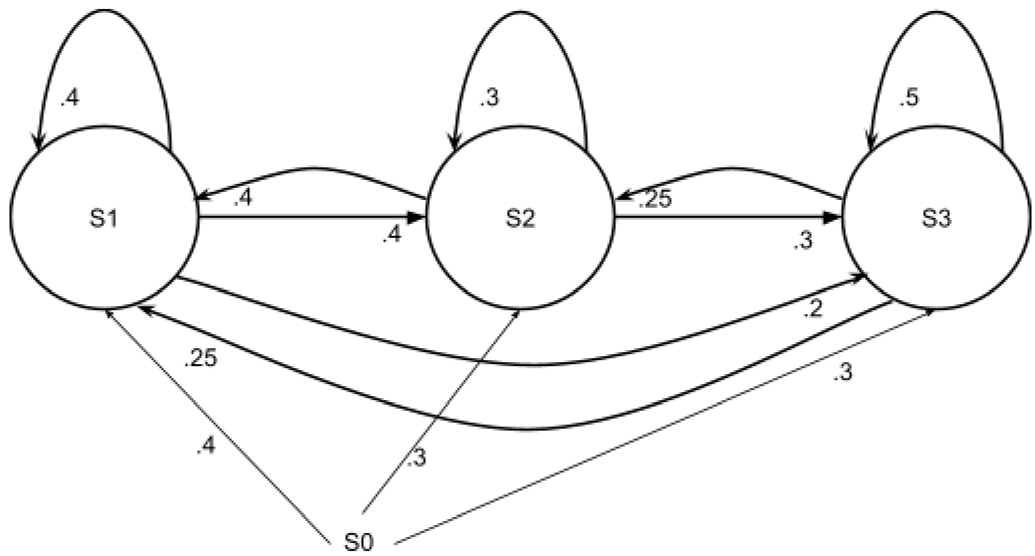
> **NOTE**: The provided link in the footnote to the **Rabiner** tutorial will be invaluable to complete this[2].
>
> Normally impacting observations in a problem such as this would require you to compute a gaussian percentage of the observation's impact. This would introduce new gaussians, and their integrals would increase computation errors. We want to ensure you know how to set up the problem, not just test your math. We have effectively turned this into the 3 state **Urn and ball** problem by using the W, R, SG observations to unitize partitions of the output. Enter state calculations for partial credit.

Assumptions

$\neg R \land \neg W \land \neg SG$	Fastest Speeds
One impacting condition $e.g.\ R$	Second Fastest Speed
Two impacting conditions	Third/Fourth Fastest Speed
Three impacting conditions	Slowest Speed
Un-denoted speed	10^{-4}
Order not enforced	R, W, G = W, R, G

State transition model

[2] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257-286. Retrieve from: [Tutorial on hidden Markov models and selected applications in speech recognition]



S1: Observations: $e.g.\ P(R \wedge W \wedge \neg SG | S1_{t1}, 35kph) = .1 * .4 = .04$

$P(R \wedge W \wedge SG) = 0.05;\ Output:\ 32kph \leq y < 34.72kph$
$P(R \wedge \neg W \wedge SG) = 0.1;\ Output:\ 34.72kph \leq y < 35.93kph$
$P(R \wedge W \wedge \neg SG) = 0.1;\ Output:\ 35.93kph \leq y < 36.67kph$
$P(R \wedge \neg W \wedge \neg SG) = 0.25;\ Output:\ 36.67kph \leq y < 38kph$
$P(\neg R \wedge \neg W \wedge \neg SG) = 0.5; Output:\ 38kph \leq y < 44kph$

State 1 models the flatter and wetter parts of the climb. There can be wind with some short steep grades along the route. It has the highest average output and the observations help explain away the variance in the outputs.

S2: Observations: $e.g.\ P(\neg W \wedge \neg R \wedge \neg SG | S2_{t1}, 36kph) = .3 * .3 = .09$

$P(W \wedge R \wedge SG) = 0.1;\ Output:\ 25kph \leq y < 30.16kph$
$P(W \wedge \neg R \wedge SG) = 0.2;\ Output:\ 30.16kph \leq y < 32.43kph$
$P(W \wedge R \wedge \neg SG) = 0.1;\ Output:\ 32.43kph \leq y < 33.24kph$
$P(W \wedge \neg R \wedge \neg SG) = 0.3;\ Output:\ 33.24kph \leq y < 35.58kph$
$P(\neg R \wedge \neg W \wedge \neg SG) = 0.3; Output:\ 35.58kph \leq y < 43kph$

State 2 models the intermediate parts of the climb, where there is more wind and more steep grades than State 1. There will be more steep grades and winds along the route and less rain than State 1. On average it has the mid-range outputs and the observations help explain away the variance in the outputs.

S3: Observations: $e.g.\ P(\neg SG \wedge \neg W \wedge \neg R | S3_{t1}, 36kph) = .3 * .1 = .03$

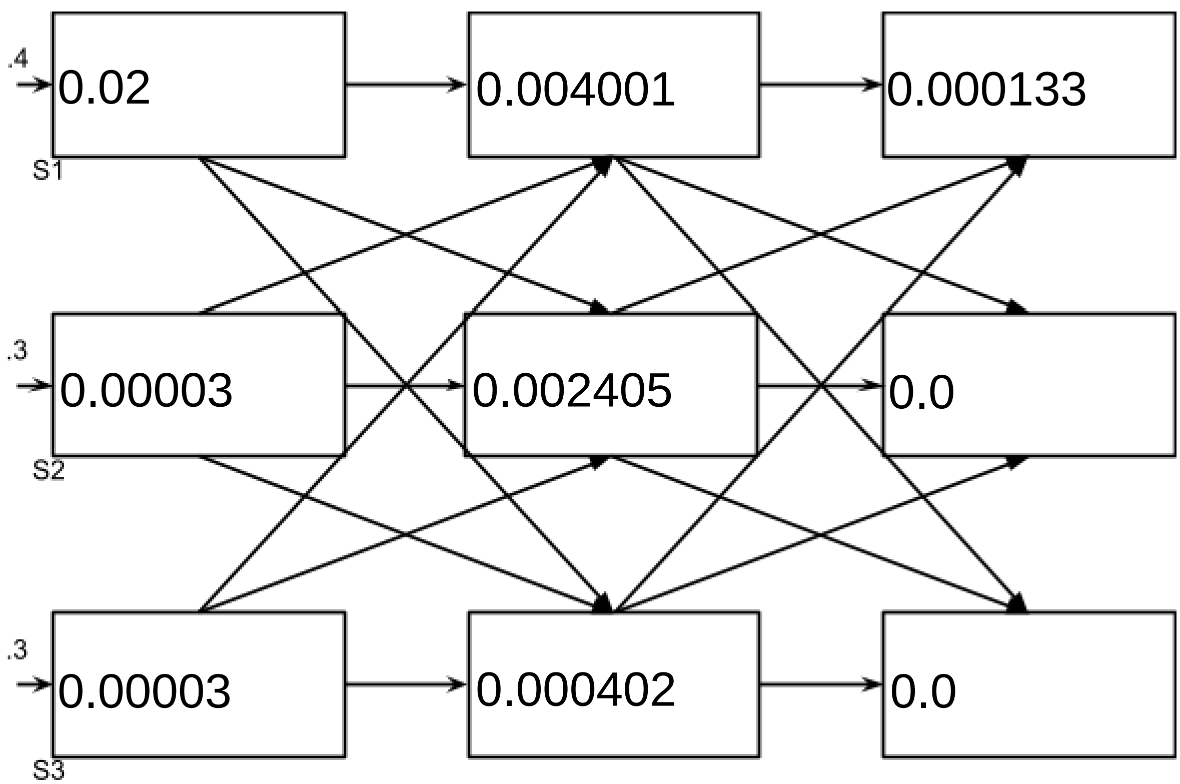
$P(SG \wedge W \wedge R) = 0.1;\ Output:\ 18kph \leq y < 24.88kph$
$P(SG \wedge \neg W \wedge R) = 0.1;\ Output:\ 24.88kph \leq y < 26.63kph$
$P(SG \wedge W \wedge \neg R) = 0.2;\ Output:\ 26.63kph \leq y < 28.98kph$
$P(SG \wedge \neg W \wedge \neg R) = 0.5;\ Output:\ 28.98kph \leq y < 35.13kph$
$P(\neg SG \wedge \neg W \wedge \neg R) = 0.1; Output:\ 35.13kph \leq y < 42kph$

State 3 models the steepest parts of the climb, where there is some wind and rain. There are longer and steeper grades than State 1 or State 2. On average it has the lowest outputs and the observations help explain away the variance in the outputs.

Sally wanted to understand the probability that an output came from the model so we will attempt an evaluation first using the forward algorithm with the Viterbi trellis.

Q8.B.1 Use the HMM to determine the likelihood that the following output came from this model. **(5 points)**

Output: 32, 38, 34
$Observations\ t1(W \wedge R \wedge SG); t2(\neg W \wedge \neg R \wedge \neg SG); t3(W \wedge R \wedge SG)$



Place your state probability values from your calculations in the node boxes provided for credit. Round to 6 decimal places.

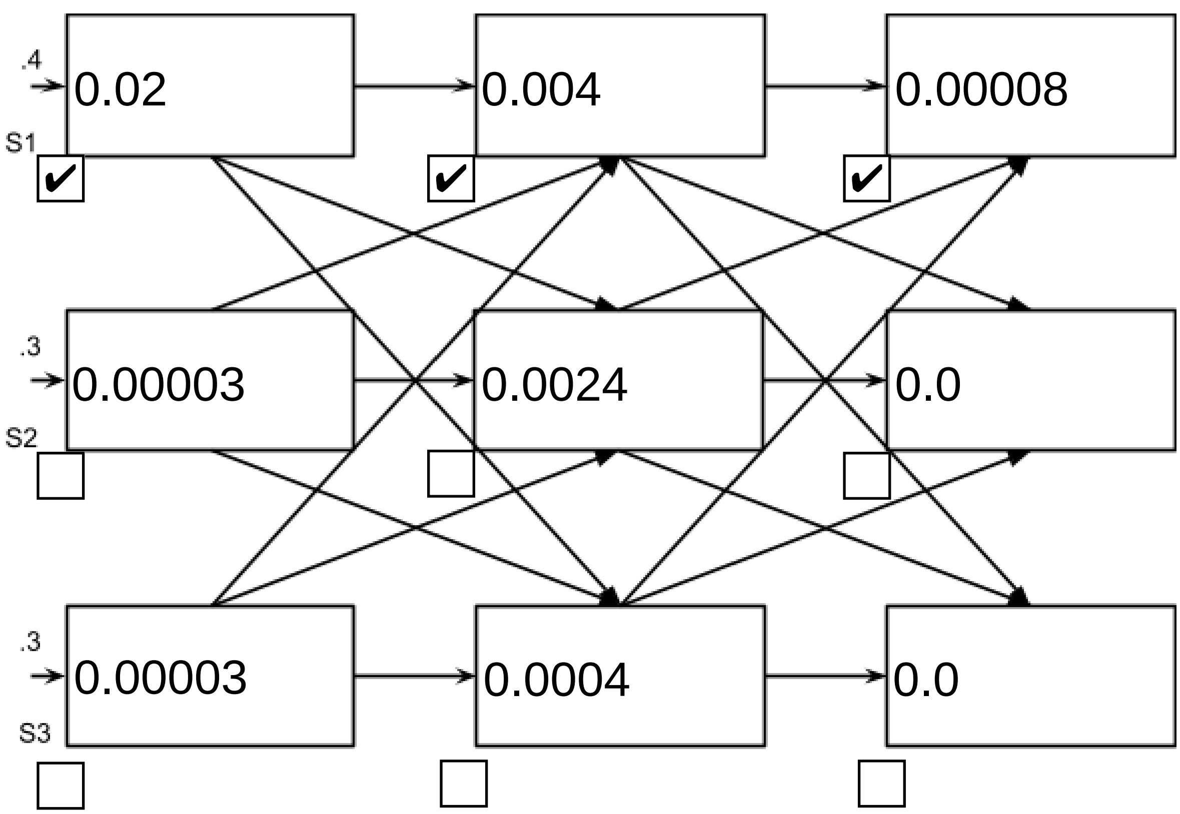
Hint: From Rabiner, you should note it will take N(N+1)(T-1)+N multiplications and N(N-1)(T-1) additions or 27 multiplications and 12 additions. Due to the construction of S0, this removed 3 additions and 6 of the 9 multiplications

Next, Sally was curious to see if she could find the most likely path that Thomas took for this climb. She decided to make a most-likely path calculation using the Viterbi algorithm.

Q8.B.2 Use the Viterbi algorithm to determine the most likely path based on the model. Use the node box for your calculation and the bottom left checkbox to indicate the nodes on the most likely path. **(5 points)**

Output: 32, 38, 34

$Observations\ t1(W \wedge R \wedge SG); t2(\neg W \wedge \neg R \wedge \neg SG); t3(W \wedge R \wedge SG)$



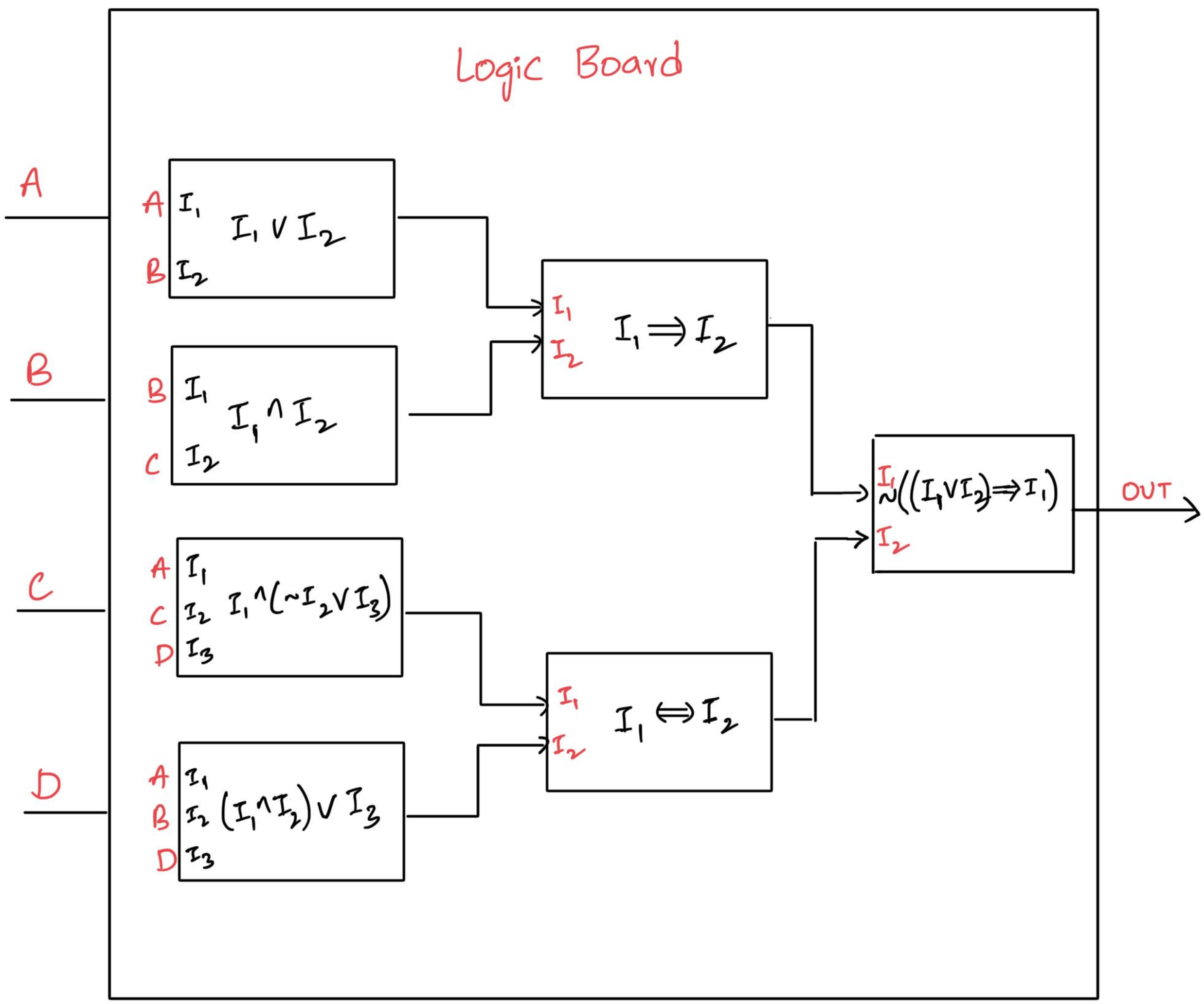
9. Logic and Planning

(12 points)

Part A: Logic

(6 points)

In 2077, a robot needs your help in determining the output from the following logic board to reboot itself. Fill the table below according to the inputs given to the logic board.



Q9.A.1 Fill out the below table with the output values from the above circuit. **(4 points)**

A	B	C	D	Out
T	T	T	T	F
T	T	T	F	F
T	T	F	T	T
T	T	F	F	T
T	F	T	T	T
T	F	T	F	T
T	F	F	T	T
T	F	F	F	F
F	T	T	T	F
F	T	T	F	F
F	T	F	T	F
F	T	F	F	T
F	F	T	T	F
F	F	T	F	F
F	F	F	T	F
F	F	F	F	F

Q9.A.2 Convert the logic in the rightmost block to [~((I1 v I2) => I1)] to its CNF (conjunctive normal form). **(2 points)**

I2 ^ ~ I1

Part B: Planning

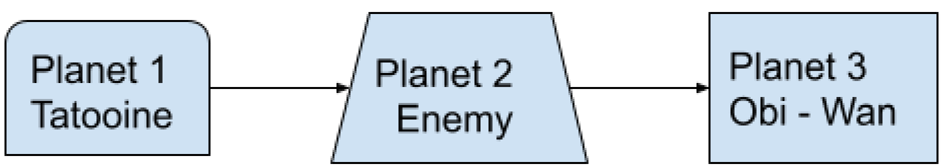
(6 points)

A New Hope

A long, long time ago in a galaxy far, far away, an evil galactic empire has taken charge over the galaxy with the help of their powerful leader Darth vader. Humanity's only hope is in a message recorded by princess Leia, which has to be delivered to Obi-wan Kenobi. Before being taken hostage by Darth Vader, Princess Leia passes the message to a robot called R2-D2. In this question you have to program in PDDL (Planning Domain Definition Language) to successfully deliver the message to Obi-wan Kenobi and help defeat the evil galactic empire.

After taking the message from princess Leia, R2-D2 barely escapes from Darth Vader and his army of stormtroopers, and reaches a planet called Tatooine. But R2-D2 has to pass by another planet which is home to a Stormtrooper base, in order to reach planet 3 where Obi-wan stays. In order to achieve this task R2-D2 will need the help of a friend named Luke Skywalker.

Now R2-D2's plan of action is to first pass the message to Luke Skywalker on planet Tatooine. Then Luke will pass the enemy planet without getting noticed by Storm troopers, so that he can reach planet 3 where he will hand the message over to Obi-wan.



Notations:
R2D2 - character (c1)
LukeSkywalker - character (c2)
Stormtroopers - character (c3)
Obi-wan - character (c4)
Message - m
Planet1, Planet2, Planet3 - p1,p2,p3

Actions:

1. MessageLuke(c1,m,p1,c2)
2. PassEnemyBase(c2,m,p2,p3,c3)
3. MessageObi(c2,m,c4,p3)

Types: <Subtype - Supertype> (a type with no Superclass is the Superclass of all types)
Object
Location - Object

Planet - Location
Character - Object
Message - Object

Predicates:
Object(o) - returns true if **o** is an object.
Message(m) - returns true if **m** is a message.
Planet(p) - returns true if **p** is a planet.
Character(c) - returns true if **c** is a character.
MeetFriend(c,f) - returns true if a character **c** meets a character **f**, and **f** is a friend of **c**.
SeeEnemy(c,e) - returns true if a character **e** is an enemy of character **c**, and character **c** gets noticed by the enemy **e**.
Hold(m,c) - returns true if **m** is a message, **c** is a character and character **c** is holding the message **m**.
At(c,p) - returns true if **p** is a planet, **c** is a character and character **c** is on the planet **p**.
Pass(m,c) - returns true if **m** is a message, **c** is a character and message **m** is being passed to character **c**.

Initializations:
init(character(c1) ∧ character(c2) ∧ character(c3) ∧ character(c4) ∧ message(m) ∧ planet(p1) ∧ planet(p2) ∧ planet(p3))

Q9.B.1 You want to add the action MessageLuke(c1,m,p1,c2), this action will cause R2-D2 (c1) to pass the message (m) to Luke (c2) on Planet1 (p1). Given the Preconditions, you have to select the effects from the options below. **(2 points)**

Preconditions: At(c1,p1) ∧ Hold(m,c1) ∧ At(c2,p1) ∧ MeetFriend(c1,c2)
Select the applicable clauses for the effect from the below options.

- [ ] ~At(c2,p1)
- [x] Pass(m,c2)
- [ ] ~SeeEnemy(c1,c2)
- [x] Hold(m,c2)
- [ ] Pass(m,c3)
- [x] ~Hold(m,c1)

Q9.B.2 You want to add the action PassEnemybase(c2,m,p2,p3,c3), this action will cause LukeSkywalker (c2) to pass the enemy base planet (p2) without getting noticed by the enemies (c3), and finally reach planet(p3) while holding the message(m). Given the Preconditions, you have to select the effects. **(2 points)**

Preconditions: ~At(c3,p2) ∧ Hold(m,c2) ∧ ~SeeEnemy(c2,c3)
Select the applicable clauses for the effect from the below options.

- [x] Hold(m,c2)
- [ ] ~Hold(m,c2)
- [ ] SeeEnemy(c2,c3)
- [ ] At(c3,p2)
- [x] At(c2,p3)

Q9.B.3 You want to add the action MessageObi(c2,m,c4,p3), which will cause Luke (c2) to pass the message(m) to Obi-wan(c4) on Planet(p3). Given the effects, you have to select Preconditions. **(2 points)**

Effect : ~Hold(m,c3) ∧ Pass(m,c4) ∧ Hold(m,c4)
Select the applicable clauses for the PreConditions from the below options.

- [x] MeetFriend(c2,c4)
- [x] At(c2,p3)
- [x] At(c4,p3)
- [ ] Hold(m,c3)
- [ ] Hold(m,c4)
- [ ] SeeEnemy(c2,c3)

10. Planning Under Uncertainty

(18 points)

Description of the Task -

Welcome to the 2021 Edition of the TriWizard Tournament! Our protagonist Harry Potter finds himself representing Gryffindor from the Hogwarts School of Witchcraft and Wizardry. Setting the theme right and moving away from any more movie references, Harry is determined to find all the Golden Snitches in the CS6601 variation of the Quidditch Tournament (where more than 1 snitch is present in the game) and meet Dumbledore at one corner to successfully complete the game. Your goal is to find the optimal way for Harry to grab hold of all the Snitches since one of them conceals the Resurrection Stone which he must use to defeat _______. (He, who must not be named).

The Quidditch Environment

Bringing the environment in the lecture videos closer to our realistic problem, Harry finds himself in a 4-by-4 grid world where each cell denotes a possible position of Harry in the space, along with 2 Golden Snitches (favourable cells). Harry will be competing against the best seeker from Quidditch World Cup - Victor Krum and a close friend from the school - Cedric Diggory who are present in different cells in the same environment (unfavourable/sink states). The game will be OVER if Harry shares a cell with any of them at any point in the process and he'll lose his only chance at winning the Triwizard Cup (a key to his final battle with ________)

Harry starts at (1,1) as shown in Table 1 and his goal is to meet Prof. Dumbledore at the diagonally opposite corner (4,4) with both the snitches. Krum and Diggory are located at the coordinates (3,1) and (2,3) respectively while the snitches are located at (4,2) and (2,4). Please note, that except for Harry, who can fly in the environment to **a neighboring cell**, all other characters and objects are *stationary*. A cell is "neighboring" if it shares a boundary. Each snitch will have a reward of 100 points to Gryffindor! If Harry is successful in making it to the other corner, Dumbledore will further award 100 points! On the contrary, landing into one of the cells occupied by Krum or Diggory would render the game over with -200 points to Gryffindor **:(**

Uncertainty of Actions

In the given Grid-World, Harry uses his Firebolt (Harry's flying broomstick) to navigate in the environment and needs to collect all the Snitches. He can fly towards a cell by moving one box at a time in any of the possible ways - i.e. he can move from a cell to another neighboring cell which is either directly up, directly down, adjacent diagonal cell, etc (see Figure 1). Upon attempting to move from one cell to the other, there is a **72% chance** of landing in the

“Planned” cell, while Harry can land at any of the other candidate **“neighboring cells”** with a uniform probability.

Initially each cell is assigned a value **0.0**, except the ones that denote the coordinates for Dumbledore, Victor and Diggory as shown in Table 2 below. The environment has spells casted by Barty Crouch Jr. disguised as Prof. Moody, which penalises Harry’s moves to an empty cell with a reward of **- 4 points** (Cells occupied with the Golden Snitches or Characters aren’t considered empty)**.** Consider a **discount factor (γ) of 0.9.**

Help Harry in determining the state values to be filled in the matrix upto 2 iterations of Value Iteration Algorithm for winning in the CS6601 Edition of Triwizard Tournament Environment!

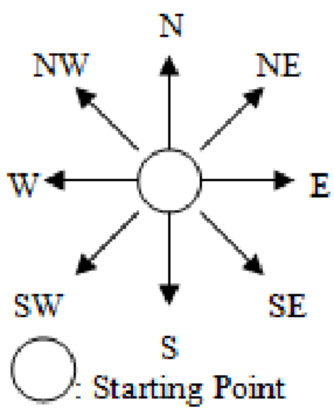


Figure 1 : Starting from a state/cell, Harry can move in any of the 8 directions within grid

(1,1) Harry	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3) Diggory, Reward = -200 (Sink)	(2,4) Snitch, Reward = +100
(3,1) Victor Krum, Reward = -200 (Sink)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2) Snitch, Reward = +100	(4,3)	(4,4) Dumbledore, Reward= +100 (Sink)

Table 1 : CS6601 Quidditch Environment

i = 0

Value = 0.0 (1,1)	Value = 0.0 (1,2)	Value = 0.0 (1,3)	Value = 0.0 (1,4)
Value = 0.0 (2,1)	Value = 0.0 (2,2)	**Value = -200** **(2,3)** **Diggory (Sink)**	Value = 0.0 (2,4) Snitch
Value = -200 **(3,1)** **Victor (Sink)**	Value = 0.0 (3,2)	Value = 0.0 (3,3)	Value = 0.0 (3,4)
Value = 0.0 (4,1)	Value = 0.0 (4,2) Snitch	Value = 0.0 (4,3)	**Value = +100** **(4,4)** **Dumbledore (Sink)**

Table 2 : Initialization of state values for the Value-Iteration algorithm

Q10.1 Fill the blanks in the matrices below. These are the state values which are computed during the Value-Iteration algorithm.

i = 1 (First Iteration of Value-Iteration)

Value = -4.0 (1,1)	Value = **-16.6** (1,2)	Value = -16.6 (1,3)	Value = -29.2 (1,4)
Value = -16.6 (2,1)	Value = -18.4 (2,2)	Value = -200 (2,3) Diggory (Sink)	Value = **+87.4** (2,4) Snitch
Value = -200 (3,1) Victor (Sink)	Value = **-18.4** (3,2)	Value = 53.6 (3,3)	Value = 48.2 (3,4)
Value = **-29.2** (4,1)	Value = 87.4 (4,2) Snitch	Value = 60.8 (4,3)	Value = +100 (4,4) Dumbledore (Sink)

Table 3 : State values after the first iteration of Value-Iteration algorithm

Q10.2 i = 2 (Second Iteration of Value-Iteration)

Value = **-19.1668** (1,1)	Value = -22.4554 (1,2)	Value = 35.9906 (1,3)	Value = 25.3436 (1,4)
Value = -22.5562 (2,1)	Value = **+13.7336** (2,2)	Value = -200 (2,3) Diggory(Sink)	Value = 122.284 (2,4) Snitch
Value = -200 (3,1) Victor (Sink)	Value = 40.0424 (3,2)	Value = 62.492 (3,3)	Value = 60.9134 (3,4)
Value = **+25.1168** (4,1)	Value = +**127.1764** (4,2) Snitch	Value = 71.5604 (4,3)	Value = +100 (4,4) Dumbledore (Sink)

Table 4 : State values after the second iteration of Value-Iteration algorithm

Note :

- Please round your values up to 6 decimal points for each of the cells
- At every step, Harry must move to a neighboring cell (Except for the sink states, in which case the game ends)
- Sink states are terminal states, which means, if entered, Harry won't be able to move out of those and the game ends
- Each blank carries one point

11. Extra Credit

(Up to 2 points)

CIOS! We will be awarding extra credit to the entire class based on CIOS completion rates.

The completion rate for CIOS is defined as follows:

(number of students who completed CIOS) / (number of students enrolled in the class)

The completion rate (up to 1.0) will be multiplied by 2 and added to the score of your final exam. We only award these points if the completion rate is at least 0.8.

Please consider taking the CIOS! As we have said over the course of the semester, this course is constantly evolving and we are looking at ways of making it better. We take CIOS feedback very seriously and incorporate it into our teaching methods wherever possible.

Checklist

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

- ✔ I have read the pinned Piazza post with the title Final Exam Clarifications Thread', and I am familiar with all of the clarifications made by the Teaching staff.
- ✔ All answers with more than 6 digits after the decimal point have been rounded to 6 decimal places.
- ✔ All pages are being uploaded in the correct order that they were presented to me, and none of the pages are missing/removed.
- ✔ Any extra pages **(including blanks)** are only attached at the END of this exam, after this page with clear pointers to wherever the actual answer is in the PDF (reference properly).
- ✔ I am submitting only one PDF and nothing else (no docx, doc, etc.).
- ✔ The PDF I am submitting is not blank (unless I want it to be).
- ✔ **I will go over the uploaded pictures on Gradescope and make sure that all the answers are clearly visible. I acknowledge that I am aware that dull / illegible / uneven scans will not be graded.**
- ✔ I have submitted a copy of the PDF to Canvas.